Universitat Politècnica de Catalunya

Unsupervised Learning
of
Relation Detection Patterns

Tesi presentada per a l'obtenció del títol de Doctor

Edgar Gonzàlez i Pellicer

Director
Doctor Jordi Turmo i Borràs

Programa de Doctorat en Intel·ligència Artificial
Departament de Llenguatges i Sistemes Informàtics

Abstract

Information extraction is the natural language processing area whose goal is to obtain structured data from the relevant information contained in textual fragments.

Information extraction requires a significant amount of linguistic knowledge. The specificity of such knowledge supposes a drawback on the portability of the systems, as a change of language, domain or style demands a costly human effort. Machine learning techniques have been applied for decades so as to overcome this *portability bottleneck*—progressively reducing the amount of involved human supervision. However, as the availability of large document collections increases, completely unsupervised approaches become necessary in order to mine the knowledge contained in them.

The proposal of this thesis is to incorporate clustering techniques into pattern learning for information extraction, in order to further reduce the elements of supervision involved in the process. In particular, the work focuses on the problem of relation detection. The achievement of this ultimate goal has required, first, considering the different strategies in which this combination could be carried out; second, developing or adapting clustering algorithms suitable to our needs; and third, devising pattern learning procedures which incorporated clustering information.

By the end of this thesis, we had been able to develop and implement an approach for learning of relation detection patterns which, using clustering techniques and minimal human supervision, is competitive and even outperforms other comparable approaches in the state of the art.

Resum

L'extracció d'informació és l'àrea del processament de llenguatge natural l'objectiu de la qual és l'obtenir dades estructurades a partir de la informació rellevant continguda en fragments textuals.

L'extracció d'informació requereix una quantitat considerable de coneixement lingüístic. La especificitat d'aquest coneixement suposa un inconvenient de cara a la portabilitat dels sistemes, ja que un canvi d'idioma, domini o estil té un cost en termes d'esforç humà. Durant dècades, s'han aplicat tècniques d'aprenentatge automàtic per tal de superar aquest *coll d'ampolla de portabilitat*, reduint progressivament la supervisió humana involucrada. Tanmateix, a mida que augmenta la disponibilitat de grans col·leccions de documents, esdevenen necessàries aproximacions completament no-supervisades per tal d'explotar el coneixement que hi ha en elles.

La proposta d'aquesta tesi és la d'incorporar tècniques de clustering a l'adquisició de patrons per a extracció d'informació, per tal de reduir encara més els elements de supervisió involucrats en el procés En particular, el treball se centra en el problema de la detecció de relacions. L'assoliment d'aquest objectiu final ha requerit, en primer lloc, el considerar les diferents estratègies en què aquesta combinació es podia dur a terme; en segon lloc, el desenvolupar o adaptar algorismes de clustering adequats a les nostres necessitats; i en tercer lloc, el disseny de procediments d'adquisició de patrons que incorporessin la informació de clustering.

Al final d'aquesta tesi, havíem estat capaços de desenvolupar i implementar una aproximació per a l'aprenentatge de patrons per a detecció de relacions que, utilitzant tècniques de clustering i un mínim de supervisió humana, és competitiu i fins i tot supera altres aproximacions comparables en l'estat de l'art.

A tots aquells que, durant els darrers 8 anys,
heu estat preguntant-me:

"*I la tesi, què? Quan l'acabes?*"

Ara ja us puc respondre.

Contents

Acknowledgements

Agraïments

Escriure la meva tesi ha estat una gran empresa, amb alguns moments—per què negar-ho—difícils. Tanmateix, em considero una persona realment afortunada perquè, a més a més d'haver tingut l'oportunitat de treballar en un camp que m'agrada amb passió durant 8 anys, ho he fet envoltat d'algunes de les persones més excepcionals que he conegut mai. No voldria entrar al cos d'aquest document sense abans aprofitar aquestes pàgines per agrair-los el seu suport i els seus ànims durant tot aquest temps.

Sens dubte, la primera persona a qui voldria mencionar és al meu director Jordi Turmo, amb qui hem estat donant forma al que finalment ha acabat essent el contingut d'aquesta tesi. A través de les incomptables reunions que hem compartit, ha estat un supervisor atent i involucrat, però que, tanmateix, m'ha deixat tota la llibertat del món per a treballar i explorar el que jo cregués convenient. Sempre ha cregut ferventment que la recerca que estàvem duent a terme acabaria portant a algun lloc—fins i tot quan jo mateix ja hauria deixat de fer-ho—i, clarament, també ha demostrat destacables dosis de paciència fins que ha arribat el dia en què això ha passat.

Tampoc no em puc imaginar el meu pas pel doctorat sense companys del Omega S107 (i afegits) que, dia rere dia, han estat qui em donava la motivació per a sortir del llit al matí i córrer cap al metro per arribar a temps a l'hora del cafè. Després dels cafès i els dinars van venir les birres i els sopars, després de les birres i els sopars van venir els caps de setmana i els viatges, i un bon dia va resultar que potser érem alguna cosa més que companys de feina. Així que gràcies a la Cris, l'Emili, la Gemma, l'Ignasi, en Jordi, en Jesús, la Leo, la Maria, la Marina, la Montse, la Muntsa, en Pere, en Roberto, la Solmaz, l'Stefan, la Txell i en Xavi, per tots els bons moments que hem passat plegats.

També vull donar les gràcies a la resta del Grup de Processament de Llenguatge Natural que, des del mateix dia que vaig entrar-hi, han estat pendents de fer-me sentir un més de l'equip—amb tots els avantatges (sovint en forma de pernils i vermuts) que això comporta. D'entre ells, amb alguns com l'Alicia, l'Horaci, els Lluïsos, en Mihai i en Xavi he tingut la sort de poder compartir moltes estones, tant esforçant-nos per poder participar en avaluacions a altes hores de la matinada, com per deixar neta una taula plena de pa, vi i formatges—o, fins i tot, per córrer contra el crono durant 21 kilòmetres i 97.5 metres.

Més enllà del nostre grup, als passadissos del departament de Llenguatges i Sistemes Informàtics també hi he trobat cares amigues rere les portes. Vull donar les gràcies en particular a en Jordi i en Salvador, tant per l'oportunitat que em van donar de ser professor de P1 (una altra de les bones experiències d'aquests anys), com per tots els cops de mà—i bons consells—que m'han donat sempre que els he necessitat. Diagonal enllà, al Departament de Lingüística de la Universitat de Barcelona, em sento afalagat d'haver merescut la confiança que han dipositat en mi la Mariona i la Toni, així com de l'amistat de les Martes.

Durant la carrera també vaig tenir sort de trobar-me amb una colla d'amics com l'Àlex, en Carles, en Dani, en David, en Derek, l'Esteve, en Néstor, l'Omar i en Xavi. Han anat passant els anys, i encara que cadascú ha seguit el seu camí (alguns dels quals a l'altra banda del món), intentem no deixar escapar mai les ocasions per a trobar-nos—ocasions a què la Sandra tampoc mai no falta, no sense abans carregar-se d'una bona dosi de paciència.

I encara que sembli mentida després d'uns quants paràgrafs, també em relaciono amb gent que no són informàtics. Tot i que no els dedico tot el temps que es mereixen, sé que l'Agnès, la Bibiana i la Glòria, la Laura, en Pau, la Silvia, en Simón i la Marta, la Sonia, o en Xavi, sempre estan a l'altra costat del telèfon, a punt per a fer un sopar i veure'ns les cares. Tampoc no voldria deixar-me als companys de Six, la gran banda en què, amb en Carles, en Daniele, el Quim i en Xavi, tinc el privilegi de fer frankfurts, birres i soroll, molt de soroll.

He deixat per al final aquells de qui hauria de dir tant que no sé ni per on començar. I és que, durant 31 anys, la meva família ha estat al meu costat, donant-me suport i un bon grapat de bons moments. Així, per als meus pares Dani i Isa, la meva germana Astrid, les meves àvies Anna Maria i Isabel, el meu oncle David, la meva tia Encarna, i l'Aguedí no tinc més que paraules d'agraïment i estimació. Ni tampoc no puc oblidar-me del meu cunyat Pere, amb qui ens uneix alguna cosa més que la família; ni del Mapashito, amb qui el temps que vam passar plegats ho va fer definitivament massa ràpid.

Per acabar, vull deixar constància de la sort que he tingut de, durant aquests últims anys, haver pogut viure el present—i ja fins i tot de mirar al futur—amb una dona com la Laia al meu costat. Sé que haver de compartir-me amb el Processament de Llenguatge Natural sovint se li fa difícil. Però també sé que, quan el deixo de banda, sempre trobo la seva mà, que agafa la meva i no la deixa anar, sigui on sigui que decidim d'anar plegats.

Acknowledgements

Writing my thesis has been a great endeavour, with some hard moments. Nevertheless, I consider myself lucky because, in addition to having been given the chance to work on a field that I like with passion for 8 years, I have done so surrounded by some of the most outstanding people I have ever met. I do not want to proceed to the body of this document without using these pages to thank them for all their support and encouragement during all this time.

I want thus to thank here the people at the Proteus Project of the New York University, in particular Professor Satoshi Sekine. They gave me not only the chance to perform a research stay at their group in the Spring of 2007, but also gave me a warm reception and made this stay one of the best experiences of my PhD. Among them, I cannot forget to mention Yusuke, with whom I shared one office, one friendship, and endless hours of great conversation.

Funding

We all need the money to keep going—and I am not an exception to this rule. So I must also acknowledge that the work in this thesis has been possible with the economical support of a number of public institutions, including:

- the *Departament d'Universitats, Recerca i Societat de la Informació* of the *Generalitat de Catalunya*, through the FI-2005 and FI-2006 grants;
- the *Departament d'Educació i Universitats* of the *Generalitat de Catalunya*, through the FI-2007 and BE-2007 grants;
- the *Comissionat per a Universitats i Recerca* of the *Departament d'Innovació, Universitats i Empresa* of the *Generalitat de Catalunya*, through the FI-2008 grants;
- the *Ministerio de Educación y Ciencia* of the *Gobierno de España*, through the Text-Mess project (TIN2006-15265-C06);
- the *Ministerio de Ciencia e Innovación* of the *Gobierno de España*, through the KNOW2 project (TIN2009-14715-C04-01).
- the European Social Fund, through the FI-2005 to FI-2008 and BE-2007 grants;
- the European Commission, through the CHIL project (IP-506909).

Unsupervised Learning
of
Relation Detection Patterns

1 Introduction

> At first the names he read were meaningless to him, as deeply anonymous as their phone numbers. The only distinction a name had was its accidental yet ineluctable place in the alphabetical order, and then whatever idiot errors the computer could dress it in, which Smoky was paid to discover. (That the computer could make as few errors as it did impressed Smoky less than its bizarre witlessness; it couldn't distinguish, for instance, when the abbreviation "St." meant "street" and when it meant "saint," and directed to expand these abbreviations, would without a smile produce the Seventh Saint Bar and Grill and the Church of All Streets.)
>
> John Crowley
> *Little, Big*

THE AMOUNT of information produced and stored in the World is increasing every day at a strenuous rate. Lyman and Varian (2003) already talk about *information explosion*, and estimate that "*new stored information grew about 30% a year between 1999 and 2002*". However, although in terms of bits information consumption is still rising fast in developed countries, this increment mainly comes from the improvements in transmission quality. In terms of words, consumption is going up at a much slower pace, as it is "*constrained by human physical limits, including the length of a day and reading speed*", and its growth "*will never exceed a few percent per year*" (Bohn and Short, 2009).

The mismatch between information production and consumption growth rates creates an expanding gap between them, and has spurred the development of information-processing systems. These systems monitor, filter, analyze, route and/or integrate the produced information, with the common ultimate goal of making it reach, in a more or less processed form, its potential consumers. A number of systems focus on textual information, and we can hence talk of text-processing systems and text-processing tasks.

1.1 Information Extraction

Appelt and Israel (1999) consider a small subset of these text-processing tasks, which is detailed in Table 1.1. As seen in the table, between generic file manipulation, in which text documents are treated as opaque sequences of bytes, and full text understanding, as (hopefully) performed by humans, an spectrum of tasks exists, which require different intermediate levels of linguistic understanding.

In some cases, only simple text manipulation is involved, comparable to that performed by Unix utilities as `grep` or `wc`. While this kind of tools have been widely used since the early seventies, especially with the rise of the Unix family (which was initially designed as a "*word-processing system*"; Ritchie, 1979), it was not until the early eighties that more complex systems (in text-

A text document. . .	
. . . *is a sequence of bytes.*	File manipulation
. . . *is a sequence of characters.*	Text manipulation
. . . *is a sequence of words (perhaps meaningful units).*	Information retrieval
. . . *contains meaningful phrases/clauses, relevant to a particular topic.*	Information extraction
. . . *is an article, an essay, a story, a novel. . .*	Text understanding

Table 1.1: Text-processing tasks (adapted from Appelt and Israel, 1999)

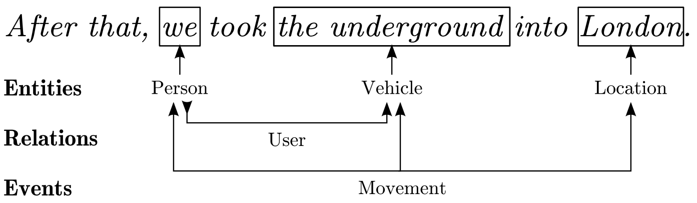


Figure 1.1: Information extraction from a sample sentence

processing terms) began to be developed. The purpose of those first text-based intelligent (TBI) systems was to automatically obtain information by manipulating documents, instead of relying on manual introduction of knowledge by human experts (Jacobs, 1992). At that time, roughly two major TBI areas were considered and distinguished: information retrieval (IR) and information extraction (IE).

IR techniques aim at retrieving the documents from a collection which satisfy a given set of restrictions, defined by means of a *query*. The documents that are judged relevant according to the query can be directly provided to the human user, or alternatively be used as input to an information acquisition process. The prototypical, and so far most usual, setting for IR is that of finding those documents containing a number of keywords; and even if there exists a wide variety of IR systems, the use of linguistic knowledge and technologies within them has been marginal, and much controversial (Spärck-Jones, 1999; Smeaton, 1999; Voorhees, 1999).

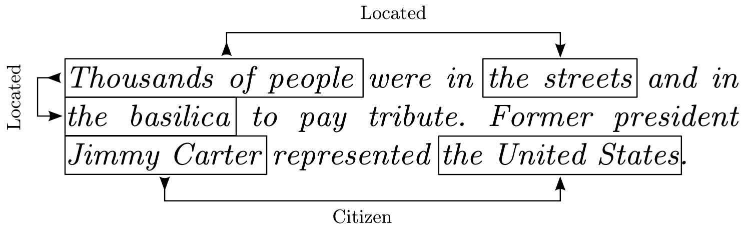
On the contrary, IE technology aims at a deeper understanding of the texts, to identify the relevant content within the relevant documents and extract it in a structured way, suitable for human inspection, insertion in a structured database, or further automatic processing. Text documents used in IE usually belong to a restricted domain, and there exists an a priori definition of the set of concepts which are considered relevant: the so-called *scenario of extraction*. The goal of IE is to extract instances of these concepts occurring in the documents. In order to do so, IE systems require significant amounts of linguistic knowledge (Muslea, 1999). For this reason, IE is universally considered to be part of the much broader area of natural language processing (NLP).

The concepts defined for extraction can differ enormously from scenario to scenario. However, even if the semantics are varied, the *nature* of the involved concepts is usually restricted, being:

- *entities* having an identity in the real world,
- *relations* between these entities, either binary or n-ary;
- and *events* in which these entities are involved.

An example of the information which may be extracted from a sample sentence, at each one of these three levels, is shown in Figure 1.1. Other scenarios may require the detection of different types of entities, relations and events; but the fact that concepts belonging to these three kinds have to be detected will remain.

The identification of each kind of information may require different strategies, and this has resulted in the crystallization of entity, relation and event recognition as separate tasks within the full IE problem. The distinction has been acknowledged by researchers, who have built IE systems


Located
Located
Thousands of people were in the streets and in the basilica to pay tribute. Former president Jimmy Carter represented the United States.
Citizen

(a) Passage, entities and relations

Located	
Thousands of people	*the basilica*
Thousands of people	*the streets*

Citizen	
Jimmy Carter	*the United States*

(b) Tabular view

Figure 1.2: Relation extraction from a sample passage

which solve only one of these problems, or more than one but independently; and by evaluation organizers, who have defined separate entity, relation and event recognition tracks in the major IE evaluations carried out so far (Chinchor, 1998; Doddington et al., 2004).

A number of variations on these and other tasks have been defined through the years, and new challenges for researchers in IE appear every year.

1.1.1 Relation Extraction

Among the three presented IE tasks, *relation extraction* has been chronologically the last one to be considered. Even though IE evaluations were already being held in the late eighties, and the entity and event recognition problems were distinguished no later than 1995, the first major evaluation to feature a separate relation recognition track did not take place until 1998 (Chinchor, 1998).

However, relations are a cornerstone of human cognitive processes, and have been acknowledged as such since early times. Aristotle himself includes the *relatives* (τα̈ προ̈ς τι, *things toward something*) as one of the ten basic *categories* of speech (τα̈ λεγόμενα, *things that are said*) in his homonymous work *Categories*, written in the 4th century BCE. Hence, the ability to capture relational information is tantamount to the development of useful IE systems.

In more modern terms, the definition of relation in IE follows its mathematical use, and is that of a set of n-tuples of entities (whichever entities are defined in the scenario of extraction). Figure 1.2a shows the results of relation extraction on a sample passage in which the entities had been previously recognized, and figure 1.2b contains the same information in tabular form, a representation which emphasizes the tuple-like nature of relations.

Even if the term *relation detection* has sometimes been used as a synonym of *relation extraction* (e.g., by Zhao and Grishman, 2005), in this document we will use its more constrained meaning, as the task of identifying the tuples of entities which are related in some way. Relation extraction can then be seen as the task which encompasses both the detection and the classification of the relations into a set of categories—predefined or not.

1.2 Machine Learning for Adaptive Information Extraction

As mentioned in Section 1.1, IE requires a significant amount of specific linguistic knowledge. At the core of most IE systems lies a set of linguistic patterns which are used to extract the concepts present in the text, and which are highly language-, domain- and style-specific. A change of a single one of these factors often render the patterns ineffective. Moreover, the manual acquisition, or adaptation, of these patterns—and other required linguistic knowledge—can become highly expensive, demanding human experts on the extraction scenario to undertake a lengthy involvement.

The specificity of the knowledge required for IE hence supposes a drawback on the portability of IE systems. This *portability bottleneck* of IE, as well as the success of machine learning (ML) and corpus-based approaches in producing competitive systems in many NLP tasks—and at a lower cost than their hand-crafted equivalents—has encouraged research in the application of ML techniques to the construction of *adaptive* IE systems, which could be easily ported to other domains, styles and languages. During the last two decades, a plethora of such adaptive IE systems have been developed (Turmo et al., 2006).

1.2.1 Towards Unsupervised Information Extraction

ML approaches allow the acquisition of knowledge for IE at a lower cost than those based on manual introduction by a human expert. However, in many cases the burden of the acquisition task still falls mostly on the expert, as she is responsible for feeding the ML system the examples from which to learn. Traditional supervised systems demand hand-annotated corpora; and even if techniques such as active learning (in which the system interactively demands the annotation of those examples from which it expects to learn the most) can reduce the required volumes of data, the production of such corpora remains a strenuous endeavour.

For this reason, efforts have been devoted to remove elements of supervision in ML processes in general, and in that of extraction pattern learning in particular. This has given birth to a number of weakly supervised methods for IE pattern acquisition, which have successively reduced the elements of human involvement in the process. Weakly supervised methods can benefit from unannotated text collections, and may only require from the expert:

- a definition of the domains present in the collection;
- and/or the classification of a small set of documents (the *seed documents*) into the different domains, or, alternatively, a small set of patterns representative of these domains (*seed patterns*).

Nevertheless, this decrease in human supervision presents a drawback: the more reduced the input from the expert, the more sensitive to it the learning process. Human decisions can hence introduce a strong bias in weakly supervised approaches. To minimize the impact of such decisions at an early stage, it may be necessary to browse a significant fraction of the documents in the collection. However, this becomes not only expensive, but even unfeasible, when trying to exploit, for instance, the huge textual databases that are becoming increasingly available.

Additionally, the fact that the selection of relevant seed documents or patterns demands an a priori definition of the domains reduces the utility of IE as an exploratory tool for new and unknown domains.

It remains an open issue, hence, how we can exploit, in an exploratory and unbiased manner, large document collections, which may come, partially or wholly, from unfamiliar domains. Or, otherwise stated: how we can build a system which achieves *unsupervised information extraction.*

1.3 Clustering

Data clustering can be traced back to the first half of the twentieth century (Tryon, 1939), and can be defined as "*the organization of a collection of patterns [...] into clusters based on similarity. Intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster*" (Jain et al., 1999).

The utility of clustering techniques as an automatic tool for exploratory analysis of data collections is well-known (Hartigan, 1975; Dubes and Jain, 1980; Dimitriadou, 2003). Clustering has been successfully used in areas as diverse as image segmentation (Silverman and Cooper, 1988), bridge building (Reich and Fenves, 1991) or genetics (Eisen et al., 1998).

In particular, the application of clustering to fields such as genetics is of particular interest to our discussion because of the challenges they pose. The volumes of data involved are huge, and there is often ignorance about what these data might contain. Unsupervised approaches are indispensable in this context, a fact which is acknowledged for instance by Eisen et al. (1998): "*As we have little a priori knowledge of the complete repertoire of expected gene expression patterns for any condition, we have favored unsupervised methods or hybrid (unsupervised followed by supervised) approaches*".

Moreover, the presence of noisy or just irrelevant data also poses problems of its own. In this context, traditional clustering methods may need to be replaced by *robust clustering* (Davé and Krishnapuram, 1997) and *minority clustering* (Ando and Suzuki, 2006) ones.

1.4 Our Proposal

The starting point of this thesis are the two facts we have exposed in the last two sections:

- Weakly supervised approaches for IE pattern acquisition are unsuitable for exploration of large document collections from unfamiliar domains. Elements of supervision strongly bias the learning process, and avoiding them requires comprehensive examination of the data, with increasing human costs (Section 1.2.1).

- Unsupervised clustering techniques have been successfully applied in areas involving huge collections of data whose contents were mostly unknown (Section 1.3).

It seems thus reasonable to think that clustering techniques can effectively help to overcome the limitations of weakly supervised approaches, and push the state of the art of IE pattern acquisition one step closer to the goal of unsupervised information extraction.

In this thesis, we propose the incorporation of clustering techniques into the process of IE pattern learning, to effectively remove elements of human supervision. We believe that guiding the whole process of pattern learning by an automatic analysis of the data, without an explicit human bias, will avoid the drawbacks of existing seed-based approaches.

Our goal is hence to develop a methodology that, from a completely unannotated collection of documents, and without the need of any kind of expert-given seed documents or patterns, produces good quality patterns, useful for IE and possibly other NLP tasks. And our proposal to achieve this goal is to enhance the pattern generation process with clustering techniques.

A graphical representation of the differences between existing approaches and the one we propose is depicted in Figure 1.3. As mentioned in Section 1.2.1 weakly supervised approaches often require manual selection of a set of seeds from which to start the pattern learning process (Figure 1.3a). Our proposal is to remove this process of manual seeding with the help of clustering techniques.

We conceive at least three different ways in which clustering and pattern learning can be combined:

- Figure 1.3b shows clustering and pattern learning as independent processes in a sequential combination, the latter taking as input the output of the former. This is the simplest approach.

- Figure 1.3c shows clustering and pattern learning as independent collaborative processes, each one receiving input from and giving output to the other.

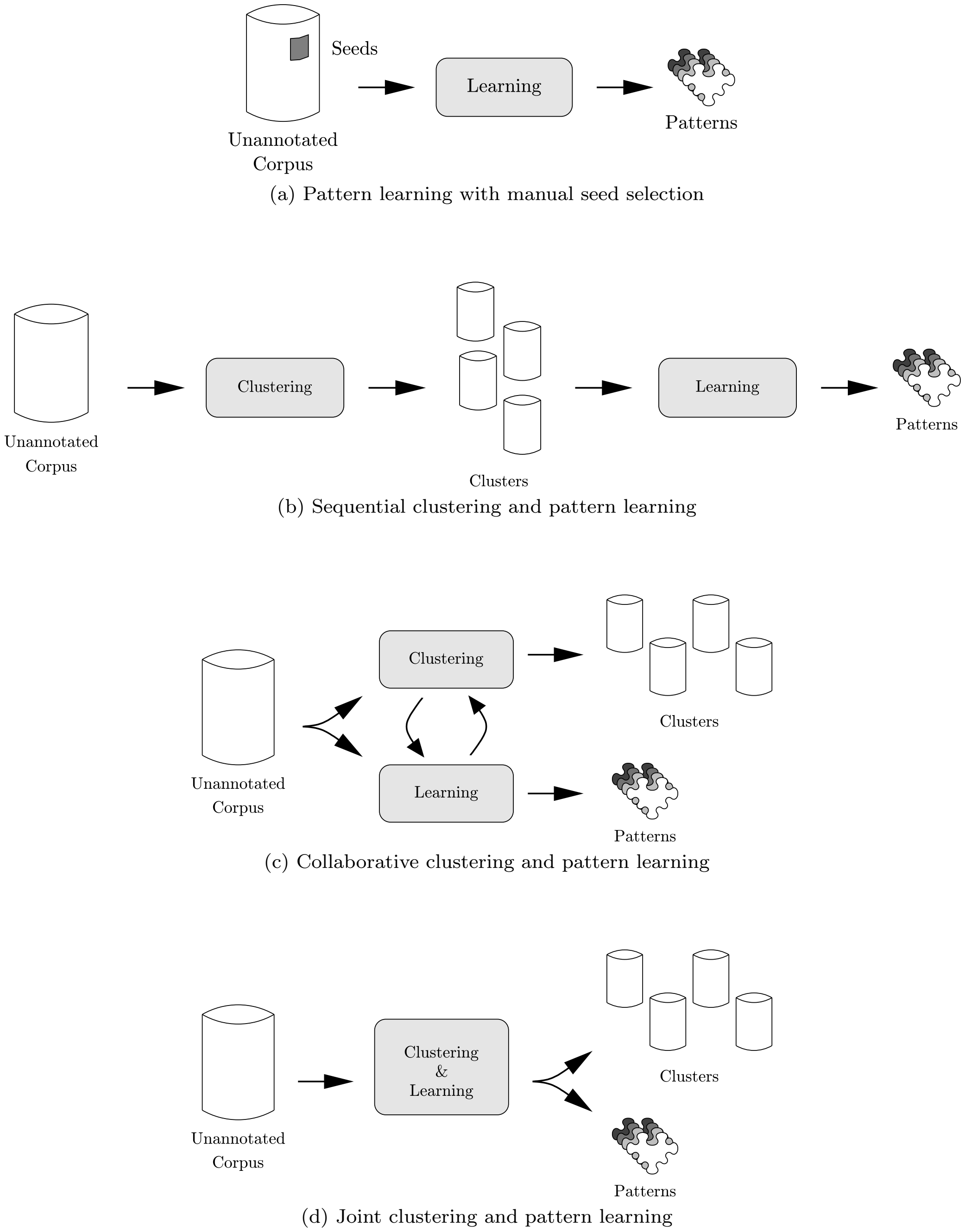
- Figure 1.3d shows clustering and pattern learning as a unique joint process. Clusters and patterns are learned at the same time by a single learner.

The feasibility of enhancing pattern learning by clustering depends on the availability of suitable clustering methodologies. For this reason, in a first step our primary focus of research has been clustering, and then shifted towards the application of this newly developed clustering methodologies to enhance the pattern learning process.

In order to concentrate our efforts, we have deliberately chosen to focus our research only on relation detection patterns (Section 1.1.1). However, we believe that the same or similar techniques can be successfully applied to entity and event extraction.

1.5 Overview of this Document

The rest of this document is organized as follows. Chapter 2 contains a historical review, up to the state of the art, of the development of weakly supervised pattern acquisition approaches for relation extraction. Chapter 3 is the first one to consider the problem of clustering, focusing on the task in its usual setting, and Chapter 4 incorporates the techniques studied therein to devise and evaluate pattern learning approaches using sequential and collaborative schemes. Chapter 5 also deals with clustering, but considers the alternative setting of minority clustering. Minority



(a) Pattern learning with manual seed selection

(b) Sequential clustering and pattern learning

(c) Collaborative clustering and pattern learning

(d) Joint clustering and pattern learning

Figure 1.3: Approaches for IE pattern learning

clustering techniques are key to the development of learning approaches based in a joint scheme, which are described and evaluated in Chapter 6. Finally, Chapter 7 summarizes all work done, draws conclusions from it, and sketches possible future lines of research.

A number of appendices, with supplementary material not included within the body of the thesis for fluidity of the exposition, are included at the end. Appendix A provides the definitions of common mathematical concepts which are used through the document. Appendix B contains the formal proofs of secondary propositions which are omitted in the main text; and Appendix C contains, in a diagrammatic fashion, the evolution of the entity and relation type hierarchies used in the ACE evaluations—the main IE framework which we have used for our evaluation. Finally, Appendix D lists the publications which have been produced with the research in this thesis.

2 State of the Art

Qui oblidi els mestres està destinat a la mediocritat i a la repugnància artística.

Arnau Tordera

This chapter presents an overview of the state of the art in weakly supervised approaches to pattern acquisition for relation extraction.

Section 2.1 contains a historical perspective of the development of IE, centered around the evaluations that have shaped it. Next, Section 2.2 focuses on the approaches which have been proposed for weakly supervised relation extraction, and Sections 2.3 and 2.4 provide an overview at the elements of supervision and pattern formalisms which have been used within these approaches, respectively. Last, Section 2.5 discusses the issue of IE evaluation, and the proposals which have been made regarding it.

One of the first reported IE systems operating on texts of unrestricted topic was implemented by de Jong (1979, 1982). His FRUMP system monitored a newswire using simple scripts to cover news stories. Sager (1981) mentions an even earlier project, before 1970, directed by Sager herself, from the Linguistic String Project Group at New York University. Sponsored by the American Medical Association, the work sought to convert patient discharge summaries (filled out in English) into a structure for a traditional database management system.

2.1 Development of Information Extraction

Nevertheless, the development of IE is clearly tied to the series of evaluations which, sponsored by the Defense Advanced Research Projects Agency (DARPA) of the United States government, have been defining IE tasks and assessing the performance of IE systems since 1987: the Message Understanding Conference (MUC), Automatic Content Extraction (ACE) and Text Analysis Conference (TAC) series. Even if other conferences have held IE evaluations, those in the MUC-ACE-TAC triad have undoubtedly been the longest lasting and most influential ones.

Figure 2.1 contains a timeline of the IE evaluations which took place between 1987 and 2011, showing the various relations between the tasks therein defined. Solid arrows denote continuity in a task; whereas dashed arrows denote the incorporation of elements or ideas from previous tasks.

Next Sections 2.1.1 to 2.1.3 give a historical overview of the periods spanned by the MUC, ACE and TAC evaluations, respectively, and detail the various IE tasks that were considered at each time.

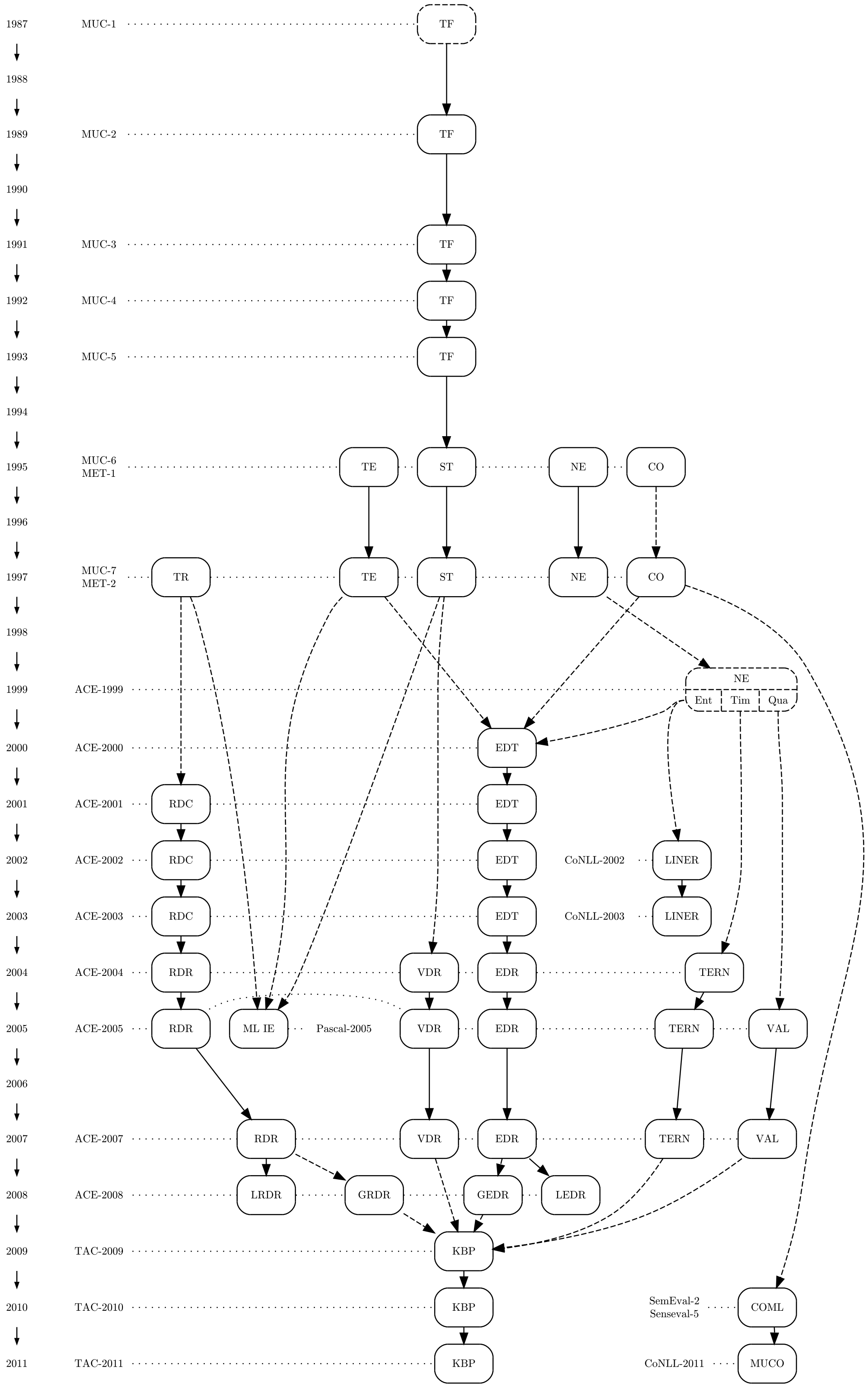


Figure 2.1: Timeline of tasks in IE evaluations

Coreference Resolution (CO), Coreference Resolution in Multiple Languages (COML), Entity Detection and Recognition (EDR), Entity Detection and Tracking (EDT), Global Entity Detection and Recognition (GEDR), Global Relation Detection and Recognition (GRDR), Knowledge Base Population (KBP), Local Entity Detection and Recognition (LEDR), Language-Independent Named Entity Recognition (LINER), Local Relation Detection and Recognition (LRDR), Machine Learning for Information Extraction (ML IE), Modelling Unrestricted Coreference in OntoNotes (MUCO), Named Entity (NE), Relation Detection and Recognition (RDR), Relation Detection and Characterization (RDC), Scenario Template (ST), Template Element (TE), Time Expression Recognition and Normalization (TERN), Template Filling (TF), Template Relation (TR), Value Detection and Recognition (VAL), Event Detection and Recognition (VDR)

2.1.1 Message Understanding Conference Era

The Message Understanding Conferences (MUC) were started in 1987 by the US Navy (the Naval Ocean Systems Center, San Diego). They soon attracted the attention of DARPA, which decided, in 1991, to start the TIPSTER text program to fund the research efforts of several of the MUC participants. The program, in addition to IE, promoted research in IR and automatic summarization, and also launched, in parallel, the Text Retrieval Conferences (TREC) to spur the development of these two technologies.

The first five editions of MUC, held between 1987 and 1993, focused on the *Template Filling* (TF) task: a certain domain was considered (*Naval tactical operations*, *Latin American terrorism*, *joint ventures* or *microelectronics*), and for each occurrence of an event within a predefined set, a template with a number of slots had to be filled, capturing its information. Whereas the MUC-1 organizers did not define the slots in the template nor any evaluation metrics, by MUC-5 two sets of object-oriented templates were being used, and the evaluation included the classical precision, recall and F1 metrics, as well as an alternative set of error-oriented ones (see Section 2.5).

From the experience in previous evaluations, in 1995 the organizers of MUC-6 decided to split Template Filling into a number of subtasks, with three main goals in mind:

- *Identifying domain-independent components among those which were being developed.* To meet this goal, the organisers proposed the *Named Entity* (NE) subtask, which involved recognizing and classifying entities (organizations, people, locations), temporal expressions (dates, times) and quantities (monetary values, percentages).

- *Focusing on the portability of the IE task to different event classes.* The organization proposed to standardize low-level objects (people, organizations. . .) since they were involved in many different types of events. The *Template Element* (TE) subtask was proposed with this aim. The old-style MUC task of detecting the events in which the template elements were involved was named *Scenario Template* (ST) task.

- *Encouraging work on deeper understanding.* Three more subtasks were proposed with this goal, namely *Coreference Resolution* (CO), *Word Sense Disambiguation* (WSD) and *Predicate-Argument Syntactic Structuring* (PASS). Finally, due to lack of agreement about the definition of the other two, only the first task was evaluated.

The final installment of the MUC series, MUC-7, held in 1997, continued the 4 tasks evaluated in MUC-6, and added that of *Template Relation* (TR), which required the identification of relations such as *location-of*, *employee-of* and *product-of* holding between template elements.

It is interesting to mention that, by that time, the conferences were no longer solely focused on the English language: From MUC-5 on the evaluations contained subtasks in Japanese, and MUC-7 also contained a Chinese subtask. Moreover, MUC-6 and MUC-7 were held jointly with the two editions of the Multilingual Entity Task (MET) conference, which included an NE evaluation on Japanese, Chinese and Spanish documents.

2.1.2 Automatic Content Extraction Era

At the end of TIPSTER, DARPA decided to launch the Translingual Information Detection, Extraction and Summarization (TIDES) program, which focused on making information in other languages accessible to English speakers. TIDES continued the TREC conferences; and held between 2001 and 2007 the Document Understanding Conference (DUC) series, which evaluated automatic summarization systems.

Regarding IE, the main evaluation vehicle of TIDES were the Automatic Content Extraction (ACE) evaluations, organized by the National Institute of Standards and Technology (NIST). In addition to the stronger emphasis on multilinguality (the English, Chinese and Arabic languages were included), ACE also differed from MUC on the sources of the used documents: in addition to newswire text, the first ACE corpora included broadcast news and newspaper text, both manually and automatically transcribed. The aim was to study the drop in performance experimented by IE systems when processing degraded inputs.

Even if OCR versions of newspaper were dropped in ACE-2003, and ASR versions of broadcast news in ACE-2005, this last evaluation incorporated in their place documents coming from broadcast conversations, conversational telephone speech, Usenet groups, and weblogs.

The first sketches of the ACE evaluation plan were written in 1999, and at that time a MUC-style NE task was considered, including as subtasks proper named entities (persons, locations, organizations), temporal expressions and quantities. However, by the time the first pilot ACE evaluation was held the next year, there had been a significant change with respect to previous tests. In the ACE evaluations "*the research objectives are defined in terms of the target objects (i.e., the entities, the relations, and the events) rather than in terms of the words in the text*" (Doddington et al., 2004). This distinction between the *objects* and the *mentions to the objects* lead to the definition, in most ACE evaluations, of diagnostic tasks at the mention level, in parallel to the corresponding main tasks at the object level.

The tasks that were considered during the lifespan of ACE were the following:

- The *Entity Detection and Tracking* (EDT) task, introduced in 2000, focused only on the *entity* subtask from the 1999 NE task. The fact that it was object- rather than mention-based implied that systems had to incorporate some kind of coreference resolution so as to merge the mentions corresponding to the same discourse entity. It was hence related to the NE, TE and CO tasks from MUC. The task was renamed to *Entity Detection and Recognition* (EDR) for ACE-2004.

- The *Relation Detection and Characterization* (RDC) task, introduced in 2001, required the identification of explicit and implicit relations between the found entities, and was hence an extension of the TR task from MUC. The task was also renamed to *Relation Detection and Recognition* (RDR) for ACE-2004.

- The *Event Detection and Recognition* (VDR) task, introduced in 2004 but not evaluated until 2005, required the identification of events in which the found entities were involved. However, the concept of event was simpler than that in ST from MUC, being always expressed by means of a trigger word.

- The *Time Expression Recognition and Normalization* (TERN) task, also introduced in 2004, required the detection of time, date and duration expressions, as defined in the TIMEX2 standard (Mani et al., 2001). It was the extension of the *time* subtask from the 1999 NE task.

- The *Value Detection and Recognition* (VAL) task, introduced in 2005, required the detection of non-entity values which were involved in any of the events from VDR. It was an ad hoc extension of the *quantity* subtask from the 1999 NE task, biased towards the type of events that were to be recognized in the evaluation.

For each object and mention which was detected, a number of attributes had to identified. In particular, several two-level hierarchies of types and subtypes were used for entities, relations and events in the successive evaluations.

The only edition in the series to differ significantly from this evaluation scheme was the last one. For ACE-2008, the organizers dropped all tasks but EDR and RDR, and incorporated two corresponding tasks of *Global Entity Detection and Recognition* (GEDR) and *Global Relation Detection and Recognition* (GRDR), in which mentions of entities and relations had to be merged across several documents. The previous document-wise tasks were retronymly renamed *Local Entity Detection and Recognition* (LEDR) and *Local Relation Detection and Recognition* (LRDR).

During the timespan of ACE, another two IE-related evaluations were carried out. The first one of them was the *Language Independent Named Entity Recognition* (LINER) shared task, organized within the CoNLL conference in 2002 and 2003. The task required the identification of non-nested proper named entities in Spanish, Dutch, English and German, and was more ML- than IE-oriented.

The second one was the 2005 Pascal Challenge *Machine Learning for Information Extraction*, organized by the Dot.Kom European project. Several tasks were defined according to the conditions in which the learning was performed, but they all required slot filling from semi-structured English documents containing calls for workshops. The detection of workshop events was implicit, in the sense that every document contained one and only one of them, and of a single type (the *one-per-document* assumption, usual in this and related domains; Freitag, 1998; Califf, 1998). This evaluation, too, was more a testbed for ML algorithms than an actual IE task.

			IE														
	IR			Manual			ML				Inference						
	Query expansion	Trigger insertion	Coreference resolution	Shallow patterns	Syntactic patterns	Semantic rules	Supervised learning	Distant supervision	Bootstrapping	Question answering	Manual constraints	DB validation	Re-classification	Scoring	Re-ranking	Manual revision	Probabilistic inf.
(Byrne and Dunnion, 2010)	●	·	·	●	·	·	·	·	·	●	·	·	·	●	·	·	·
(Castelli et al., 2010)	●	·	●	·	·	●	●	·	·	·	·	·	●	●	·	·	·
(Chada et al., 2010)	●	·	●	·	●	●	·	·	·	·	·	·	·	·	·	●[1]	·
(Chen et al., 2010)	●	·	●	·	·	·	●	●	·	●	●	●	·	·	●	·	●
(Chrupała et al., 2010)	·	●	●	·	·	·	·	●	·	·	·	·	·	●	·	·	·
(Gao et al., 2010)	●	·	·	●	·	·	●	·	·	·	·	·	·	●	·	·	·
(Grishman and Min, 2010)	·	·	●	·	●	·	·	·	●[2]	·	·	·	·	●	·	·	·
(Intxaurrondo et al., 2010)	·	·	·	·	·	·	·	●	·	·	·	·	·	●	·	·	·
(Lehmann et al., 2010)	●	·	●	·	·	●	●	·	·	●	·	●	·	●	·	·	·
(Nemeskey et al., 2010)	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·
(Song et al., 2010)	●	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·
(Surdeanu et al., 2010)	·	●	·	·	·	·	·	●	·	·	·	·	·	●	·	·	·
(Varma et al., 2010)	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·
(Yu et al., 2010)	●	●	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·

[1] Only for the *Surprise Slot-Filling* task.

[2] With manual post-edition.

Table 2.1: Systems taking part in TAC-KBP 2010

2.1.3 Text Analysis Conference Era

In 2008, NIST decided to merge three evaluations, which up to that moment had been being held separately, under the common umbrella of a new Text Analysis Conference (TAC): the *Question Answering* (QA) track from TREC, the summarization exercises from DUC, and the *Recognizing Textual Entailment* (RTE) challenge, which had been organized since 2004 by the Pascal Network. In 2009, the QA task was replaced by *Knowledge Base Population* (KBP), which was more IE-oriented and hence absorbed the ACE evaluations.

KBP combines in a single task elements that had been considered in isolation in previous evaluations. The goal is expanding a given reference knowledge base, in which (possibly incomplete) information about a number of entities is already present. This requires linking entities present in a document collection to the ones in the base, as well as merging additional attributes and relations that can be extracted from the documents. Wikipedia infoboxes have been used in all evaluations so far as reference knowledge base—even if exact compliance with Wikipedia is not intended.

The task thus presents elements of entity, time, value, relation and event detection, as well as cross-document coreference resolution. Additional complexities arise from the need to combine information coming from several extractions, which may contain contradictions or redundancies with one other or with the reference knowledge base. We can hence consider systems participating in the KBP task as the cutting edge of IE technology.

Table 2.1 contains an overview of the systems that took part in the KBP evaluation in year 2010, focusing on whether they use (•) or not (·) of a number of different technologies. As we can see, the contending systems are in most cases fairly complex, and integrate components and methods from areas as diverse as:

	Prc	Rec	F1		Prc	Rec	F1
(Chada et al., 2010)	66.8	64.8	65.8	(Gao et al., 2010)	14.0	14.4	14.2
Human	70.1	54.1	61.1	Median	21.4	10.5	14.1
(Surdeanu et al., 2010)	54.0	59.6	56.7	(Chrupała et al., 2010)	20.1	10.3	13.7
(Chen et al., 2010)	28.0	44.3	34.3	(Varma et al., 2010)	36.3	5.4	9.4
(Byrne and Dunnion, 2010)	66.6	18.7	29.2	(Nemeskey et al., 2010)	4.1	4.7	4.4
(Castelli et al., 2010)	31.0	25.9	28.2	(Song et al., 2010)	1.3	1.4	1.4
(Lehmann et al., 2010)	44.9	19.4	27.1	(Intxaurrondo et al., 2010)	4.6	0.5	0.9
(Grishman and Min, 2010)	28.0	26.0	27.0	(Yu et al., 2010)	0.3	2.4	0.5

Table 2.2: Performance of systems presented at TAC-KBP 2010

- *information retrieval*, including query expansion and trigger word insertion;
- *information extraction*, including:
  - *manually built patterns*, of shallow, syntactic and semantic nature,
  - *automatically learned patterns*, acquired using supervised or unsupervised (distant supervision, bootstrapping) approaches;
- *inference*, including validation by external databases, re-ranking and probabilistic reasoning.

Table 2.2 contains a summary of the results obtained by the presented systems, in terms of precision, recall and F1 score (see Section 2.5). For each one, the run with the highest F1 score is reported. As can be seen, although the top-ranking systems significantly stand out from the rest (even exceeding the reported human performance on the task), the majority of the F1 scores lie in the 10-30% range. Until the proceedings of the last KBP 2011 evaluation become available, these figures give a measure of both the difficulty of present-time IE tasks, and of the power of the systems which are attempting to solve them.

Finally, despite not being IE-oriented, in these last years two evaluations on the related task of coreference resolution have been performed: the 2010 SemEval-2/Senseval-5 *Coreference Resolution in Multiple Languages* (COML) task, which required the detection of full coreference chains between named entities, pronouns and full noun phrases in Catalan, Dutch, English, German, Italian and Spanish; and the CoNLL-2011 shared task on *Modelling Unrestricted Coreference in OntoNotes* (MUCO), which presented a similar challenge, but in English only, and on the OntoNotes corpus.

At the view of all this IE research frenzy, we believe that, well into the second decade of the 21st century, the field seems to be in really good shape: new challenges are appearing every year, and its research community is more active than ever—and willing to take them.

2.2 Weakly Supervised Relation Extraction

The first IE systems that took part in the MUC evaluations, and still many of them nowadays, had their knowledge hand-coded by human experts. As mentioned in Section 1.2, this supposes a drawback on the portability of IE systems to other languages, domains and writing styles.

To reduce this cost, two different strategies have been considered: the development of supporting tools to aid the human experts in the task of adaptation (e.g., Yangarber and Grishman, 1997), and the use of machine learning techniques to acquire the required knowledge. We will focus here on the second approach. Research on ML for IE has been encouraged by the success of corpus-based approached in other NLP tasks (Young and Bloothoft, 1997; Manning and Schütze, 1999). A number of surveys of the application of ML methods to IE can be found in the literature (Cardie, 1997; Yangarber and Grishman, 2000; Turmo et al., 2006).

Most of the approaches proposed so far have been based on supervised learning methods. In them, the learning process requires a corpus where relevant concepts are annotated. The applicability of these approaches depends on the availability of such resources; and in order to stimulate research on IE, the organizers of evaluations like MUC or ACE would provide this kind of corpora to the groups taking part in them. Supervision can also be online, if the system requests examples to the user as the learning progresses (e.g., within the active learning framework).

However, this dependence on the examples to learn from, in turn, constitutes a bottleneck of supervised pattern learning methods, as annotating a corpus requires a significant human effort. Weakly supervised approaches seek to lower the costs of learning by reducing the supervision required from the user. In this way, the overall cost is also reduced. Additionally, most of these approaches can benefit from completely unannotated data. In a world in which the availability of large amounts of textual information is unlimited in practice, these approaches are extremely attractive for adaptive IE.

Next Sections 2.2.1 to 2.2.4 give a brief exposition of some among these weakly supervised approaches, roughly grouped by the kind of strategies they use. A summary of all such systems is presented in Table 2.3. Given the aim of this thesis, we will focus on relation extraction systems, but weakly supervised approaches have been proposed for other related tasks such as entity extraction (Collins and Singer, 1999; Riloff and Jones, 1999; Yangarber et al., 2002; Davidov and Rappoport, 2006), event extraction (Basili et al., 2000; Harabagiu and Maiorano, 2000), or semantic relation extraction[1] (Chklovski and Pantel, 2004; Turney, 2006), just to name a few.

2.2.1 Statistical Approaches

A number of approaches for weakly supervised relation extraction can be considered *statistical* in the sense that they gather diverse statistics from a corpus and determine the related entities and the relevant patterns from the collected information. They often perform a single pass over that dataset, a property that can make them more scalable to large collections than other approaches. However, it also limits the range of techniques they can incorporate.

AutoSlogTS The first IE pattern learning approach not to require complete supervision was AutoSlogTS (Riloff, 1996). It did require however, the classification of all documents in the training set as either relevant or non-relevant to the extraction task. AutoSlogTS follows a two-step scheme: in a first pass, one or multiple patterns are generated for every noun phrase in the corpus. The kind of patterns that are generated is determined by a set of *meta-patterns*, manually built to capture usual grammatical constructions. In a second pass, frequency statistics are gathered for each generated pattern. They are used to determine their *relevance rate*, found as the ratio between the number of occurrences of the pattern in the relevant documents and those in the overall collection. The relevance rate is scaled by the logarithm of the absolute frequency of the pattern, and the highest ranked ones according to this score are then reviewed and annotated, for use within the CIRCUS IE system (Lehnert, 1991).

DIRT The system proposed by Lin and Pantel (2001), *Discovery of Inference Rules from Text* (DIRT), takes as input a syntactic path expressing a relation between two entities, and produces a set of paths which express the same relation—in most cases, paraphrases of the original one. The assumption behind it is what the authors call the *extended distributional hypothesis*[2]: "*If two paths tend to occur in similar contexts, the meanings of the paths tend to be similar*". The system works by finding all paths in a corpus which satisfy a small set of constraints, and ranking them according to the mutual information between the distributions of the words in their slots and of those of the original path. A number of heuristics are incorporated to further prune the search space.

Sudo et al. 2003, ODIE Sudo et al. (2003) describe an approach in which the user provides the system a set of narrative sentences describing an scenario, and the system gives as output a set of extraction patterns. From the scenario description, a query is constructed for an IR engine. After syntactic parsing of the retrieved documents, all syntactic subtrees from the parse trees therein are indexed using a frequent tree mining algorithm (Abe et al., 2002), and sorted according to their tf-idf scores. Finally, the highest ranked ones are taken as extraction patterns.

[1] Extraction of relations between concepts in an ontology instead of relations between entities.

[2] As a generalization of the original *distributional hypothesis* of Harris (1954).

	Strategy								Supervision												Patterns									Evaluation					
	Statistical	Clustering	Bootstrapping	Co-training	Counter-training	Label propagation	Multiple instance learn.	Distant supervision	Meta-patterns	Document relevance	Query	Seed documents	Seed patterns	Seed entities	Seed tuples	Seed contexts	Entity types	Trigger words	Wikipedia pages	Relation database	Word sequence	Bag of words	POS sequence	Parse dependencies	Parse link chain	Parse subtree	Semantic graph	Complex feature sets	+ Named entities	Pattern inspection	Rel. mention extraction	Relation classification	Relation extraction	Text filtering	Entity extraction
AutoSlog TS (Riloff, 1996)	●	·	·	·	·	·	·	·	●	●	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	●[1]	·	·	·	·
DIPRE (Brin, 1998)	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·
Snowball (Agichtein and Gravano, 2000)	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	●	·	·	·	·	·	·	●	·	·	·	●	·	·
ExDisco (Yangarber et al., 2000)	●	·	●	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	●[2]	·	·	·	·	●	·	●[1]	·	·	●	·
DIRT (Lin and Pantel, 2001)	●	·	·	·	·	·	·	·	●[3]	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	●	·	·	·	·	·
(Sudo et al., 2003)	·	●	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	●	·	·	·	·	·	●
(Yangarber, 2003)	·	·	●	·	●	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	●[2]	·	·	·	·	●	·	·	·	·	●	·
KnowItAll (Etzioni et al., 2004)	●	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	●	·	·	●	·	·	·	·	·	·	·	●[4]	·	·	·	●	·	·
(Hasegawa et al., 2004)	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	●	·	·	·	●	·	·
(Stevenson, 2004)	·	·	●	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	●[2]	·	·	·	·	●	·	·	·	·	●	·
(Zhang, 2004b)	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●[5]	●	·	·	●	·	·	·
(Chen et al., 2005)	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	●	·	·	·	·	·	·	●	·	·	●	·	·	·
(Stevenson and Greenwood, 2005)	·	·	●	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	●[2]	·	·	·	·	●	·	·	·	·	●	·
(Zhang et al., 2005)	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	●	·	·	·	●	·	·
(Chen et al., 2006)	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	●	·	·	●	·	·	·
(Greenwood and Stevenson, 2006)	·	·	●	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	●	·	·	·	·	●	·
(Hassan et al., 2006)	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	●	·	·	·	·	·	●	·	●	·	·	·	·
Tplex (McLernon and Kushmerick, 2006)	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	●	·	·	·	·	·	·	·	·	·	●[6]	·	·	·	·
URES (Rosenfeld and Feldman, 2006)	●	·	·	·	·	·	·	·	·	·	●	·	·	·	●	·	·	·	·	·	●	·	·	·	·	·	·	·	●[7]	·	·	·	●	·	·
ODIE (Sekine, 2006)	·	●	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	●	·	·	·	●	·	·
(Shinyama and Sekine, 2006)	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	●	·	·	·	●	·	·
(Surdeanu et al., 2006)	·	·	●	●	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●[2]	·	·	·	·	●	●	·	·	·	●	·
TextRunner (Banko et al., 2007)	●	·	·	·	·	·	·	·	●[3]	·	·	·	·	·	·	·	·	·	·	·	●[8]	·	●[8]	·	·	·	·	·	·	·	·	·	●	·	·
(Bunescu and Mooney, 2007)	·	·	·	·	·	·	●	·	·	·	·	·	·	·	●[9]	·	·	·	·	·	●[5]	·	·	·	·	·	·	·	●	·	·	·	●	·	·
URIES (Rosenfeld and Feldman, 2007)	●	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	●	·	·
(Andrews and Ramakrishnan, 2008)	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	●	·	·	·	·	·	·	·	●[4]	●	●	·	·	·	·
O-CRF (Banko and Etzioni, 2008)	●	·	·	·	·	·	·	·	●[3]	·	·	·	·	·	·	·	·	·	·	·	●[10]	·	●[10]	·	·	·	·	·	·	·	·	●	●	·	·
IDEX (Eichler et al., 2008)	·	●	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	●	·	·	·	●	·	·
(Zhou et al., 2008)	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	●	·	·	●	·	·	·
(Mintz et al., 2009)	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	●[11]	●	·	·	·	●	·	·
(Qian et al., 2009)	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●[5]	●	·	·	●	·	·	·
(Yan et al., 2009)	·	●	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	·	●	·	●	·	·	·	·	●	·	·	●	●	·	·	·	·	·
(Qian and Zhou, 2010)	·	·	●	·	·	·	·	·	·	·	·	·	·	·	·	●	·	·	·	·	·	·	·	·	·	·	·	●[5]	●	·	·	●	·	·	·
(Yao et al., 2011)	●	·	·	·	·	·	·	·	●[3]	·	·	·	·	·	·	·	·	·	·	·	●[12]	·	●[12]	·	●[12]	·	·	·	●	·	·	·	●	·	·

Table 2.3: Weakly supervised systems for relation extraction

[1] After manual annotation [2] Verbal predicate-argument tuples [3] Internal to the algorithm [4] Using domain-specific classes found in a weakly supervised manner [5] Used within a hyperplane separator (SVM) [6] With one-per-document assumption [7] Optionally [8] Used within a naive Bayes classifier [9] Both positive and negative [10] Used within a CRF [11] Used within a multiclass logistic classifier [12] Used within a generative model

Sekine (2006) extends this approach by incorporating information from a paraphrase database built offline (Sekine, 2005). In the resulting *On-Demand Information Extraction* (ODIE) system, the found extraction patterns are merged into pattern sets, according to whether they have been found to be paraphrases of each other. After generation, these patterns are applied again on the corpus and the extracted entity pairs for each pattern set are integrated into tables, for later human inspection or automated use.

KnowItAll Etzioni et al. (2004) present a detailed description and evaluation of *KnowItAll*, a complex system which is mainly devised for extraction of *facts* (lists of entities belonging to a certain class), but which is also capable of extracting binary or n-ary relations. The input of KnowItAll is the *information focus*: a description of the extraction task which contains, for the target relations and the types of the entities involved, a number of keywords which may act as triggers for them.

In a first phase, these keywords are applied within a number of domain-independent templates to obtain *extraction patterns* (used to locate the candidates within sentences) and *assessment patterns* (used to query search engines and validate the extracted facts). The best assessment patterns for each pair are determined using a *bootstrapping* loop (see Section 2.2.3), and used as features for a naive Bayes classifier.

After the bootstrapping phase, the obtained extraction patterns are applied on the corpus sentences to obtain entity candidates, and the naive Bayes classifier is used to discriminate these candidates and reject spurious extractions. After the entities are extracted, the same process is repeated to find the relations existing between them.

A number of extensions are proposed to this basic scheme (*pattern learning*, *subclass extraction* and *list extraction*), but they deal mainly with the entity extraction task and will hence be omitted here for brevity.

Hassan et al. 2006 Starting with a pair of entity types provided by the user, the first step of the approach of Hassan et al. (2006) is to gather POS n-gram frequencies from a corpus, and to generate a Markov model from these statistics. Among all the POS sequences up to a certain length and which contain the two demanded types, those which are assigned a highest probability by the model are selected as patterns.

After the patterns are applied to the corpus and the matched entity tuples are collected, assessment of the confidence of both patterns and tuples is found using the HITS algorithm (Kleinberg, 1999). Optionally, to improve the matching process, the entity tuples can be previously clustered, using the Markov cluster algorithm (van Dongen, 2000).

URES The proposal of Rosenfeld and Feldman (2006) is *Unsupervised Relation Extraction System* (URES), a system which takes as input a description and a set of seed tuples for the target relation[3].

In a first step, the description is used to generate a query for an IR engine. The seed tuples are then searched across the retrieved documents, and the contexts in which they co-occur are taken as positive examples for the pattern learner. Positive examples for other relations are taken as negative examples, as well as pairs formed with other entities which co-occur in the same sentence with a seed tuple. The patterns are generated by generalization of all pairs of positive contexts, using optimal string alignment, and the negative contexts are then used to assess their precision. Patterns whose precision falls below a certain threshold are removed.

The found patterns are then applied on all the sentences in the corpus, using one of two extraction engines: a simple one which uses a shallow parser to determine the possible entities[4], and a full-fledged generic one (TEG; Rosenfeld et al., 2004).

TextRunner, O-CRF Banko et al. (2007) start by identifying heterogeneity and efficiency as the main problems to deal with by unsupervised IE systems processing large corpora, such as the Web, and propose *TextRunner*. In order to deal with the former, no named entity recognition

[3] To reduce supervision, the authors suggest the use of a high-precision and low-recall unsupervised IE system, specifically KnowItAll, to provide the seeds. In that case, the supervision requirements become those of the seeding system.

[4] The use of named entity recognition within this engine is studied by Feldman and Rosenfeld (2006).

component is incorporated, and all noun phrases are considered as entities instead. To achieve the latter, the algorithm performs the extraction task in a single pass over the data corpus.

In a first phase, TextRunner takes a small sample of the input corpus and performs *self-supervised learning*: the sentences therein are parsed, and for every pair of co-occurring entities, they are considered related if the parse tree containing them satisfies a number of predefined conditions, and unrelated otherwise. The dataset thus generated is used to train a naive Bayes classifier. However, to avoid dependency on a parser, which are known to suffer from brittleness over heterogeneous document collections, the model uses form- and POS-based features only.

After the classifier is learned, extraction is performed on the whole corpus, collecting all entity pairs which are judged related by it. To increase recall, the contexts are simplified using chunk-based rules (e.g., removal of adverbial and prepositional phrases). The relatedness of the collected entity pairs is finally assessed using a probabilistic redundancy model (Downey et al., 2005), and those considered related are indexed to allow for future user queries.

Later, Banko and Etzioni (2008) proposed *O-CRF*, which follows the same scheme but replaces the naive Bayes classifier by a conditional random field, and uses Resolver (Yates and Etzioni, 2007) to find relation synonyms. The work also contains a proposal for combination of unsupervised and supervised relation extraction learners, using stacking (Wolpert, 1992).

2.2.2 Clustering Approaches

Given the unsupervised nature of the clustering task, clustering techniques are a natural choice as the backbone of weakly supervised relation extraction approaches. The main assumption behind these approaches is that entity pairs with similar syntactic contexts will be similarly related, so groups (clusters) of similar contexts (patterns) will contain pairs bound by the same relation. However, the determination of a suitable similarity metric is crucial to the performance of these approaches, and their larger computational requirements with respect to the previously presented simpler statistical methods can become an issue when facing large document collections.

Hasegawa et al. 2004, Zhang et al. 2005 In the work of Hasegawa et al. (2004), the authors present a purely clustering-based approach for the construction of tables of related entity pairs in a corpus. After named entity recognition, all pairs of co-occurring entities are collected, and the intervening words in all these co-occurrences are merged into a single context vector for each pair. The entity pairs are clustered with the hierarchical agglomerative clustering algorithm (HAC; Murty and Krishna, 1980), using the cosine distance, the complete-link criterion, and a manually set merging threshold. Small clusters are discarded from the resulting clustering, and each one among the remaining ones is labelled with the most frequent context words in it, as an indicator of the relation existing between its entity pairs.

The approach of Zhang et al. (2005) follows the same scheme with minor changes. It uses the minimum spanning parse trees between the entity pairs instead of a bag-of-words vector to represent their context, and a tree kernel is defined to replace the cosine in the similarity calculations. Group-average is used instead of complete-link within the agglomerative algorithm, and the most frequent root of the context parse trees is used as relation label.

Chen et al. 2005 The approach proposed by Chen et al. (2005) requires two entity types as input. All pairs of entities belonging to one type each and co-occurring in the same sentence are collected. Similarly to previous work, the words appearing in their contexts are used to generate a dataset of bag-of-words vectors, which are then clustered using k-means. However, this approach uses a joint strategy, based on resampling, to determine both the optimal feature subset and number of clusters (Levine and Domany, 2001). After clustering, discriminative category matching (Fung et al., 2002) is used to determine a relation name from each one of the clusters.

Unrestricted Relation Discovery Shinyama and Sekine (2006) present *Unrestricted Relation Discovery*, a more complex approach in which clustering is used at two different levels, and which tries to solve the task of *preemptive information extraction*, i.e.: "*[the creation] of all feasible IE systems in advance without human intervention*".

The system is tailored for news articles. After web crawling across a number of online news providers, the collected articles are clustered to form *basic clusters*, each one hopefully containing articles related to a single event. This clustering is performed using tf-idf scores of the bag-of-words in each document, cosine similarity, and an incremental clustering algorithm (similar to Nearest Neighbour Clustering; Lu and Fu, 1978). The named entities present across the basic clusters are linked using single- and cross-document coreference chains.

For each document, a semantic graph representation of its sentences is then built (GLARF; Meyers et al., 2001), and *basic patterns*, corresponding to subgraphs within it, are detected. Finally, all mappings between the entities of every two basic clusters are considered, and the similarity of the two clusters according to each mapping is calculated[5]. If the similarity exceeds a certain threshold, the basic clusters are merged to form *meta-clusters*. Finally, the entity tuples in each meta-cluster are collected in tables: each one of these tables will correspond to a single relation.

URIES Rozenfeld and Feldman (2006) and Rosenfeld and Feldman (2007) extend the previously presented URES approach, using clustering to provide the required seed tuples. This new approach is named *Unsupervised Relation Identification and Extraction* (URIES) by the authors.

Similarly to URES, in the *unsupervised relation identification* phase no named entity recognition is performed and instead all noun phrases which are judged to contain a proper noun are taken as entities. All contexts in which any two such entities co-occur are gathered. The Apriori association rule mining algorithm (Agrawal and Srikant, 1994) is used to obtain URES-style patterns to be used as features, and an entropy-based criterion (Dash et al., 2002) is used to perform selection among them.

Once the data is gathered and the feature set determined, HAC with single-link clustering and cosine similarity is used to obtain clusters[6] and the clusters whose size fall below a threshold are pruned. The remaining clusters can then be used as seeds for URES, which will proceed up to the obtainment of the extracted relations.

Andrews and Ramakrishnan 2008 The approach of Andrews and Ramakrishnan (2008) allows the simultaneous unsupervised detection of domain entities and relations from large corpora, starting from a small set of seed entities. Focusing of the relation detection phase, their procedure starts by gathering all contexts in which two entities co-occur, representing them as a tf-idf bag-of-words. This context dataset is clustered according to the cosine similarity between them, using the affinity propagation algorithm (Frey and Dueck, 2007). Clusters which are deemed too small, or whose elements correspond mostly to contexts of a single entity pair are pruned as irrelevant. The remaining ones will correspond to relations present in the domain.

IDEX Eichler et al. (2008) proposes another clustering-based approach, *IDEX*, whose input is an IR-style query. After retrieval, the sentences in the obtained documents which contain at least two entities are parsed, and for each entity pair, its *parse skeleton* (parse link chain) is found. As the work focuses only on verb relations, only those skeletons in which the root is a verb and either the subject or the object is an entity are considered.

The obtained skeletons are clustered using the single-pass leaders algorithm (Hartigan, 1975), and a combination of parse, coreference and semantic (via WordNet) information as similarity measure. As usual, the resulting clusters correspond to different relations present in the corpus.

Yan et al. 2009 A recent system which benefits from the special features of Wikipedia with respect to other Web sources used as a corpus (much cleaner text, and heavy cross-linking between documents; Giles, 2005; Gabrilovich and Markovitch, 2006) is that of Yan et al. (2009). Starting from a set of Wikipedia article texts (e.g. all those belonging to a certain category) the sentences are scanned for co-occurrences of a reference of the entitled concept and an outgoing link. The entities referenced by the links are likely to share a relation with the one described in the article.

For each one of these *concept pairs*, additional documents from the Web are obtained using a search engine, and the *relational terms* (those which express the relation between the two entities)

[5] The similarity function is complex, and details of its calculation can be found in the original paper. For our purposes, it suffices to say that it involves finding the similarity between entities mapped to each other, as a function of the basic patterns they appear on; as well as the similarity between the documents, as a function of the words they contain.

[6] The authors also experimented with other linkage policies, as well as with k-means, but obtained the best results with single-link.

are detected using an entropy-based approach (Chen et al., 2005). The *surface patterns* (sequences of words containing the related terms, relational terms and functional words) in these Web documents are then collected. With regard to the original Wikipedia articles, *dependency patterns* are generated from the parse trees of the sentences which contain the related entity pairs, using a frequent tree mining algorithm (Zaki, 2002).

The clustering of the concept pairs is then performed in two steps. First, they are clustered according to the dependency patterns they share[7], using a k-means-style algorithm enhanced with a similarity threshold, which provides good precision clusters. To improve recall, the same algorithm is applied to refine the existing clustering, but using similarity between the surface patterns in the pairs[8]. Each cluster will now contain concept pairs from a single relation, which is labeled using its centroid.

Yao et al. 2011 The work of Yao et al. (2011) proposes a triad of generative models of increasing complexity, similar to latent Dirichlet allocation (LDA; Blei et al., 2003). In them, the occurrence of a relation within a document is regarded as a generative process in which the relation type, the entity types (optionally), and the linguistic features are successively drawn from conditional multinomial distributions. The model parameters and the values of the hidden variables (relation and entity types) are estimated from the observed data using expectation-maximization (EM; Dempster et al., 1977), with Dirichlet distributions as priors for all multinomials. After estimation, the inferred relation type variable is used to obtain clusters, which will contain entity pairs related by the same relation.

The features used in the model include the dependency path between the two entities and the form of the entities themselves—as well as their types, the forms and POS of the words in the context, and the presence of trigger words.

2.2.3 Bootstrapping Approaches

Bootstrapping (Yarowsky, 1995; Abney, 2004) is a meta-algorithm which uses a supervised learning algorithm (the *base learner*), a reduced set of labeled data, and a (usually much larger) set of unlabeled data. It is an iterative procedure in which, at each step, the base learner is trained using the labeled data, and the learned model is used to classify the unlabeled data. Those unlabeled samples which are classified with the highest confidence are then added to the labeled dataset, and the process is repeated.

A popular variation on this scheme is *co-training* (Blum and Mitchell, 1998), in which it is assumed that two different *views* (e.g., representations using different features) of the dataset are available, and two base learners are used, each one considering only one of the views.

The task of weakly supervised relation extraction fits quite naturally into this framework, and a significant number of approaches have hence appeared which apply bootstrapping techniques. However, even if bootstrapping is popular, it presents some drawbacks: for instance, the use of small seed sets implies that these seeds introduce a large bias on the learning process, and the lack of a stopping criterion means that the results often start degrading after a number of iterations.

DIPRE, Snowball Brin (1998) presents one of the first bootstrapping-based approaches for relation extraction. His *Dual Iterative Pattern Expansion* (DIPRE) is a system for extraction from Web documents which is based on the *pattern-relation duality*: "*given a good set of patterns, we can build a good set of tuples [...] given a good set of tuples, we can build a good set of patterns*". Starting from a small number of sample tuples belonging to the target relation, all their occurrences in the collection are found. Patterns are then acquired by simple generalization from the occurrence contexts, and applied again on the data. The occurrences of the tuples extracted by the newly found patterns can then be used to find new patterns, in an iterative fashion, until desired.

Agichtein and Gravano (2000)'s *Snowball* is an extension to DIPRE which incorporates a number of enhancements over the previous work. First, a Named Entity tagger is incorporated, and the named entity class of the two related entities is included in the patterns to improve their precision. Second, the generalization process is performed using a single-pass clustering algorithm. As a result, the contexts in the patterns are the centroids of these clusters, instead of being exact

[7] More precisely, the cosine distance between the bag-of-patterns representation of each pair is used.

[8] In this case, the sum of Levenshtein distances in a minimal matching of the two surface pattern sets is used.

sequences of tokens, and the matching of patterns and document text is performed by calculating and thresholding the similarity between the two. Lastly, the confidence of patterns and tuples is estimated in a recursive way (that of the former determined that of the latter, and vice-versa), and smoothed across the iterations of the algorithm.

ExDisco Yangarber et al. (2000) propose another bootstrapping approach. *ExDisco* starts from a number of seed patterns. The patterns induce a *split* on the document collection: those documents which contain at least one pattern will be deemed *relevant* to the relation, whereas those who do not will be *non-relevant*. All sentences in the corpus having been converted to patterns, these patterns are ranked according to how much their distribution across the collection correlates with the relevance of the documents. The highest ranked patterns are then selected, and the process can be iterated, starting with a new split.

Yangarber (2003) presents an enhancement over this basic scheme, with the goal of removing one of its main drawbacks, namely, the terminating condition. To achieve it, not only one but a number of different relations are learned at the same time, using ExDisco. The tuples learned by the other (competing) learners provide negative evidence for the target relation, and can hence be used to estimate the precision of the patterns.

Yangarber (2003) coins the term *counter-training* to denote this learning framework.

Zhang 2004b, Qian et al. 2009, Qian and Zhou 2010 The proposal for weakly supervised relation extraction of Zhang (2004b) is based on support vector machine bootstrapping. A comparison of three different approaches to achieve it is presented: *self-bootstrapping*, in which the whole input dataset is used; *bagging-based bootstrapping*, in which a number of datasets are generated from the input using bagging (Breiman, 1996); and *bootstrapping using random feature projection*, in which the datasets are generated by selecting a subset of the input features at random and projecting the input over these dimensions.

However, Zhang (2004b) only apply their method on the relation classification task, i.e., the entity pairs in the dataset are known to be related, and it is the nature of this relation which has to be established.

Qian et al. (2009) extend this approach using *stratified sampling*: the ratio of relation classes in the unlabeled samples which are incorporated at each iteration is constrained to be the same as that in the domain, yet the approach remains a relation classification, not a relation extraction, one.

However, knowledge of the distribution of all relation classes is not always available. In order to overcome this limitation, Qian and Zhou (2010) proposes to extend the previous approach, using clustering to partition the unlabeled data, and using clusters instead of classes as strata for sampling.

Stevenson 2004, Stevenson and Greenwood 2005, Greenwood and Stevenson 2006 The approach of Stevenson (2004) is based on semantic similarity between patterns: starting from a number of seed patterns, at every step the candidate patterns which are the most similar to the already learned ones are incorporated. Thresholds are used to discard too frequent and too rare patterns, and the similarity measure of Lin (1998) is used, applied over the WordNet hypernymy hierarchy.

Stevenson and Greenwood (2005) present a minor modification of the previous work, in which the similarity measure of Jiang and Conrath (1997) is used instead; and Greenwood and Stevenson (2006) also extend the approach, replacing the simple subject-verb-object patterns by dependency chains between the related entities. A modified version of the parse tree kernel of Culotta and Sorensen (2004) is used as similarity function.

Tplex McLernon and Kushmerick (2006) propose a system which differs from the other reviewed approaches in the fact that it is tailored for tasks with the aforementioned one-per-document assumption. As a result, *Tplex* detects text fragments which correspond to fields, and it uses patterns to detect fragment boundaries (i.e., their beginning and end) instead of the fragments themselves.

From a number of seed documents, in which these fragments have been annotated, the system generates a starting set of boundary patterns, which are generalized and applied on the unannotated documents. New patterns are obtained from the new boundaries therein detected, but they are not relaxed in order to avoid overgeneralization. After the patterns and boundaries are detected, their scores are assigned iteratively. Finally, the assembly of fragments from the detected boundaries is performed using beam search.

Surdeanu et al. 2006 The approach of Surdeanu et al. (2006) uses patterns to solve a task of document classification into different domains, with the assumption that those patterns which are relevant for classification will express relations particular to each one of the domains.

The acquisition is performed using co-training between a word-based naive Bayes probabilistic model (Nigam et al., 2000) and a pattern-based decision list (Yarowsky, 1995). Different criteria for the selection of the patterns into the list are explored. After convergence, the top-ranked patterns are selected.

2.2.4 Other Approaches

A number of the approaches proposed in the literature do not fit neatly into any of the aforementioned families, or even use completely different strategies. Some of the most relevant among these alternative schemes are described below.

Label Propagation Chen et al. (2006) propose a graph-based approach for weakly supervised relation extraction. Each entity pair co-occurring in the documents is mapped to a vertex of the graph, and edges are weighted according to the similarity between the contexts in which the pair occurs. Pairs which are given as seeds are assigned a label corresponding to their relation type, and a complete labelling is inferred using the label propagation algorithm (Zhu and Ghahramani, 2002). A number of different feature sets and similarity metrics for context representation are compared.

(Zhou et al., 2008) merges the ideas from this work and the previously presented bootstrapping using random feature projection of Zhang (2004b), to devise a scheme in which only those examples which have been deemed crucial by the bootstrapping procedure (i.e., those selected as support vectors) are used as labelled data for the label propagation step.

However, and also similarly to Zhang (2004b), both approaches only deal with the relation classification task.

Multiple Instance Learning Differing from other ML settings, in *multiple instance learning* (Dietterich et al., 1997) examples for each class come grouped in two sets or *bags*: a *positive bag*, which is ensured to contain *at least one* positive example, and a *negative bag*, which is ensured to contain only negative examples.

Bunescu and Mooney (2007) propose an unsupervised relation extraction approach within this framework. The system requires a set of input tuples, both positive and negative, and looks for sentences in which these entity pairs co-occur. Given a large enough input corpus, it is likely that there exists at least one sentence in which each positive seed pair occurs and the relation between them is expressed, so one can construct positive bags from then. Sentences which contain negative seed pairs can be used to build negative bags.

After the bags are built, the multiple instance learning problem is reduced to a standard supervised learning one (Ray and Craven, 2005), and solved with a support vector machine using a subsequence kernel (Bunescu and Mooney, 2005).

In order to reduce the burden of the user, the number of seed tuples should be low, and so will be the number of bags. This poses special problems in the context of multiple instance learning, and a word weighting scheme is required within the subsequence kernel to avoid biases.

Distant Supervision A more recent approach, which benefits from the growth of online freely-available structured databases (such as Freebase, or the Wikipedia infoboxes) is that proposed by Mintz et al. (2009), and which the authors name *distant supervision*.

For each pair of entities which appears in the database tables for the target relation types, all sentences in which the pair co-occur are collected. The context of each occurrence is represented using a window of words, POS and syntactic dependencies, and all contexts for a single pair are aggregated into one feature vector. Negative examples are generated by randomly taking pairs of entities which do not appear in any of the target relations.

The generated dataset is used to train an ML classifier, more specifically, a multinomial logistic regression model. To find relations in new data, the contexts in which all entity pairs in the new documents appear are collected. Each one of their aggregated contexts is then fed to the classifier, so as to determine the nature of their relation, if any.

Despite being supervised—albeit not in the usual sense—we have included this approach in our review as a number of groups (Chen et al., 2010; Chrupała et al., 2010; Intxaurrondo et al., 2010; Nemeskey et al., 2010; Song et al., 2010; Surdeanu et al., 2010) used different variations on it to participate in the TAC KBP 2010 evaluation (see Section 2.1.3).

2.3 Elements of Supervision

After the review of the presented approaches to weakly supervised relation extraction, it is clear that the elements of supervision vary significantly both in terms of nature and volume across them.

Language Bias One particular element which is implicitly present in all proposed approaches is the language bias imposed by the used pattern formalism (see next Section 2.4). The relations the different approaches will extract are only those whose grammatical condition can be captured by the considered patterns. A few approaches go one step further, and restrict the possible patterns using *meta-patterns* (i.e., patterns on the patterns; Riloff, 1996; Etzioni et al., 2004). However, even if these are the only works for which the meta-patterns are explicitly mentioned as user input, in many other cases restrictions on the patterns do exist, but are an internal part of the algorithm not available for change (e.g., Lin and Pantel, 2001; Banko et al., 2007; Yao et al., 2011).

Document Relevance An element of supervision that a number of approaches require to different degrees is the relevance of the documents in the corpus to the extraction scenario. Even if the first approaches to weakly supervised extraction needed the user to classify *all* documents of the collection as relevant or irrelevant (Riloff, 1996), this requirement was soon lifted, and now often only a user query, to be used with an IR system, is demanded (Sudo et al., 2003; Rosenfeld and Feldman, 2006; Sekine, 2006; Eichler et al., 2008). In other cases, it is tacitly assumed that the whole corpus is relevant to the task, as the system works in an exploratory fashion (e.g., Hasegawa et al., 2004; Hassan et al., 2006).

Seeds However, the most usual form of supervision for weakly supervised systems in this task is that of *seeds*: a small set of elements that are taken as prototypical example of their class. All bootstrapping approaches, by definition, require it, and some other statistical or clustering ones also do. However, the nature of the seeds differs from system to system—they may be documents (Surdeanu et al., 2006), patterns (e.g., Yangarber et al., 2000; Stevenson, 2004), entities (Andrews and Ramakrishnan, 2008), tuples (pairs of related entities; e.g., Brin, 1998; Agichtein and Gravano, 2000) or contexts (sentences in which the relation between a pair of related entities is expressed; e.g., Zhang, 2004b). In most of the cases, only positive seeds are needed, but a few approaches require both positive and negative seeds as input (Bunescu and Mooney, 2007).

Other Forms of Supervision Some approaches may only require a pair of entity types: the relations between entities belonging each to one of the types will be extracted (Chen et al., 2005; Hassan et al., 2006). Finally, it is common for approaches which are tailored for some subtask of relation extraction to require particular elements of supervision: systems requiring trigger words for the target relations (Etzioni et al., 2004), Wikipedia pages (Yan et al., 2009) or even a complete database of relations (Mintz et al., 2009) can be counted among them.

2.4 Pattern Formalisms

The kind of patterns used for relation extraction also varies across the different approaches. Figure 2.2 contains a sample sentence and sample patterns, using different pattern formalisms, that could capture the relation therein expressed.

*. . . when **Company Co.** appointed **John Doe** as president.*

(a) Sentence

<**X: ORG**> appointed <**Y: PER**> $*$ president .

(b) Word sequence (with wildcards)

$$\begin{pmatrix}\text{when: .05}\end{pmatrix} \ \langle\textbf{X: ORG}\rangle \begin{pmatrix}\text{appointed: .3}\\ \text{nominated: .7}\end{pmatrix} \langle\textbf{Y: PER}\rangle \begin{pmatrix}\text{as: .8}\\ \text{CEO: .3}\\ \text{president: .7}\end{pmatrix}$$

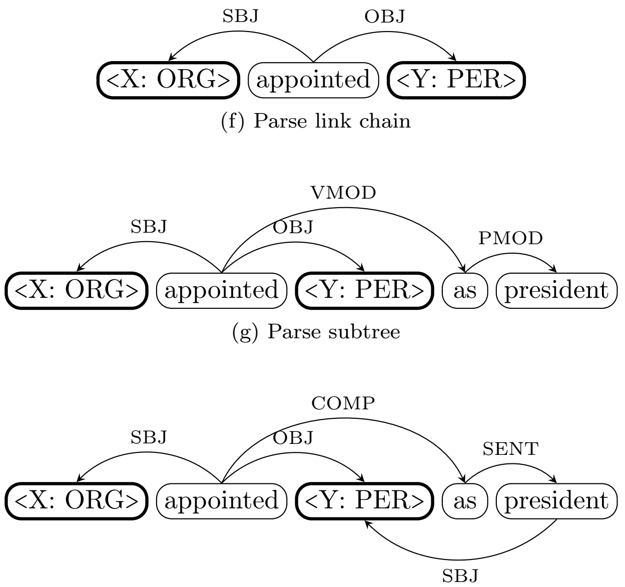
(c) Bag-of-words

<**X: ORG**> VBD <**Y: PER**> IN NN .

(d) POS sequence

(S: <**X: ORG**>, V: appoint, O: <**Y: PER**>, IO: president)

(e) Parse dependencies



(f) Parse link chain

(g) Parse subtree

(h) Semantic graph (GLARF)

Figure 2.2: Comparison of extraction pattern formalisms

Word Sequences One of the first and still most popular formalisms are *word sequence* patterns, in which the context of the entity pair is split into left, middle and right contexts (according to whether the words are found before the first entity in the pair, between the first one and the second one, or after the second one), and each partial context is matched against fixed sequences of word forms. A number of devices have been incorporated to this base formalism to increase its flexibility. We can name among them the use of optional words and wildcards (Rosenfeld and Feldman, 2006) or the replacement of a word by its semantic class (either automatically inferred or assigned by a named entity recognizer; Etzioni et al., 2004; Andrews and Ramakrishnan, 2008). In those approaches where the calculation of similarity between this kind of patterns is required, subsequence kernels (Bunescu and Mooney, 2007) or the Levenshtein distance (Yan et al., 2009) have been used.

A similar formalism is that of *bag-of-words*, in which the ordering of words within each partial context is lost, and the contexts for several occurrences may be aggregated. Each one of the partial contexts thus contains the relative frequencies of occurrence of several words. These bags-of-words are usually the result of generalization or, in particular, clustering of individual contexts (Agichtein and Gravano, 2000; Hasegawa et al., 2004).

A last formalism related to word sequences is that of *POS sequences*, in which the contexts are matched using fixed sequences of POS (Hassan et al., 2006).

Syntactic Patterns Even if deemed as unsuitable for use in heterogeneous domains such as the Web by some (e.g., Banko et al., 2007), a number of approaches use patterns based on the syntactic representation of contexts between entities. According to the amount of syntactic context of the individual occurrences that is preserved in the patterns, we can speak of different formalisms.

The simplest of them only preserves *parse dependencies*, i.e., the dependencies of certain fixed elements in the parse tree, which are used as anchor. A particular case of this are *predicate-argument* tuples (as introduced by Yangarber et al., 2000), consisting of a (verbal) predicate, together with the heads of its subject, object and/or prepositional complement (and hence also called, because of their subject-verb-object structure, *SVO patterns*). The main—and obvious—drawback of this kind of patterns is their inability to capture relations that are not expressed by means of the selected anchors.

Other syntactic formalisms allow for arbitrary subtrees of the parse tree to be matched against the pattern. However, we can distinguish between the *parse link chain* model (also called simply *path* or *chain* model; Lin and Pantel, 2001; Sudo et al., 2003), in which only the dependencies directly joining the two entities are considered, and the full *parse subtree* model, in which sibling nodes can be present and influence the matching process. A comparison of the performance of the three presented syntactic models is found in Sudo et al. (2003). The conclusion drawn by its authors is that the subtree model allows a gain in recall with respect to the other two, while preserving high precision.

A related formalism is that used in Shinyama and Sekine (2006), in which patterns are expressed according to a *semantic graph* representation of the sentences. More specifically, the GLARF framework (Meyers et al., 2001) is used in this work (see Section 2.2.2).

Complex Feature Sets At this point of the exposition, one may argue that the formalisms presented so far are all rather lightweight, and wonder why experimentation with complex feature sets—incorporating lexical, syntactic and even semantic information at the same level, and common in other NLP tasks (in particular, in supervised relation extraction; Kambhatla, 2004; Zhao and Grishman, 2005; Surdeanu and Ciaramita, 2007)—has not been thoroughly performed.

However, there is one factor that has to be taken into account in unsupervised learning settings in general, and it is that unsupervised methods can easily falls prey of the so-called *ugly duckling theorem* of Watanabe (1969): "*Insofar as we use a finite set of predicates that are capable of distinguishing any two objects considered, the number of predicates shared by any two such objects is constant, independent of the choice of objects*" (Watanabe, 1985). For the task of clustering, Jain et al. (1999) state that "*this implies that it is possible to make any two arbitrary patterns equally similar by encoding them with a sufficiently large number of features*". Without the bias provided by supervision, the incorporation of new features will not always cause an increase in performance– on the contrary, it may produce a significant drop. For this reason simpler representations may outperform more complex ones in unsupervised tasks.

As a result, so far only approaches which transform the problem into a supervised learning one, and use supervised learning methods such as support vector machines (Zhang, 2004b; Qian et al., 2009; Qian and Zhou, 2010) or logistic regression (Mintz et al., 2009), have used complex pattern feature sets similar to those used in supervised relation extraction.

Named Entities Finally, it is interesting to note that, even if standard named entity recognition methods have also been criticised for being non-portable to heterogeneous corpora (Downey et al., 2007), most weakly supervised relation extraction approaches incorporate named entity information in their patterns. Some systems use predefined named entity hierarchies, such as those of the MUC (e.g., Yangarber et al., 2000) or ACE evaluations (e.g., Zhang et al., 2005), or the one proposed by Sekine et al. (2002) (Sudo et al., 2003); whereas other use entity classes and instances also determined in a weakly supervised way (Etzioni et al., 2004; Andrews and Ramakrishnan, 2008).

2.5 Evaluation

The evaluation of weakly supervised relation extraction approaches, in particular those that acquire knowledge from large corpora such as the Web, presents some challenges of its own.

In this, as in other tasks, we can distinguish two main families of evaluation strategies. Some of them are directly related to the target task (i.e., relation extraction) and can hence be qualified as *direct* or *intrinsic* evaluations. Alternative *indirect* or *extrinsic* evaluations can also be considered, in which the acquired knowledge is incorporated in a system for a different task. In this second case, the quality of the extraction approach is measured by the performance of this later system, using the metrics proper of the task.

The next two Sections 2.5.1 and 2.5.2 describe some of the strategies in the direct and indirect families, respectively, which have been used so far to assess the quality of weakly supervised relation extraction approaches.

2.5.1 Direct or Intrinsic Evaluation

Pattern Evaluation Few approaches have chosen to carry out direct evaluations of the acquired patterns. This kind of evaluation is necessarily manual, and properties that human reviewers need to assess may include their *correctness* (Lin and Pantel, 2001) or their *relevance* (Riloff, 1996; Surdeanu et al., 2006). However, this is a costly process, which becomes impractical as the size of the learning corpora and the number of learned patterns increases, and the evaluation needs to be restricted to the highest ranked patterns or to random samples. In addition, the task of deciding whether a pattern is correct (cfr. relevant) or not for a given domain is not trivial, mainly due to the ambiguity of the patterns. Thus, the process must be carried by more than one judge, so that the judgements for ambiguous patterns can be agreed upon.

Relation Mention Extraction It is hence much more usual to evaluate the acquisition process by the output of the extraction phase. Nevertheless, a number of different tasks can be considered at this step.

One possibility is the evaluation of *relation mention extraction*, i.e., the determination of whether two entities, *in a certain context*, are related or not, and the nature of this *relation mention.* This task is the one which is most similar to that presented in IE evaluations, and hence the authors which choose to perform it usually use a subset of the MUC (Riloff, 1996; Yangarber et al., 2000) or ACE (Chen et al., 2006; Hassan et al., 2006; Zhou et al., 2008) corpora as evaluation data. These corpora provide a gold standard, and MUC-style metrics of *precision*, *recall* and *F1 score* (Chinchor, 1992) are commonly used in this context[9].

Given that this is the main evaluation scheme we will be following (see Sections 4.5.2 and 6.3.2), a number of results reported using it are shown in Table 2.4. Both weakly supervised and supervised approaches have been included. Nevertheless, it is important to note that the differences in evaluation conditions and corpora do not allow for any kind of comparison of the results across

[9] An alternative set of metrics was proposed for the ACE evaluations, but their complexity has hampered their spread, and they have seldom been used outside the competition.

multiple works. These figures are thus only included for information, and we refer to the original papers for details[10].

Beyond this setting, other approaches perform the extraction in domains where the *one-per-document* assumption holds (McLernon and Kushmerick, 2006), whereas a third group assume the mention detection step is solved, and focus on the *relation classification* phase only (e.g., Zhang, 2004b; Chen et al., 2006).

Relation Extraction A slightly different approach for evaluation is that of a *relation extraction* task. In this case, the aim is the determination of the relatedness of two entities *globally* (without reference to a certain sentence where that relation is expressed). This evaluation scheme is certainly popular, and there are several ways in which the extraction performance can be quantified. A common one is the comparison of the extracted pairs of related entities to a gold standard list—even if the construction of such lists is a problem by itself. For small or medium domains, manual lists can be collected on purpose (Hasegawa et al., 2004; Zhang et al., 2005), or derived from online resources (Wikipedia, IMDB, Freebase…; Agichtein and Gravano, 2000; Etzioni et al., 2004; Yao et al., 2011), and *precision/recall* metrics are used. In other cases, manual evaluation is required, and, given the potentially huge size of the extracted tuple sets, the evaluation can only be performed on a random subsample of the collection. Concepts such as *correctness* (Brin, 1998; Lin and Pantel, 2001; Sekine, 2006; Banko et al., 2007), *fitness to table* (Shinyama and Sekine, 2006) or *precision* (Rosenfeld and Feldman, 2006) have been used by human judges to assess the quality of the extractions. The number of obtained tuples has also been used as an indicator of quality (and even been improperly misnamed *recall*; Rozenfeld and Feldman, 2006).

2.5.2 Indirect or Extrinsic Evaluation

A common indirect approach to relation extraction evaluation is to use the detected patterns for text filtering, and evaluate the classification of the documents in a collection which is induced by the patterns, using standard classification metrics like *precision/recall* (Yangarber et al., 2000; Yangarber, 2003; Stevenson, 2004; Surdeanu et al., 2006). However, this kind of evaluation is assuming that the obtained patterns correspond to domain-specific entities, relations and events. Some approaches go beyond, and extend this document filtering approach to sentence filtering (Stevenson and Greenwood, 2005, using a version of the MUC-6 corpus which had been annotated with events at sentence level by Soderland, 1999).

Potentially, the use of indirect evaluations could open the door to uncountable and more or less ad hoc methods to assess the quality of the extraction process. However, among those which have been used in the approaches compared in this chapter, only the use of relation extraction patterns to detect entities involved in events in a certain domain (Sudo et al., 2003) remains to be mentioned.

[10] For reasons of brevity, we have chosen not to include results for any other among the presented approaches. Moreover, the differences in task, conditions, data, metrics and protocol would render their fair comparison impossible.

	Corpus	Prc	Rec	F1
AutoSlog TS (Riloff, 1996)	MUC-4/TST3	36	58	44
ExDisco (Yangarber et al., 2000)	MUC-6	72	52	60
(Hassan et al., 2006)	ACE 2004	47	58	52
TextRunner (Banko et al., 2007)[1]	B&M, 2007	86	23	36
(Andrews and Ramakrishnan, 2008)	AIMED	30	61	40
	CB	65	69	67
O-CRF (Banko and Etzioni, 2008)	B&M, 2007	88	45	60

[1] As reported by (Banko and Etzioni, 2008)

(a) Weakly supervised approaches

	Corpus	Prc	Rec	F1
AutoSlog (Riloff, 1993)	MUC-4/TST3	56	43	49
(Zelenko et al., 2003)	Newswire	91	83	87
(Kambhatla, 2004)[2]	ACE 2004	67	59	60
(Zhao and Grishman, 2005)	ACE	69	70	70
(Zhou et al., 2005)	ACE	63	49	55
(Culotta et al., 2006)	Wikipedia	71	54	61
(Surdeanu and Ciaramita, 2007)	ACE 2007	63	53	59
(Zhou et al., 2007)	ACE 2003	81	68	74
	ACE 2004	82	70	76

[2] As reported by (Hassan et al., 2006)

(b) Supervised approaches

Table 2.4: Published relation mention extraction results

3 Clustering

In one case out of a hundred a point is excessively discussed because it is obscure; in the ninety-nine remaining it is obscure because it is excessively discussed.

Edgar Allan Poe

This chapter presents our experiments on the task of clustering. Our work has focused on the problem of document clustering using unsupervised ensemble methods. More specifically, we have performed a comparison of two strategies for generation of the ensembles, both between them and against individual algorithms.

Section 3.1 introduces the problem of unsupervised clustering and motivates the use of ensemble methods. An overview of related work is presented in Section 3.2. Section 3.3 presents our formalization of the problem of clustering. Section 3.4 describes the compared approaches, and Section 3.5 presents a summary of the experiments performed and the results obtained with each one of the considered approaches. Lastly, Section 3.6 extracts conclusions from the evaluation.

Parts of this work are also described in (Gonzàlez and Turmo, 2005, 2008b,a).

Cluster analysis lies at the core of most unsupervised learning tasks. Tasoulis and Vrahatis (2004) define clustering as "*the process of partitioning a set of patterns into disjoint and homogeneous meaningful groups, called clusters*", and Jain et al. (1999) state that "*intuitively, patterns within a valid cluster are more similar to each other than they are to a pattern belonging to a different cluster*". In addition to *pattern*, each element to be clustered has also received the names of "*object, record, point, vector, [...] event, case, sample, observation, or entity*" (Tan et al., 2005, Ch. 2). To avoid confusion (for instance, with extraction *patterns*) we will stick to the term *object* thorough this and following chapters.

3.1 Unsupervised Ensemble Clustering

Research on clustering is active in the field of pattern recognition. Several surveys of clustering methods, whose reading also provides a historical perspective of the evolution of the field, have been successively elaborated by Dubes and Jain (1980), Jain et al. (1999) and Xu and Wunsch (2005).

Even if clustering is mainly an unsupervised task, elements of supervision remain in many clustering methods. The number of clusters k is often required to be provided by the user or by external sources. Additionally, they may depend on a set of starting conditions to which their output is sensitive.

A number of authors have aimed at reducing or completely removing these elements of supervision, and have thus emphasized the *unsupervised* in the titles of their works: we can find articles about *unsupervised cluster analysis* (Roberts, 1997), *unsupervised learning of models* (Figueiredo and Jain, 2002; Zivkovic and van der Heijden, 2004), *unsupervised classification* (Essaqote et al., 2005), or *unsupervised cluster discovery* (Sakai et al., 2007). In these works, *unsupervised clustering* is defined as the clustering task in which "*the 'optimal' number of partitions is unknown a priori*" (Roberts, 1997); and an *unsupervised clustering algorithm*, as one which "*is capable of selecting the number of components [(clusters)]*", and "*does not require careful initialization*" (Figueiredo and Jain, 2002). For the purpose of our discussion, we will speak of *supervised clustering* when referring to the task and the algorithms which do not fulfill this definition: in particular, to those situations in which the number of clusters is known in advance[1].

A common approach to unsupervised clustering is to repeatedly apply a supervised clustering algorithm, with different numbers of clusters and starting conditions, and then choose the best one among the obtained clusterings using a model selection criterion (Milligan and Cooper, 1985). Other approaches are able to automatically estimate the number of clusters in a preliminary or final phase, or to increase or decrease it during the clustering process itself (see Section 3.2.1).

However, each one of these methods has intrinsic and particular biases (implicit or explicit, and of selection, language or search type; Whigham, 1996), uses a certain data representation, and depends on a document similarity measure. All these assumptions guide the clustering process, and lead it to a particular solution that may not be the most suitable one for all datasets.

The limitations of individual algorithms have long since been identified in supervised learning scenarios too, and combination methods have been shown to outperform single-classifier approaches. One of the first works in that direction was that of Hansen and Salamon (1990), who, inspired by fault-tolerant software engineering methods (Eckhardt and Lee, 1985), proposed a *neural network ensemble* architecture, in which a number of feed-forward neural networks are trained to solve the same problem, and their outputs are combined for classification. Other popular approaches to combination of several classifiers trained on the same data (or subsamples of it) include boosting (Schapire, 1990) and bagging (Breiman, 1996); whereas Kittler et al. (1998) propose a theoretical framework for combination of learners which use different underlying representations of the data.

Spurred by the success of aggregation methods in supervised learning, recent research has produced a number of clustering combination approaches. From a general point of view, the problem of *clustering combination* can be defined as: "*Given multiple clusterings of the data set, find a combined clustering with better quality*" (Topchy et al., 2005). In the scenario where combination takes into account the clusterings only, without accessing the representation (or representations) of the data, it is usual to refer to the problem as *ensemble clustering* (Strehl and Ghosh, 2002).

Ensembles have been proven to outperform their individual components in terms of clustering quality and robustness, as well as to provide a useful framework for applications such as knowledge reuse, or distributed and privacy-preserving clustering (Strehl and Ghosh, 2002). A number of clustering combination algorithms have been proposed, as well as methods for generating the clustering ensembles to be later combined (see Section 3.2.2).

3.1.1 Document Clustering

Within the NLP area, document clustering is a basic task which can be useful both by itself or as a first step towards further linguistic processing. Besides, the task has often been used as a testbed for experimentation on clustering methods over linguistic data. A number of clustering algorithms have been applied to document datasets, and empirical comparisons between them have been published—such as those of Zhong and Ghosh (2005) on generative-model-based algorithms, and of Ghosh and Strehl (2006) on distance-based ones.

[1] Incidentally, some works have considered a *semi-supervised clustering* task, in which more supervision is incorporated, usually in the form of pairwise constraints between element which should or should not be clustered together (Grira et al., 2004). In this context, a different usage of *unsupervised clustering* exists, as the term is employed in contraposition to this semi-supervised clustering, and refers to all unconstrained clustering methods, without regard to their supervision or non-supervision in the previous sense. Given that semi-supervised clustering is outside the scope of this thesis, we will disregard this usage without danger of confusion, and henceforth stick to the most habitual meaning of *unsupervised clustering*.

However, given the availability of unsupervised clustering methods, both individual and ensemble-based, a comparison of the performance of different approaches when applied over document collections was required. More specifically, we considered that the following two questions remained unanswered:

- **How well do ensemble methods perform for unsupervised document clustering?** Unsupervised methods had not been tested thoroughly on document collections.
- **How well do different individual clustering strategies perform in the context of unsupervised ensemble document clustering?** The influence of the strategy used to find individual clusterings to be later combined had often been overlooked. Different strategies needed to be compared.

Our work on clustering attempts to answer both questions. We have evaluated non-parametric clustering algorithms on a variety of real-world document collections; and we have performed an empirical comparison of the effectiveness of two different strategies for the generation of clustering ensembles: one relying on massive randomization of a single algorithm, and another relying on few but heterogeneous different algorithms.

As will be seen later, the conclusions drawn at the light of the results of this evaluation significantly influenced the path of further research, especially in the task of minority clustering (see Chapter 5).

3.2 Related Work

Clustering algorithms have been traditionally split into two major families: that of *hierarchical* algorithms, which produce a sequence of partitions with hierarchically nested clusters (usually represented as a tree-like structure, the *dendrogram*, which has a single cluster with the whole dataset as root, and the set of singleton clusters for each object as leaves); and that of *partitional* algorithms, which produce a single partition

Hierarchical Clustering Hierarchical agglomerative clustering (HAC; Murty and Krishna, 1980) is by far the most popular hierarchical algorithm. In HAC, every object starts in a cluster of its own, and at every step the two closest clusters are merged. A number of different rules for updating the distances from this new cluster to the other ones exist; among the most used ones we can mention single-link (Sneath and Sokal, 1973), complete-link (King, 1967) and UPGMA (Sokal and Michener, 1958). Being supervised, a number of devices have been incorporated to allow HAC to detect the number of clusterings. Some of them are *stop rules*, i.e., they give a criterion to stop the cluster merging process. Other approaches obtain a dendrogram up to the single-cluster root, and the use of a selection criterion to determine the most suitable partition therein. A comparison of the performance of 30 such stop rules and criteria for HAC was carried out over synthetic datasets by Milligan and Cooper (1985).

Partitional Clustering Within partitional algorithms, we can also distinguish two families: those which provide a *hard clustering* of the data, in which each object belongs to a single cluster; and those which perform *soft clustering*, and allow for each object to belong, with a certain grade of membership, to several clusters. Because of its connection to fuzzy sets (Zadeh, 1965), soft clustering is often also named *fuzzy clustering.*

Among the most widely used partitional algorithms, we can mention the distance-based hard k-means (MacQueen, 1967) and its soft counterpart c-means (Dunn, 1973). The algorithm is iterative in nature, and successively refines a starting clustering by assigning each object to the cluster whose centroid is closest (resp., assigning a soft membership to each cluster inversely proportional to the distance to its centroid), and then updating the centroids to become the average (resp., the weighted average by grade of membership) of the objects in their cluster.

Expectation-maximization (EM; Dempster et al., 1977) is a general algorithm, also of iterative refinement nature, to fit probabilistic models using incomplete data which has also found large applicability as a natural method for model-based clustering. For instance, mixtures of Gaussian (Zhuang et al., 1996) or multinomial (Meilă and Heckerman, 2001) distributions have been successfully used for this task.

Banerjee et al. (2005) present a theoretical framework for clustering, based on Bregman divergences (Bregman, 1967). The authors derive two generic algorithms, one for hard clustering and another one for soft, and prove that several previously proposed ones, including k-means and EM with distributions from the exponential family, are particular cases of them.

3.2.1 Unsupervised Clustering

A number of strategies have been considered to remove the elements of supervision present in existing clustering algorithms—or to altogether devise new approaches without them. Some of the explored directions are listed below.

Model Selection As mentioned in the introduction, unsupervised clustering can be performed with supervised algorithms by generating multiple clusterings of the dataset, using different conditions, and choosing one among them according to some model selection criteria. We might name this the *generate-and-select* strategy.

This scheme has been applied to partitional clustering algorithms (e.g., k-means; Peña et al., 1999) and to hierarchical ones (e.g., HAC; Milligan and Cooper, 1985). For the latter, instead of generating multiple dendrograms, the selection is often performed among the cuts obtained at each level of the dendrogram.

A considerable number of selection criteria have been proposed, including probabilistic ones, such as minimum message length (MML; Boulton and Wallace, 1969), Akaike information criterion (AIC; Akaike, 1974), minimum description length (MDL; Rissanen, 1978), or Bayesian information criterion (BIC; Schwartz, 1978); and distance-based ones, including Caliński and Harabasz' index (Caliński and Harabasz, 1974), or Silhouette (Rousseeuw, 1987).

Direct Estimation Among the approaches which are able to estimate the number of clusters from properties of the data in advance to the clustering process itself, we can mention those of Girolami (2002) and Li et al. (2004a). In both cases, the eigenvalue decomposition of the kernel matrix of the dataset (containing the dot products of each pair of objects) is found: if the data naturally contain k clusters, the k largest eigenvalues of the matrix will be significantly larger than the rest. Hence, a significant drop in the magnitude of the sorted sequence of eigenvalues can be used to detect the value k.

On the flipside, in mean-shift clustering (Fukunaga and Hostetler, 1975), objects are iteratively shifted towards their cluster representatives, so after convergence the number of clusters is determined by the number of different representatives obtained. Another approach which also defers cluster number detection to the end of the process, support vector clustering (SVC; Ben-Hur et al., 2002), starts by finding a hypersphere in an implicit high-dimensional space (as defined by a kernel function) which includes the images of all objects in the dataset. Afterwards, a connectivity graph is found: a pair of objects is deemed connected if the image of the path between them is completely contained within this hypersphere. Clusters will correspond to connected components in this graph.

Hybrid Combination A number of authors have experimented with combined hierarchical and partitional clustering algorithms to obtain unsupervised methods. For instance, the method of Fraley and Raftery (1998) starts by applying HAC to the dataset. It then uses the successive partitions inferred by the top-level branches of the dendrogram as starting clusterings for EM, and selects one among them using BIC. Surdeanu et al. (2005) also propose the use of HAC to obtain an initial clustering for EM, but their method is based on detecting subtrees in the dendrogram which are particularly *tight* and can hence be used as *seeds* for EM. A similar idea had been previously presented by Pantel and Lin (2002)—in this case, the sets of closely-related objects from which to start the clustering receive the name of *committees*.

Model Updating Finally, as also mentioned before, algorithms which are able to increase or decrease the number of clusters during the clustering process itself have also been proposed. This is often done by incorporating parsimony terms to optimization-based clustering algorithms. For instance, Frigui and Krishnapuram (1999) add a regularization term to the c-means objective function, whereas Figueiredo and Jain (2002) incorporate the MML criterion within the inner loop (not only as a final step) of an EM algorithm for finite mixture models.

3.2.2 Ensemble Clustering

The application of ensemble clustering to practical problems requires the development both of strategies to generate the clusterings in the ensemble, and of algorithms to combine them into a final consensus clustering. During the last decade, a number of approaches, both supervised and unsupervised, have been proposed for each one of these two steps.

3.2.2.1 Combination Methods

The problem of clustering combination has been formulated using a number of different paradigms.

Co-Association Matrices One of the first approaches to clustering combination was that proposed by Fred and Jain (2002). Their *evidence accumulation* framework introduced the concept of *co-association matrix*, containing, for each pair of objects, the fraction of input clusterings which assign them to the same cluster. After the co-association matrix is constructed, the single-link HAC algorithm, with a merging threshold of 0.5, is applied to obtain the consensus clustering. Fern and Brodley (2003) present a modification of the previous work which allows soft clusterings in the input ensemble.

A number of later works have also used the concept of co-association matrix. For instance, Li et al. (2004b) derive coefficients for an integer lineal programming problem from whose solution a consensus clustering can be constructed; and Gionis et al. (2005) reformulate ensemble clustering in terms of correlation clustering over this same matrix—and give a triad of methods to solve the problem. Moreover, one of the three methods proposed by Nguyen and Caruana (2007) uses its entries as similarity metric for a version of k-means (equivalent to kernel k-means; Girolami, 2002). Finally, Li et al. (2007) gives a proof that ensemble clustering can be reduced to non-negative matrix factorization of the co-association matrix.

Reclustering A second common strategy is that of generating a new feature set for the objects to be clustered, using the labels assigned to them by each one of the input clusterings. Categorical algorithms can then be applied on this dataset to obtain the final clustering. In this direction, Topchy et al. (2003) show that using k-means on the binarized and standardized form of this matrix maximizes the proposed criterion of *partition utility*, whereas Topchy et al. (2004) model this new dataset using a mixture of multivariate multinomial distributions, one for each output cluster, which is then fitted using EM. Also, Nguyen and Caruana (2007), in addition to the aforementioned co-association-based method, propose two iterative refinement schemes which depend on Hamming distances using this feature representation.

Graph Reduction A third family of combination approaches is that of graph-based methods. The seminal work in this group is due to Strehl and Ghosh (2002). The authors propose two direct reductions of the ensemble clustering problem to graph or hypergraph partitioning, respectively, as well as a third and more complex transformation which also includes graph partitioning at one step. Fern and Brodley (2004) propose a fourth different reduction to graph partitioning, and Punera and Ghosh (2007) present generalizations to soft clustering ensembles of the methods in the previous works.

Voting The last major group of ensemble clustering approaches is that of voting strategies. Dimitriadou et al. (2002) and Sevillano et al. (2009) propose two merging-and-voting methods which present a number of similarities between them. In both, clusters are aligned across different clusterings, and the membership of objects to the sets of aligned clusters (which become the clusters of the consensus clustering) is found using voting. Boulis and Ostendorf (2004) take a different approach, and solve the alignment problem using either linear programming or singular value decomposition. A final clustering stage is applied to obtain the output clustering, using the aligned cluster memberships as features.

Other Approaches In addition to the aforementioned ones, researchers have considered a number of alternative frameworks to solve the problem of clustering combination. For instance, some authors have also experimented with simulated annealing (Filkov and Skiena, 2003), or genetic algorithms (Faceli et al., 2006).

Weighted Ensembles Finally, it is worth mentioning that weighted ensemble clustering algorithms, able to determine the confidence of each input clustering in an unsupervised fashion, and to produce an output clustering biased towards agreement with more confident clusterings, have appeared recently. For instance, Li et al. (2007) incorporate weighting to their own previous method (Li et al., 2004b), whereas Gullo et al. (2009) propose a weighting scheme that may be used to enhance a wide variety of existing combination algorithms.

3.2.2.2 Ensemble Generation Strategies

Regarding the generation of the ensemble of clusterings to be later combined, a number of strategies have been distinguished[2]:

Plain Some stochastic element of an individual clustering algorithm (such as the starting centroids of k-means) is repeatedly seeded with different values (e.g., Fred, 2001).

Random-k A supervised clustering algorithm is repeatedly run, requiring different numbers of clusters in the output. This may be used in combination with the previous strategy (e.g., Fred and Jain, 2002).

Random-k+ Similar to **random-k**, but using a number of cluster deliberately larger than the one expected to be present in the dataset (e.g., Ghosh et al., 2002).

Random projection Data are linearly projected to a lower dimensional space, using randomly generated projection matrices (Fern and Brodley, 2003; Topchy et al., 2003).

Random subspacing A particular case of **random projection**, in which the projection is performed by selecting a subset of the original dimensions (Greene et al., 2004).

Bagging Mimicking bagging in supervised learning, clusterings are repeatedly built on random subsamples of the dataset (Leisch, 1999).

Artificial data Additional data is artificially generated, and the obtained extended datasets are clustered (Luo et al., 2007).

Heterogeneousness Instead of relying on a single clustering algorithm, the data are clustered using a heterogeneous set of them (Strehl and Ghosh, 2002). A variation on this approach is to use a single algorithm, but several data representations (Li et al., 2004b; Sevillano et al., 2006).

A few authors have tried to compare the effectivity of these heuristics. For instance, Hadjitodorov and Kuncheva (2007) use genetic algorithms to determine which combination of heuristics produced the best results across a collection of 18 datasets—as well as the minimum required ensemble size for these results to stabilize.

3.3 Problem Definition

In accordance to its subjective nature (Dubes and Jain, 1976), we give here an *intentional* (one may even say *anthropic*) definition of clustering: a clustering differs from an arbitrary partition of a dataset in the fact that it has been generated with the *aim* to maximize a criterion function F (e.g., similarity of objects within a cluster, for a particular similarity function)—even if it does not actually maximize it.

3.3.1 Partitional Clustering

Assume we have a dataset $\mathcal{X} = \{x_1 \dots x_n\}$. We can then define:

Definition 3.1 (Hard partitional clustering)
*A **hard (partitional) clustering** Π of dataset $\mathcal{X}$ is a partition*[3] $\Pi = \{\pi_1 \dots \pi_k\}$ *of $\mathcal{X}$ whose aim is to maximize a certain criterion function F. Each one of the subsets $\pi_c \in \Pi$ is a **hard cluster**.*

[2] Adapted from Greene et al. (2004).

[3] Following the usual set-theoretical definition (see Definition A.1 in Appendix A.1).

Remark Because of the bijection between partitions and equivalence relations, a hard clustering infers an equivalence relation $x_i \overset{\Pi}{\sim} x_j$ between elements of $\mathcal{X}$:

$$x_i \overset{\Pi}{\sim} x_j \quad \longleftrightarrow \quad \exists \pi_c \in \Pi : \; x_i \in \pi_c \wedge x_j \in \pi_c$$

We will read $x_i \overset{\Pi}{\sim} x_j$ as x_i and x_j **are clustered together** in Π. When the clustering Π is clear from the context, the lighter $x_i \sim x_j$ notation will be used instead.

Definition 3.2 (Soft partitional clustering)
*A **soft (partitional) clustering** Π of dataset $\mathcal{X}$ is a fuzzy pseudopartition*[4] $\Pi = \{\pi_1 \ldots \pi_k\}$ *of $\mathcal{X}$ whose aim is to maximize a certain criterion function F. Each one of the fuzzy subsets $\pi_c \in \Pi$ is a **soft cluster**.*

Remark A hard clustering can be seen as a particular case of soft clustering where the grade of membership of a certain x_i to the π_c is zero for all but exactly one cluster, for which the grade is one.

Definition 3.3 (Clustering cardinality)
*The **cardinality** of a hard of soft clustering is the number of clusters it contains.*

Remark Being sets of clusters, this definition of cardinality is nothing but the usual set-theoretical one.

Definition 3.4 (Partitional clustering problem)
Given

- *a dataset $\mathcal{X}$*
- *a criterion function F*
- *a hypothesis space for hard/soft clusterings Ω_Π*

*the **partitional clustering problem** is that of finding the clustering $\Pi \in \Omega_\Pi$ which maximizes the value of $F(\Pi)$.*

Remark If the hypothesis space is restricted to contain clusterings with a fixed number k of clusters, we will talk of the **supervised partitional clustering problem**. Otherwise, the name **unsupervised partitional clustering problem** will be used.

3.3.2 Hierarchical Clustering

It is also possible to define:

Definition 3.5 (Subsumption tree)
*A **subsumption tree** Ψ over a dataset $\mathcal{X}$, of cardinality $n = |\mathcal{X}|$, is a sorted sequence of sets $\Psi = (\psi_1 \ldots \psi_d)$ such that:*

- *The length d of the sequence is $d = 2n - 1$*
- *The first n sets are singletons containing all the objects in $\mathcal{X}$:*

$$\forall j \in \{1 \ldots n\} : \; \exists x_i \in \mathcal{X} : \; \psi_j = \{x_i\} \qquad \forall x_i \in \mathcal{X} : \; \exists j \in \{1 \ldots n\} : \; \psi_j = \{x_i\}$$

- *The next $n - 1$ sets are the union of two preceding sets:*

$$\forall j \in \{n+1 \ldots d\} : \; \exists a, b < j : \; a \neq b \wedge \psi_j = \psi_a \cup \psi_b$$

*We shall say in this case that ψ_j **directly subsumes** or is the **parent** of ψ_a and ψ_b, which are its **children**. The sets ψ_a and ψ_b can thus be referred to as **siblings**.*

[4] Following the definitions of Bezdek (1981) and Klir and Yuan (1995) (see Definition A.3 in Appendix A.2).

- *All sets but the last are directly subsumed by exactly one succeeding set:*

$$\forall j \in \{1 \dots d-1\} : \exists m > j : \psi_j \subset \psi_m$$
$$\forall j \in \{1 \dots d-1\} : \nexists\, m, m' > j : \psi_j \subset \psi_m \wedge \psi_j \subset \psi_{m'} \wedge m \neq m'$$

- *The last set ψ_d is equal to the complete dataset*[5]*: $\psi_d = \mathcal{X}$.*

Remark The sets within the subsumption tree, linked by the direct subsumption relation, form a rooted tree structure when regarded a directed graph. We can thus name each set a **node** of the tree; the first n sets, the **leaves**; and the next $n-1$, the **branches**—with the particular case of the last ψ_d being the **root**.

Definition 3.6 (Node level)
*Given a subsumption tree Ψ of dataset $\mathcal{X}$, of cardinality n, the **level** of a node ψ_j is*

$$\text{level}(\psi_j) = \begin{cases} n & \text{if } j \in \{1 \dots n\} & \text{(the node is a leaf)} \\ 2n - j & \text{if } j \in \{n+1 \dots 2n-1\} & \text{(the node is a branch)} \end{cases}$$

In particular, the root of the tree is at level 1.

Definition 3.7 (Tree cut)
*Given a tree Ψ over dataset $\mathcal{X}$, of cardinality n, a **cut** of the tree at level $l \in \{1 \dots n\}$ is the set of nodes at level larger or equal than l, and which are not subsumed by another node at level larger or equal than l.*

$$\text{cut}(\Psi, l) = \{\psi_j \mid \psi_j \in \Psi \wedge \text{level}(\psi_j) \geq l \wedge (\nexists\, \psi_m \in \Psi : \text{level}(\psi_m) \geq l \wedge \psi_j \subset \psi_m \wedge \psi_j \neq \psi_m)\}$$

It can be proved that:

Proposition 3.8
A cut of tree Ψ over dataset $\mathcal{X}$ at any valid level l is a partition of $\mathcal{X}$.

Proof See Appendix B. ∎

We may then proceed to additionally define:

Definition 3.9 (Dendrogram/hierarchical clustering)
*A **dendrogram** or **hierarchical clustering** Ψ of a dataset $\mathcal{X}$ is a subsumption tree $\Psi = (\psi_1 \dots \psi_d)$ whose aim is to maximize a certain criterion function F when cut at every level. Each one of the nodes $\psi_c \in \Psi$ is an **internal cluster**.*

Definition 3.10 (Hierarchical clustering problem)
Given

- *a dataset $\mathcal{X}$*
- *a criterion function F*
- *a hypothesis space for dendrograms Ω_Ψ*

*the **hierarchical clustering problem** is that of finding the dendrogram $\Psi \in \Omega_\Psi$ which maximizes the value of $F(\Psi, l))$ at every level l.*

[5] This is a consequence of the previous properties (see proof in Appendix B).

3.3.3 Ensemble Clustering

Finally, we can define:

Definition 3.11 (Clustering ensemble)
A ***hard/soft clustering ensemble*** *E for dataset $\mathcal{X}$ is a set of hard/soft clusterings $E = \{\Pi_1 \ldots \Pi_r\}$ of $\mathcal{X}$ (with corresponding criterion functions $\{F_1 \ldots F_r\}$, equal or different).*

Definition 3.12 (Consensus clustering problem)
Given

- *a dataset $\mathcal{X}$*
- *a consensus function G*
- *a clustering ensemble E*
- *a hypothesis space for hard/soft clusterings Ω_E*

the ***consensus clustering problem*** *is that of finding the clustering $\Pi_E \in \Omega_E$ which maximizes the value of $G(\Pi_E; E, \mathcal{X})$.*

Remark Similarly to Definition 3.4, the names **supervised consensus clustering problem** or **unsupervised consensus clustering problem** can be used, according to whether the hypothesis space is restricted or not to contain clusterings with a fixed number k of clusters.

Definition 3.13 (Ensemble clustering problem)
If the consensus function G does not access the original features of dataset $\mathcal{X}$, the consensus clustering problem receives the name of ***ensemble clustering problem****.*

The names ***supervised ensemble clustering problem*** *or* ***unsupervised ensemble clustering problem*** *are also correspondingly defined.*

3.3.4 Clustering Model

Some algorithms are only devised to build a clustering of a input dataset, and do not provide any device to determine the hypothetical assignments of new objects to one of the obtained clusters. This is the case, for instance, of most hierarchical (including HAC) and ensemble clustering (such as Ghosh et al., 2002; Gionis et al., 2005) algorithms. However, most popular partitional methods—starting with k- and c-means, and continuing with all probabilistic mixture algorithms—provide, as a byproduct of the clustering process, a *clustering model* which may then be later used as a classification model for new data, after identifying the obtained clusters with classes.

If the dataset $\mathcal{X}$ used to construct a partitional clustering Π is drawn from a domain X—i.e., $\mathcal{X} \subset X$—we can define:

Definition 3.14 (Hard partitional clustering model)
A ***hard (partitional) clustering model*** *$\mathcal{M}$ for a hard clustering Π over the domain X is a function from X to Π.*

Remark The image $\mathcal{M}(x_x) = \pi_x$ of an object x_x is the cluster in Π to which the object would belong, should it had been included in the dataset used to generate the clustering.

Definition 3.15 (Soft partitional clustering model)
A ***soft (partitional) clustering model*** *$\mathcal{M}$ for a soft clustering Π, with k clusters, over the domain X is a function from X to the k-simplex Δ_k.*

Remark The elements in the image $\mathcal{M}(x_x) = \delta_x = (\delta_{x1} \ldots \delta_{xk})$ of an object x_x are the grades of membership to each one of the clusters in Π that the object would have had, should it had been included in the dataset used to generate the clustering[6].

[6] The conditions stated in these remarks should be regarded with caution. The inclusion of an extra object in the dataset might radically change the clusterings produced by the algorithm. One possible way to specify these functions more precisely is to require their output to be the cluster of the object in the closest clustering (according to some clustering distance function inherent to the algorithm and model) such that the assignment of all objects in the starting dataset $\mathcal{X}$ remains the same as in Π.

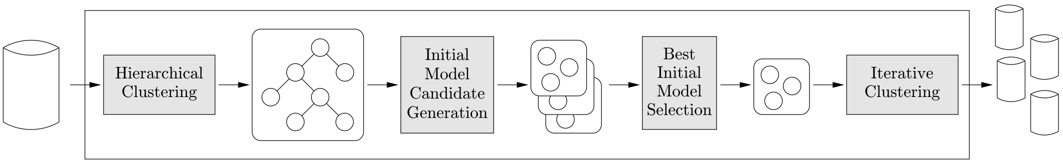

Hierarchical Clustering
Initial Model Candidate Generation
Best Initial Model Selection
Iterative Clustering

Figure 3.1: Hybrid unsupervised clustering method (Surdeanu et al., 2005)

3.4 Unsupervised Clustering Approaches

This section describes the components of the unsupervised clustering approaches we have considered for our comparison. More specifically, Section 3.4.1 describes the set of unsupervised clustering algorithms we have used; and Section 3.4.2 is concerned with ensemble clustering, including both the combination method and the generation strategies.

3.4.1 Individual Approaches

As mentioned in the chapter introduction, the particular biases of individual clustering methods, as well as the kind of object representation and similarity measure used by these methods, imply a different point of view over the datasets to be clustered. We have considered a heterogeneous set of individual unsupervised clustering methods:

- The hybrid method of Surdeanu et al. (2005), which has been shown to obtain good results in unsupervised document clustering of different real world data.

- An adaptation of the previous hybrid method using information theoretical components.

- A *generate-and-select* method, which combines a hierarchical algorithm with a model selection criterion.

A detailed description of each one of them follows.

3.4.1.1 Geometric Hybrid Method

The motivation behind the system of Surdeanu et al. (2005) is that, among partitional clustering algorithms, "*iterative refinement clustering techniques are extremely popular due to their good performance, relative simplicity, and good theoretical foundations*". However, they present two obvious drawbacks: first, being supervised, they require the number of clusters to be given a priori; and, second, they are sensitive to the choice of the initial model parameters.

The proposal of the authors is graphically depicted in Figure 3.1, and starts by using a hierarchical clustering algorithm to obtain a dendrogram of the dataset. According to a number of heuristics, several subsets of nodes in the dendrogram are selected as *initial model candidates*. A criterion function is used to choose a single one among them, and it is this best candidate which is finally used as initial model for the iterative algorithm.

Although this scheme admits several algorithms and measures, Surdeanu et al. give a concrete implementation, using a *geometric* approach. The implementation is described in detail below, as it is the basis for our own information theoretical approach (Section 3.4.1.2).

We will henceforth refer to this method as GEO.

Document Representation Documents are represented using the popular *bag-of-words* formalism. Given a dictionary $\Omega = (\omega_1 \dots \omega_z)$, a document x_i is represented as a vector $\vec{x}_i = (x_{i1} \dots x_{iz})$, where $x_{iw} \in \mathbb{N}$ is the number of occurrences of word ω_w in document x_i (also known as term frequency).

Similarity between documents is quantified by the cosine of their vector representations. To avoid the predominance of frequent words in the calculation, the usual tf-idf weighting is applied:

each term frequency (tf) is multiplied by the inverse document frequency (idf) of the term (Spärck-Jones, 1972). The measure is then effectively calculated as:

$$\begin{aligned}\text{sim}(x_i, x_j) &= \frac{\sum_{w=1}^{z} (x_{iw} \cdot \text{idf}_w) \times (x_{jw} \cdot \text{idf}_w)}{\sqrt{\sum_{w=1}^{z} (x_{iw} \cdot \text{idf}_w)^2 \times \sum_{w=1}^{z} (x_{jw} \cdot \text{idf}_w)^2}} \\ \text{idf}_w &= -\log \frac{|\{x_i \mid x_i \in \mathcal{X} \wedge x_{iw} > 0\}|}{|\mathcal{X}|}\end{aligned}$$

In situations where a dissimilarity (distance) instead of a similarity is required, cosine distance is used:

$$\text{dist}(x_i, x_j) = 1 - \text{sim}(x_i, x_j)$$

Hierarchical Clustering In order to obtain a dendrogram Ψ of the dataset, the authors resort to HAC. The algorithm starts assigning each object to a singleton cluster. It then iteratively merges, at every step, the two most similar clusters, and replaces them with their union, up to the point where a single cluster with the whole dataset remains.

Similarity between clusters is defined using the UPGMA rule:

$$\text{sim}_{\text{U}}(\pi_c, \pi_d) = \frac{1}{|\pi_c| \cdot |\pi_d|} \sum_{x_i \in \pi_c} \left(\sum_{x_j \in \pi_d} \text{sim}(x_i, x_j) \right)$$

Candidate Generation After the dendrogram Ψ is found, the initial model candidates can be generated. The intuition here is to look for internal clusters in the dendrogram which are *tight* (i.e., the distances between objects within the cluster are small) and *separated* (i.e., the distances to objects outside the cluster are large). Surdeanu et al. determine four functions which quantify to which extent nodes in the dendrogram possess these properties:

Within distances (W) Nodes ψ_c corresponding to tight clusters should have small pairwise distances *within* the objects in them:

$$\text{W}(\psi_c) = \frac{1}{|\psi_c| \cdot (|\psi_c| - 1)} \sum_{x_i, x_j \in \psi_c} \text{dist}(x_i, x_j)$$

Between distances (B) Nodes ψ_c corresponding to separated clusters should have large distances *between* their objects and those in the rest of the dataset:

$$\text{B}(\psi_c) = \frac{1}{|\psi_c| \cdot (|\mathcal{X}| - |\psi_c|)} \sum_{x_i \in \psi_c} \sum_{x_j \in \mathcal{X} \setminus \psi_c} \text{dist}(x_i, x_j)$$

Neighbourhood distances (N) Nodes ψ_C corresponding to separated clusters should, in particular, have large distances with respect to their neighbours. This function tries to minimize the effect large groups of distant objects on the B function, and is defined as the UPGMA distance between the node and its sibling in the dendrogram:

$$\text{N}(\psi_c) = \text{dist}_{\text{U}}(\psi_c, \text{sibling}(\psi_c))$$

Growth (G) Nodes ψ_c corresponding to tight clusters should have been produced by merging two nodes that were already close. This function tries to avoid the bias exhibited by the other three functions towards small clusters. The proposed way to quantify it is the node *growth*, defined as the expansion at the union which generated node ψ_c (i.e., the UPGMA distance), relative to the internal density of its two children ψ_a and ψ_b:

$$\begin{aligned}\text{G}(\psi_c) &= \frac{\text{dist}_{\text{U}}(\psi_a, \psi_b)}{\text{density}(\psi_a, \psi_b)} \\ \text{density}(\psi_a, \psi_b) &= \frac{\sum_{x_i, x_j \in \psi_a} \text{dist}(x_i, x_j) + \sum_{x_i, x_j \in \psi_b} \text{dist}(x_i, x_j)}{|\psi_a| \cdot (|\psi_a| - 1) + |\psi_b| \cdot (|\psi_b| - 1)}\end{aligned}$$

W	$1/(\mathrm{W}(\psi_c))$	**GW**	$1/(\mathrm{G}(\psi_c)\cdot\mathrm{W}(\psi_c))$
WB	$\mathrm{B}(\psi_c)/(\mathrm{W}(\psi_c))$	**GWB**	$\mathrm{B}(\psi_c)/(\mathrm{G}(\psi_c)\cdot\mathrm{W}(\psi_c))$
WN	$\mathrm{N}(\psi_c)/(\mathrm{W}(\psi_c))$	**GWN**	$\mathrm{N}(\psi_c)/(\mathrm{G}(\psi_c)\cdot\mathrm{W}(\psi_c))$

Table 3.1: Cluster quality measures (Surdeanu et al., 2005).

Six *cluster quality measures*, which are combinations of these four functions, are proposed to score the nodes. Their exact formulae are described in Table 3.1.

For each cluster quality measure, the procedure starts by ranking the internal clusters in the dendrogram with respect to it. An additional coverage parameter $\gamma \in (0 \dots 1)$ is considered at this point, with the aim of taking only the most confident objects: the topmost ranked nodes which do not include more objects than a fraction γ of the dataset size are selected. Nodes subsumed by other nodes in the set are removed, and the remaining ones become clusters of an initial model candidate.

The number of clusters is thus determined as a byproduct, being equal to the number of nodes in this set. It is also important to note than, because of the candidate generation process, some objects will not belong to any cluster in the candidate—i.e., the candidates are *partial clusterings*.

The process is performed for all six cluster quality measures and for different coverage values γ. As a result, a pool of initial candidates candidates is obtained.

Candidate Selection To select the best clustering candidate within the pool, the authors propose the use of a *global quality measure*. Following the results of the comparison of criterion functions reported by Milligan and Cooper (1985), Caliński and Harabasz' index is used. The index is a normalized ratio of *between distances* (i.e., distances between objects belonging to different clusters) and *within distances* (i.e., distances between objects belonging to the same cluster).

If we define the *centroid* $\bar{x}_c$ of a cluster π_c as the average of all objects $x_i \in \pi_c$, and the *metacentroid* $\bar{x}$ of a dataset $\mathcal{X}$ as that of all objects $x_i \in \mathcal{X}$:

$$\bar{x}_c = \frac{\sum_{x_i \in \pi_c} x_i}{|\pi_c|} \qquad \bar{x} = \frac{\sum_{x_i \in \mathcal{X}} x_i}{|\mathcal{X}|}$$

The Caliński and Harabasz index C can then defined as:

$$\begin{aligned}
\mathrm{C}(\Pi;\mathcal{X}) &= \frac{\mathrm{C_B}(\Pi;\mathcal{X})}{\mathrm{C_W}(\Pi;\mathcal{X})} \times \frac{|\mathcal{X}| - |\Pi|}{|\Pi| - 1} \\
\mathrm{C_B}(\Pi;\mathcal{X}) &= \sum_{\pi_c \in \Pi} \mathrm{dist}(\bar{x}_c, \bar{x}) \\
\mathrm{C_W}(\Pi;\mathcal{X}) &= \sum_{\pi_c \in \Pi} \sum_{x_i \in \pi_c} \mathrm{dist}(x_i, \bar{x}_c)
\end{aligned}$$

where $\mathrm{C_B}$ contains the between distances and $\mathrm{C_W}$ the within ones.

The selection is then performed in a two-step procedure: first, for each cluster quality measure, the last local maximum of the C index for increasing values of coverage γ is found using grid search; and, second, the maximum of all measure-wise local maxima is finally selected as initial model.

Partitional Clustering In the last step, that of iterative partitional clustering, the selected candidate is used as initial model to fit a generative probabilistic model. In particular, a mixture of multinomial distributions is used. This model was proposed for document clustering by Meilă and Heckerman (2001), but it is identical to the one previously used for document classification by Nigam et al. (2000).

The considered model uses a mixture of k components, each one of them corresponding to a cluster. Withing each component, the occurrence of a word in is modelled using a multinomial distribution, and each occurrence is considered independent of the others given the component (naive Bayes assumption). For each object x_i, a hidden (unknown) variable $y_i \in \{1 \dots k\}$ will contain the component which generated it.

The resulting model can be formulated as:

$$\begin{aligned}
p(x_i\,;\Theta) &= \sum_{c=1}^{k} p(y_i = c\,;\Theta)\cdot p(x_i \mid y_i = c\,;\Theta)\\
p(y_i = c\,;\Theta) &= \alpha_c\\
p(x_i \mid y_i = c\,;\Theta) &= \prod_{w=1}^{z} p(x_{iw} \mid y_i = c\,;\Theta)\\
&= \prod_{w=1}^{z} {\vartheta_{cw}}^{x_{iw}}
\end{aligned}$$

The $\{\alpha_c\}$ and $\{\vartheta_{cw}\}$ are the parameters of the model, which, additionally, should accomplish the restrictions:

$$\sum_{c=1}^{k} \alpha_c = 1 \qquad \forall c \in \{1\dots k\} : \sum_{w=1}^{z} \vartheta_{cw} = 1$$

i.e., the $\{\alpha_c\}$ and each set of $\{\vartheta_{cw}\}$ must belong to a simplex.

Both maximum likelihood (ML) and maximum a posteriori (MAP) estimations of parameters, $\widehat{\Theta}$, can be found using the EM algorithm. Given that, as mentioned, the model parameters fall inside an simplex, a Dirichlet distribution can be used as conjugate prior. In particular, a symmetric Dirichlet distribution with a parameter value of 1 has been used—under these conditions, MAP estimation is equivalent to Laplace smoothing (Manning and Schütze, 1999).

To obtain the clustering from the estimated model $\widehat{\Theta}$, each component is mapped to a cluster, and each object is assigned to the cluster whose component has the largest probability of having generated it:

$$\begin{aligned}
\Pi &= \{\pi_1 \dots \pi_k\}\\
\pi_c &= \{x_i \mid x_i \in \mathcal{X} \wedge \arg\max_{d} p(y_i = d \mid x_i\,;\widehat{\Theta}) = c\}
\end{aligned}$$

3.4.1.2 Information-Theoretical Hybrid Method

The field of information theory (IT) goes back to the seminal work of Shannon (1948). Given the attention devoted in the last decade to its application to the task of document clustering (Slonim and Tishby, 1999; Gokcay and Principe, 2002; Dhillon et al., 2003), we propose a different implementation of the aforementioned hybrid method using information-theoretical components, which we describe below[7].

We will henceforth refer to this method as INFO.

Document Representation As IT is mainly concerned with probability distributions, the most natural, and usual, document representation in this framework is that of documents as discrete conditional probability distributions. Given, as in Section 3.4.1.1, a dictionary of words $\Omega = (\omega_1 \dots \omega_z)$, we can define two random variables, Y on Ω and X on $\mathcal{X}$, whose joint distribution can be decomposed as:

$$p(\omega_w, x_i) = p(\omega_w \mid x_i)\cdot p(x_i)$$

Each document x_i is thus represented as the conditional distribution $p(\omega \mid x_i)$. It is often assumed that all documents have the same a priori probability:

$$\forall x_i \in \mathcal{X} : \; p(x_i) = \frac{1}{|\mathcal{X}|}$$

Using maximum likelihood estimation, it is possible to obtain the conditional distribution from the bag-of-words representation as:

$$p(\omega_w \mid x_i) = \frac{x_{iw}}{\sum_{w=1}^{z} x_{iw}}$$

However, other forms of estimation, such as maximum a posteriori, are also sometimes used within IT-based algorithms (Dhillon and Guan, 2003).

[7] For the sake of fluency, the definitions of common IT concepts (e.g., entropy, divergence...) are omitted from the presentation. They are however fully detailed in Appendix A.3.

Hierarchical Clustering The generation of the dendrogram Ψ from dataset $\mathcal{X}$ is accomplished using the agglomerative information bottleneck algorithm (aIB; Slonim and Tishby, 1999).

The authors observe that a clustering Π over a dataset $\mathcal{X}$ defines a new random variable $\tilde{X}$ on Π, where:

$$\tilde{X} = \pi_c \quad \longleftrightarrow \quad X = x_i \wedge x_i \in \pi_c$$

The aim of aIB is to find the clustering $\hat{\Pi}$ for which $\tilde{X}$ preserves the maximum mutual information between the random variables Y and X:

$$\begin{aligned} \hat{\Pi} &= \arg\min_{\Pi} I(Y\,;X) - I(Y\,;\tilde{X}) \\ &= \arg\max_{\Pi} I(Y\,;\tilde{X}) \end{aligned}$$

Similarly to HAC, aIB is agglomerative in nature (i.e., constructs the dendrogram in a bottom-up fashion), but instead of using cosine similarity, it merges, at each step, the two clusters whose weighted Jensen-Shannon divergence (Lin, 1991) is minimal:

$$JS^W(\tilde{x}_i, \tilde{x}_j) = (p(\tilde{x}_i) + p(\tilde{x}_j)) \times \mathrm{JS}(p(\omega \mid \tilde{x}_i) \parallel p(\omega \mid \tilde{x}_j))$$

The process is iterated until only one cluster remains, containing the whole dataset.

Candidate Generation The candidate generation process mimics the one devised by Surdeanu et al.. The only difference is our adaptation of the six proposed cluster quality metrics, replacing cosine distance by an IT-based dissimilarity measure.

In particular, our proposal is to use Jensen-Shannon divergence between the word conditional distributions of the documents:

$$\mathrm{dist}(x_i, x_j) = \mathrm{JS}(p(\omega \mid \tilde{x}_i) \parallel p(\omega \mid \tilde{x}_j))$$

Other measures exist, coming from information theory and which could be useful as object dissimilarity, such as Kullback-Leibler divergence (Kullback and Leibler, 1951) or even mutual information (Shannon, 1948). However, on the contrary of Jensen-Shannon divergence, they are not symmetric, and require absolute continuity of one distribution with respect to the other—two properties that make them unsuitable for our purposes.

Candidate Selection We tried to adapt Caliński and Harabasz' index to also use Jensen-Shannon divergence—but we found, in preliminary experiments, its performance to be surpassed by that of a more IT-motivated criterion that we present here.

For a given dataset $\mathcal{X}$ and clustering Π, the proposed criterion function is calculated as the sum of *between cross-entropies* (i.e., cross-entropies between the centroid of each cluster and the metacentroid of the dataset) and *within cross-entropies* (i.e., cross-entropies between the objects in each cluster and the corresponding centroid), normalized by the number of objects in the dataset:

$$\begin{aligned} \mathrm{H}(\Pi\,;\mathcal{X}) &= \frac{\mathrm{H_B}(\Pi\,;\mathcal{X}) + \mathrm{H_w}(\Pi\,;\mathcal{X})}{|\mathcal{X}|} \\ \mathrm{H_B}(\Pi\,;\mathcal{X}) &= \sum_{\pi_c \in \Pi} \mathrm{H}^{\times}(p(\omega \mid \bar{x}_c), p(\omega \mid \bar{x})) \\ \mathrm{H_W}(\Pi\,;\mathcal{X}) &= \sum_{\pi_c \in \Pi} \sum_{x_i \in \pi_c} \mathrm{H}^{\times}(p(\omega \mid x_i), p(\omega \mid \bar{x}_c)) \end{aligned}$$

where, $\mathrm{H}^{\times}(p, q)$ represents the cross-entropy between the two probability distributions p and q[8]. In order to select the best candidate, the clustering Π for which the normalized sum of cross-entropies $\mathrm{H}(\Pi\,;\mathcal{X})$ is minimum is chosen.

Minimization of cross-entropy has been previously used as selection criterion in IT settings. In particular, proper *minimum cross-entropy* (originally proposed as *minimum discrimination information* by Kullback, 1959) is a theoretically sound framework which has been successfully applied

[8] We have diverted from the standard notation $\mathrm{H}(p, q)$ for cross-entropy, in order to avoid the clash with the use of the same representation to denote the related, but different, measure of joint entropy.

for classification and clustering tasks (Shore and Gray, 1982)—and which includes the popular maximum entropy framework (Jaynes, 1957) as a particular case.

Additionally, by the Kraft-McMillan theorem (Kraft, 1949; McMillan, 1956), it can be proved that the value of the cross-entropy between distributions p and q is equivalent to the average length of coding a signal whose distribution is p using the optimal code derived from distribution q. Hence, the proposed criterion is related to the length, normalized by the number of objects, of sending the dataset using a two-step procedure:

1. Send the distribution of words in the centroid of each cluster, using a code derived from the overall distribution of the collection (i.e., the metacentroid).

2. Send the distribution of words in each document, using a code derived from the distribution in of the centroid of its cluster.

This fact allows us to establish a connection with other classical information theoretical model selection criteria, such as minimum message length and minimum description length (see Section 3.2.1)—which we could not use at this stage because of their requiring a probabilistic model.

The criterion is also appealing because it includes an implicit measure of the *goodness* of the number of clusters: a larger number of clusters implies a larger $\mathrm{H_B}(\Pi\,;\mathcal{X})$ but a smaller $\mathrm{H_W}(\Pi\,;\mathcal{X})$ and vice-versa, so both subestimations and overestimations of this number are penalized.

Partitional Clustering Finally, the iterative refinement algorithm applied to the selected candidate is divisive information theoretical clustering (DITC; Dhillon and Guan, 2003).

Similarly to aIB, the algorithm tries to find the clustering $\hat{\Pi}$ which minimizes the loss of mutual information. The barebones version of DITC, as proposed by Dhillon et al., iteratively reestimates the distribution of words in each cluster from the current assignment, and then assigns each document to the cluster with which it has the smallest Kullback-Leibler divergence on word distributions, until convergence:

$$\begin{aligned} p^{t+1}(\omega \mid \bar{x}_c) &= \frac{\sum_{x_i \in \pi_c^t} p(x_i) \cdot p(\omega \mid x_i)}{\sum_{x_i \in \pi_c^t} p(x_i)} \\ \pi_c^{t+1} &= \{x_i \mid x_i \in \mathcal{X} \wedge \arg\min_d \mathrm{KL}(p(\omega \mid x_i) \mid p^{t+1}(\omega \mid \bar{x}_d)) = c\} \end{aligned}$$

However, Dhillon and Guan state that the performance of the algorithm degrades when faced to high-dimensional and sparse data, such as document collections. For instance, sparsity can produce infinite values of Kullback-Leibler divergence, because of the absolute continuity requirement imposed by this measure. Additionally, the increase in dimensionality makes it easier for the algorithm to get trapped in local maxima.

In order to reduce the impact of these phenomena, the authors propose two enhancements to the basic scheme:

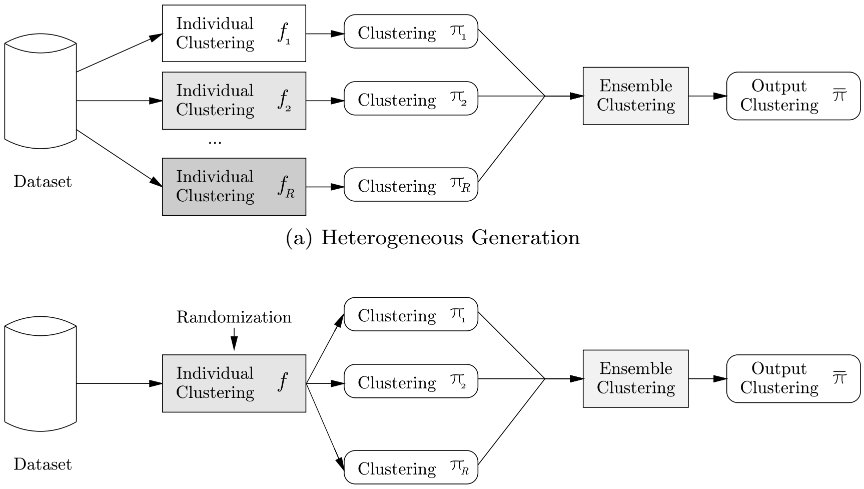
Priors Instead of ML, use MAP for the estimation of the conditional words distributions of the documents, with a symmetric Dirichlet distribution as prior—so as to avoid the problems caused by sparsity.

Local search After convergence of the iterative refinement procedure, perform a local search step—so as to step out of the local maxima caused by high dimensionality. The hill climbing procedure uses as search operator chains of successive moves of a single object to a different cluster, of up to a certain fixed length.

The resulting procedure alternatively performs iterative refinement and local search, until convergence. At every iteration, the value of the parameter of the prior distribution is decreased, reducing its influence (in a fashion similar to the temperature parameter in simulated annealing; Kirkpatrick et al., 1983).

3.4.1.3 Hierarchical Method

The third and last considered individual unsupervised clustering method is a simple two-stage generate-and-select approach, consisting of a hierarchical algorithm and a model selection criterion. More specifically:



(a) Heterogeneous Generation

(b) Randomized Generation

Figure 3.2: Unsupervised ensemble clustering

1. A dendrogram is built using aIB over the conditional distribution representations of the documents.

2. The Caliński and Harabasz' index is found for every level of the dendrogram, using the cosine distance between the bag-of-words representations of the document.

   The lowest level at which a local maximum of the index occurs is selected, and the dendrogram is cut at that level.

Despite being simple, we believe the method is interesting because it combines information from several sources by decoupling the clustering and selection representations.

We will henceforth refer to this method as Hi.

3.4.2 Ensemble Approaches

As mentioned in Section 3.2.2, the use of ensemble clustering methods requires both a strategy to generate an ensemble, and a method to combine the clusterings in this ensemble into a consensus one.

We have considered two different strategies for ensemble generation, which are graphically depicted in Figure 3.2:

(a) a *heterogeneous* generation strategy, which performs a single run of multiple unsupervised clustering algorithms;

(b) and a *randomized* generation strategy, which performs multiple runs of a single supervised clustering algorithm.

These two strategies come from opposite decisions in the informedness versus cost trade-off: the former generates a reduced number of clusterings coming from informed, and hence potentially expensive, unsupervised algorithms; whereas the latter, as the set of random initializations of an algorithm is virtually unlimited, allows the generation of a much larger number of clusterings.

Sections 3.4.2.1 and 3.4.2.2 describe these two strategies, respectively.

Regarding the combination method, we have implemented the three algorithms proposed by Gionis et al. (2005), which are based on the reduction of ensemble clustering to the correlation clustering problem. All three algorithms are unsupervised, and hence able to detect the number of clusters in the final clustering.

Section 3.4.2.3 presents an overview of the used combination methods.

Algorithm 3.1 Minor ensemble strategy

```
Input: A dataset $\mathcal{X}$
Input: A set of unsupervised clustering functions (algorithms) $F = \{f_1 \ldots f_R\}$
Output: A clustering ensemble $E$
Initialize the ensemble $E$ to the empty set
    $E \leftarrow \varnothing$
For $r = 1 \ldots R$ do
    Apply $f_r$ to $\mathcal{X}$, to obtain clustering $\Pi_r$
        $\Pi_r = f_r(\mathcal{X})$
    Append the obtained clustering $\Pi_r$ to the ensemble $E$.
        $E \leftarrow E \cup \{\Pi_r\}$
Return the ensemble $E$
```

3.4.2.1 Heterogeneous Generation

The heterogeneous generation strategy simply applies, in turns, a number of individual unsupervised clustering algorithms to the dataset, and returns the ensemble of clusterings produced by each one of them. This scheme was identified by Greene et al. (2004) as *heterogeneous ensembles* (see Section 3.2.2.2), and its procedural description is presented in Algorithm 3.1.

For our experiments, we have used the three algorithms presented in Section 3.4.1 (Geo, Info and Hi) to generate the ensembles by this strategy. Because of the more reduced number of clusterings that are generated using this strategy, we will henceforth refer to it as Minor.

3.4.2.2 Randomized Generation

In opposition to Minor, the randomized generation strategy seeks to obtain diversity within an ensemble by repeatedly applying a supervised and stochastic (i.e., non-deterministic) algorithm—such as k-means or EM.

In particular, the randomized strategy can be used with a computationally undemanding algorithm to obtain large ensembles in an inexpensive way, albeit consisting of much less *individually* informed clusterings. Such ensembles of *weak* (i.e., slightly better than random) clusterings were first proposed by Topchy et al. (2003). However, combinations of weak learning algorithms have long since been used in supervised learning (Freund and Schapire, 1995).

Even if multiple randomized schemes are possible, our proposal focuses on refinement clustering algorithms, and hence consists of generating a number of starting clusterings at random, and then using each one of them as input for the supervised algorithm. To generate the starting clusters, first the number of clusters and then a number of *seed* objects to be used as centroids are selected.

The selection of the number of clusters is done in a completely uninformed way, selecting a number at random up to a user-given k_{max}. The strategy hence falls within the *random-k* category of Greene et al. (see Section 3.2.2.2).

A more detailed procedural description of this process is given in Algorithm 3.2. As described therein, to generate each clustering in the ensemble the first step is the selection of the effective number of clusters in the clustering, k_r (line 3). Any discrete distribution between 2 and k_{max}, such as the uniform distribution, can be used. Then a subset $\hat{\mathcal{X}}_r$ of size k_r is selected at random from $\mathcal{X}$ (line 4). We shall name this subset the *seed subset*, and each one of their members will be a *seed*.

The seed subset is extended to a clustering Π_r^0, which contains a singleton cluster for each one of the seeds (line 5); and this clustering is then used as input to a refinement clustering algorithm (line 6). The resulting clusterings Π_r are finally collected to generate the output ensemble E (line 7).

Algorithm 3.2 MAJOR ensemble strategy

```
Input: A dataset $\mathcal{X}$
Input: An ensemble size $R$
Input: A maximum number of clusters $k_{max}$
Input: A refinement clustering function (algorithm) $f$
Output: A clustering ensemble $E$
Initialize the ensemble $E$ to the empty set
        $E \leftarrow \varnothing$
For $r = 1 \dots R$ do
    Draw a number of clusters $k_r$ at random from the range $\{2 \dots k_{max}\}$
        $k_r \in \{2 \dots k_{max}\}$
    Select a subset $\hat{\mathcal{X}}_r$ of $k_r$ seeds from $\mathcal{X}$
        $\hat{\mathcal{X}}_r = \{\hat{x}_{r1} \dots \hat{x}_{rk_r}\} \subset \mathcal{X}$
    Generate a starting clustering $\Pi_r^0$, with a singleton cluster $\pi_{rc}^0$ for each seed $\hat{x}_{rc} \in \hat{\mathcal{X}}_r$
        $\Pi_r^0 = \{\pi_{r1}^0 \dots \pi_{rk_r}^0\} \qquad \pi_{rc}^0 = \{\hat{x}_{rc}\}$
    Apply $f$ to $\mathcal{X}$, using $\Pi_r^0$ as initial clustering, to obtain clustering $\Pi_r$
        $\Pi_r = f(\mathcal{X}; \Pi_r^0)$
    Append the obtained clustering $\Pi_r$ to the ensemble $E$.
        $E \leftarrow E \cup \{\Pi_r\}$
Return the ensemble $E$
```

The strategy has a number of parameters, namely, the ensemble size R, the maximum number of clusters k_{max}, and the used refinement clustering algorithm. Regarding the last one, in our experiments we used again EM with the mixture model of Meilă and Heckerman (2001), as described in Section 3.4.1.1. Regarding R and k_{max}, we have experimented with several values in the ranges $R \in \{10 \dots 50\}$ and $k_{max} \in \{5 \dots 50\}$. The experimental section will shed light on the question of its influence on the quality of the output clustering (Section 3.5.3.1)

Because of the potentially large number of clusterings that can be generated using this strategy, we will henceforth refer to it as MAJOR.

3.4.2.3 Unsupervised Combination

As mentioned before, Gionis et al. (2005) propose three unsupervised methods for ensemble clustering, based on reduction to the problem of correlation clustering. In order to perform the reduction, all three methods start by finding the co-association matrix $C(E) = [c_{ij}]$ of the elements in $\mathcal{X}$, whose entries c_{ij} are the fraction of the total of clusterings in the ensemble $E = \{\Pi_1 \dots \Pi_R\}$ in which the objects x_i and x_j are clustered together:

$$c_{ij} = \frac{|\{\Pi_r \in E \mid x_i \sim x_j\}|}{|E|}$$

A dissimilarity matrix $D(E) = [d_{ij}]$ can be found from $C(E)$ as:

$$d_{ij} = \frac{|\{\Pi_r \in E \mid x_i \nsim x_j\}|}{|E|} = 1 - c_{ij}$$

It can be proved that its entries satisfy the triangle equality, and it is hence correct to talk of the *distance* matrix $D(E)$.

After the co-association matrix is built, the problem of ensemble clustering can be reduced to *correlation clustering*: a variant of the partitional clustering problem in which, instead of a dataset and a similarity function, the input is a matrix (or, equivalently, an undirected weighted graph) containing the particular similarity values for each pair of objects (Bansal et al., 2002). More specifically, the objective of correlation clustering is that of finding the clustering $\hat{\Pi}$ which minimizes the correlation cost function $\mathrm{CC}(\Pi)$:

$$\hat{\Pi} = \arg\min_{\Pi} \mathrm{CC}(\Pi) \qquad \mathrm{CC}(\Pi) = \sum_{\substack{x_i, x_j \in \mathcal{X} \\ x_i \sim x_j}} d_{ij} + \sum_{\substack{x_i, x_j \in \mathcal{X} \\ x_i \nsim x_j}} (1 - d_{ij})$$

Even if the correlation clustering problem is known to be NP-complete (Bansal et al., 2004, §3), Gionis et al. propose three different algorithms to find approximate solutions:

Agglomerative (AGGLO) applies the HAC algorithm with UPGMA rule on the distance matrix. The dendrogram is then cut at the level where the distance between the two merged nodes is larger than 0.5, in order to obtain the output clustering.

Balls (BALLS) is the only one of the three methods to require an input parameter, α. The algorithm starts by sorting the objects in decreasing order of their sum of distances to all other objects in the dataset. It then takes each object x_i in turn and, if not already clustered, the following steps are taken:

1. The set $\mathcal{X}_i$ of unclustered objects whose distance to the current one x_i is less than $\frac{1}{2}$ are collected.
2. If the average distance between the objects is $\mathcal{X}_i$ and x_i is less than α, a new cluster $\pi_c = \{x_i\} \cup \mathcal{X}_i$ is created; otherwise, x_i is added to the clustering as a singleton $\pi_c = \{x_i\}$.

Even if the output of the algorithm is sensitive to the value of the parameter α, the authors provide a theoretical proof of its approximation ratio for the particular case $\alpha = \frac{1}{4}$, which is the one we have used in our experiments.

Furthest (FURTH) is a divisive algorithm, which starts by placing all objects in the same cluster. The pair of objects with the largest distance between them are selected as cluster centroids, and the remaining objects are reassigned to the cluster to whose centroid the distance is minimal.

The process is repeated, at every step selecting as centroid of a new cluster the object whose sum of distances to the previous centroids is maximal, and reassigning the objects for which this new centroid has become the closest one. The decrease of the correlation clustering cost function $\mathrm{CC}(\Pi)$ is used as terminating condition: when the newly found clustering presents a cost larger than the previous one, the process is stopped, and the latter is returned as output clustering.

Additionally, Gionis et al. propose the use of a local search procedure (LOCAL) on the output from the three methods, in order to refine the obtained clustering. The procedure performs hill climbing to minimize the correlation clustering cost function $\mathrm{CC}(\Pi)$, using the movement of a single object to another cluster as search operator.

We will denote the output from the combination of AGGLO, BALLS or FURTH with LOCAL as AGGLO+L, BALLS+L and FURTH+L.

3.5 Evaluation

In order to assess and compare the performance of the different clustering methods and strategies for ensemble generation presented in the previous section, we have performed a series of experiments on real-world data.

Next sections give details about the evaluation procedure. Section 3.5.1 describes the used datasets. Next Section 3.5.2 describes the evaluation protocol, including the considered metrics, and, finally, Section 3.5.3 exposes and discusses the obtained results.

Collection	Docs.	Cats.	Terms	Collection	Docs.	Cats.	Terms
Apw	5,000	11	27,366	Reu	2,545	10	6,734
Efe	1,979	6	10,334	Smt	5,467	4	11,950
Lat	5,000	8	31,960	Swb	2,682	22	11,565

Table 3.2: Clustering collection sizes

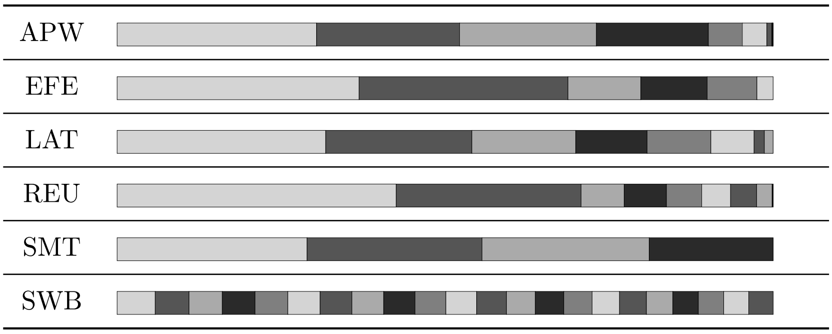


Figure 3.3: Distribution of categories within collections

3.5.1 Data

Six different real-world English document collections have been used in our experiments:

APW The Associated Press (year 1999) subset of the AQUAINT collection[9]. The document category assignment is indicated by a CATEGORY tag.

EFE A collection of newswire documents from year 2000 provided by the EFE news agency.

LAT The Los Angeles Times subset of the TREC-5 collection[10]. The categories correspond to the newspaper desk that generated the article, as done by Zhao and Karypis (2004).

REU A subset of the Reuters-21578 text categorization collection, including only the ten most frequent categories. Similarly to previous work, we used the test partition of the ModApte split (Surdeanu et al., 2005).

SMT A collection previously developed and used for the evaluation of the SMART information retrieval system.

SWB A subset of the Switchboard conversational speech corpus, which contains the 22 topics which were treated in more than fifty conversations. Each side of the conversation was considered a separate document.

Following other works (Zhao and Karypis, 2004; Surdeanu et al., 2005), the documents were preprocessed by discarding stop words and numbers, converting all words to lower case, and removing terms occurring in a single document. Table 3.2 lists relevant collection characteristics after this preprocessing step (number of documents, categories and terms).

Figure 3.3 contains a graphical representation of the distribution of documents in categories within each collection. We can see how, especially in Apw, Lat and Reu, there is set of few categories which covers most of the documents, whereas other categories have a rather marginal presence; and how, on the contrary, Smt and Swb presents almost-equally represented categories—the latter, in particular, containing a large number of them.

[9] Due to memory limitations in our test machines, the collection was reduced to the first 5000 documents.

[10] For the same reason as in APW, again only the first 5000 documents were selected.

3.5.2 Protocol

The quality of the clusterings produced by the different approaches is measured using the metrics of purity, inverse purity and F1. These metrics have been widely used to evaluate the performance of document clustering algorithms (Zhao and Karypis, 2004), and are based in comparing the clustering to a *gold standard*, i.e., a partition which is considered the *true* one. Meilă (2003) refers to their use as evaluation "*by set matching*".

Given a gold partition $\hat{\Pi}$ of the objects in $\mathcal{X}$ into $\hat{k}$ categories $\hat{\Pi} = \{\hat{\pi}_1 \dots \hat{\pi}_{\hat{k}}\}$, these metrics can be defined as:

Purity (Pur) evaluates the degree to which each cluster contains objects from a single category. The purity of a cluster is the fraction of the cluster size which its *dominant* category (i.e., the one to which the most objects in the cluster belong) represents. The overall purity of a clustering is the average of all cluster purities, weighted by cluster size (Zhao and Karypis, 2001):

$$\begin{aligned}\mathrm{Pur}(\pi_c\,;\hat{\Pi}) &= \frac{\max_{\hat{\pi}_d \in \hat{\Pi}} |\pi_c \cap \hat{\pi}_d|}{|\pi_c|} \\ \mathrm{Pur}(\Pi\,;\hat{\Pi}) &= \frac{\sum_{\pi_c \in \Pi} |\pi_c| \cdot \mathrm{Pur}(\pi_c\,;\hat{\Pi})}{\sum_{\pi_c \in \Pi} |\pi_c|}\end{aligned}$$

Inverse Purity (IPur) evaluates the degree to which the objects from a category are together in a single cluster. The inverse purity of a category is the fraction of the category size which its *dominant* cluster (i.e., the one which contains the most objects from the category) represents. The overall inverse purity is the average of all category inverse purities, weighted by category size.

$$\begin{aligned}\mathrm{IPur}(\Pi\,;\hat{\pi}_d) &= \frac{\max_{\pi_c \in \Pi} |\pi_c \cap \hat{\pi}_d|}{|\hat{\pi}_d|} \\ \mathrm{IPur}(\Pi\,;\hat{\Pi}) &= \frac{\sum_{\hat{\pi}_d \in \hat{\Pi}} |\hat{\pi}_d| \cdot \mathrm{IPur}(\Pi\,;\hat{\pi}_d)}{\sum_{\hat{\pi}_d \in \hat{\Pi}} |\hat{\pi}_d|}\end{aligned}$$

F1 tries to be a global performance score, and is calculated as the harmonic mean of purity and inverse purity (van Rijsbergen, 1974):

$$\mathrm{F}_1(\Pi\,;\hat{\Pi}) = \frac{2 \cdot \mathrm{Pur}(\Pi\,;\hat{\Pi}) \cdot \mathrm{IPur}(\Pi\,;\hat{\Pi})}{\mathrm{Pur}(\Pi\,;\hat{\Pi}) + \mathrm{IPur}(\Pi\,;\hat{\Pi})}$$

Following common practice, the figures for these metrics will always be presented as percentages.

Among the proposed components, only the MAJOR generation strategy is non-deterministic. In order to reduce the impact of randomness in its judgements, we have carried out 5 different runs for each configuration and dataset, and reported the average measures.

Given that direct commensurability of results across different collections cannot be assumed, we have also considered *relative performance* metrics. To this end, the clustering with the best F1 score has been identified in each collection, and the precision, recall and F1 values of each method have been divided by those of this reference one. The mean of these relative values is reported.

Nevertheless, relative metrics are only shown in an informative fashion. For statistical soundness, method performance comparison across multiple datasets is carried out using the Bergmann-Hommel non-parametric hypothesis test (Bergmann and Hommel, 1988). Being non-parametric, the test judges the relative performances of the different methods with respect to each other, rather than their absolute scores or score differences. Recently, works such as that of Demšar (2006) have advocated for non-parametric tests to assess significance in machine learning tasks, as the assumption of metric commensurability across datasets, required by usual parametric tests such as Student or ANOVA, is often broken. The use of the Bergmann-Hommel test in particular has been recommended by García and Herrera (2008).

The graphical presentation of the significance tests is that introduced by Demšar (2006): methods are placed along the horizontal axis according to their average ranks across datasets, and those for which no statistically significant difference can be found are joined by thick bars.

3.5.3 Results

Three series of experiments have been carried out in order to assess different aspects of the clustering and ensemble generation process. The next three sections, 3.5.3.1 to 3.5.3.3, describe each one of them.

3.5.3.1 Generation Parameters

The first series of experiments performed had as goal to determine the influence of the parameters R and k_{max} on the performance of clustering using the ensembles generated by the Major strategy. With this purpose, ensembles were generated using the values:

$$R \in \{10, 20, 30, 40, 50\} \qquad k_{max} \in \{5, 10, 20, 50\}$$

and combined using the three combination methods (Agglo, Balls, Furth), enhanced by Local search. The plots for the three considered measures, as a function of the parameter values and relative to the best clustering, are shown in Figure 3.4.

On the one hand, plots (a), (c) and (e) show the influence of the number of clusterings in the ensemble, R, in the measures of purity, inverse purity and F1 of the final clustering. It can be observed how the influence of this parameter is small: even if there is a slight increase in both purity and inverse purity—and hence also in F1—at the low end of the curves, around $R = 30$ the results are already quite stable, and do not show significant changes before reaching $R = 50$. It thus seems that, once a sufficient number has been reached, an increase of the number of clusterings does not produce significant changes on the quality of the output.

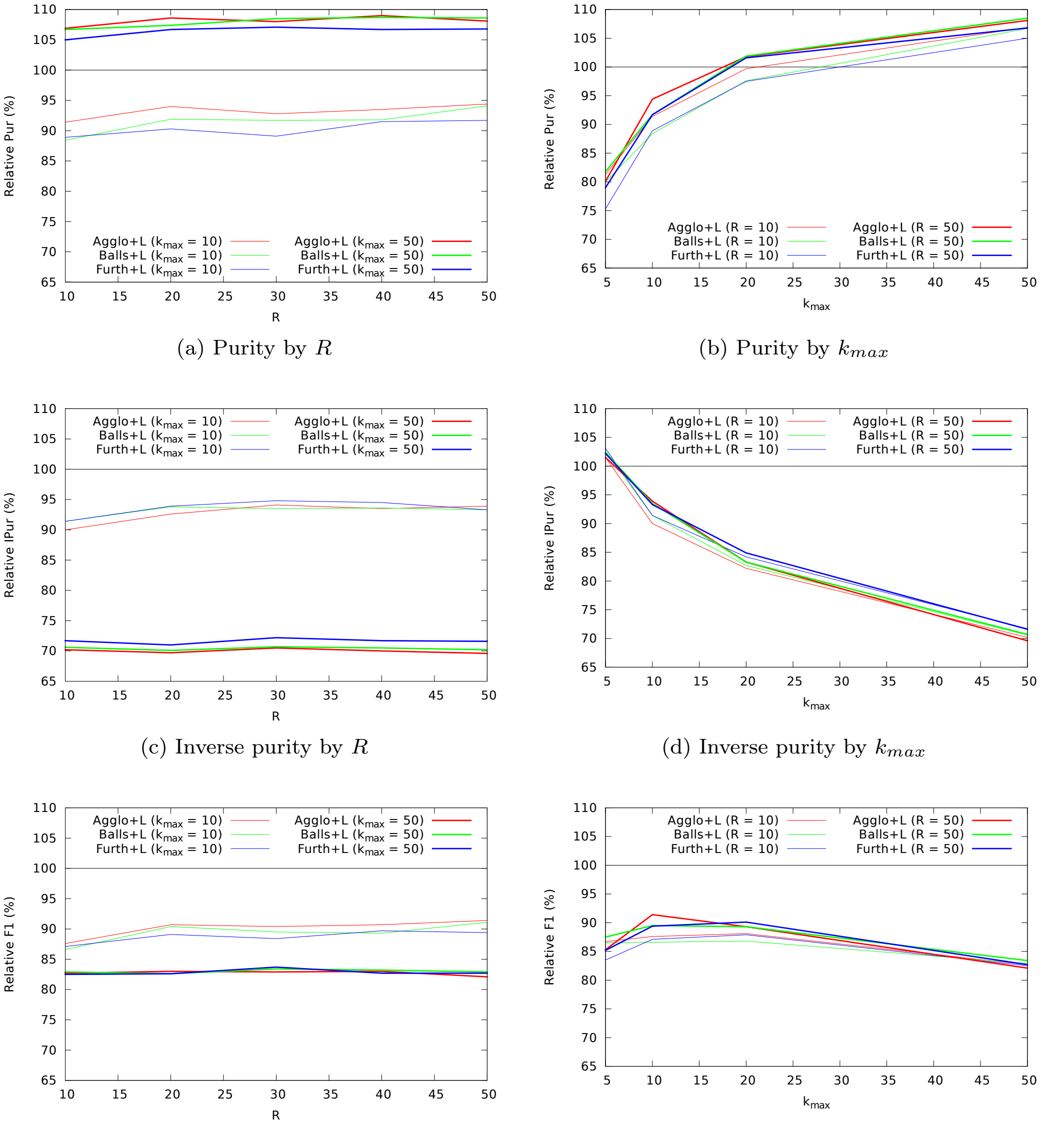
On the other hand, plots (b), (d) and (f) show the influence of the maximum number of clusters, k_{max}, on the final clustering. The picture here is radically different, as the results change dramatically with the value of k_{max}. More specifically, purity increases and inverse purity decreases with larger values of k_{max}.

The most immediate interpretation of these trends is the fact that larger values of k_{max} will allow larger numbers of clusters k_r in each one of the clusterings in the ensemble. The k_r do tip the scales in the purity–inverse purity trade-off: a clustering with more clusters will tend to favour purity, and a less fragmented one, inverse purity. Even if the use of combination methods reduces the effect of the tuning of this parameter on the final clustering, it is clear that it does not remove its influence completely.

Globally, the maximum value in terms of F1 is achieved at the relatively low value of $k_{max} = 10$. However, Figure 3.5 contains two additional plots, that put into relief the dependency of k_{max} on the data to be clustered. Plot (a) contains the values for F1 as a measure of k_{max} for clusterings over each one of the six considered document collections (using the value $R = 50$ for the other parameter, and the Agglo+L method). It can be observed how not only the location of its maximum, but also the behavior of F1 as a function of k_{max}, changes significantly from collection to collection. Plot (b) further illustrates this, showing the correlation between the number of categories in the collection, $\hat{k}$, and the optimal value of k_{max} in terms of F1.

These results seem to contradict previous works (e.g., Ghosh et al., 2002), in which ensembles of clusterings with a much larger number of clusters than those actually present in the data have been used successfully. However, we believe that one possible explanation to this behavior may lie in the fact that we are using EM as weak clustering algorithm: the number of parameters in the mixture model increases with the number of clusters, and hence their estimation by the EM algorithm becomes less accurate as this number increases, degradating the clustering quality (a particular case of the "*curse of dimensionality*"; Bellman, 1961). The obtained clusterings are hence rendered less useful for combination within an ensemble. However, it is interesting to note that these results do agree with those reported by some other authors—such as Topchy et al. (2005), who, when generating weak clustering ensembles using k-means, did not consider values of k_{max} beyond the interval $k_{max} \in \{2 \dots 10\}$.

The determination of a suitable value of k_{max} hence seems to remain the main drawback of the Major strategy—at least, when using EM as inner clustering algorithm. For the rest of this section, the results presented for Major will be those using the parameter setting which obtains the best F1 measures, namely $R = 50$, $k_{max} = 10$.



(a) Purity by R

(b) Purity by k_{max}

(c) Inverse purity by R

(d) Inverse purity by k_{max}

(e) F1 by R

(f) F1 by k_{max}

Figure 3.4: Influence of ensemble generation parameters in document clustering results (Major strategy)

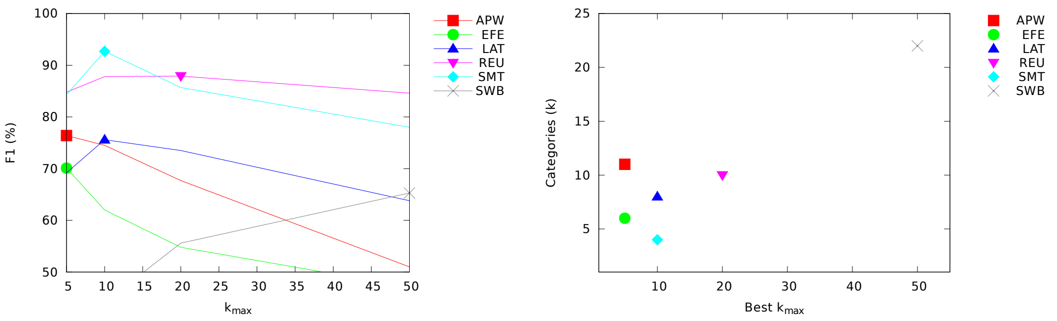
3.5.3.2 Combination Method

The second group of experiments attempt to determine the ensemble combination method which is the most suitable to our purposes. Their results are summarized in Table 3.3, which contains the purity, inverse purity and F1 values obtained, on each collection, by the three combination methods (Agglo, Balls, Furth), with and without Local. Following the choice taken at the end of the previous section, the ensembles were generated using Major[11] with parameters $R = 50$ and $k_{max} = 10$.

The first clear conclusion at the light of these results is that Local does improve the results obtained by the combination methods alone: in all cases, Local improves or does not significantly change the purity scores, and in most cases it also improves inverse purity. Overall, only in the case of Furth over Efe the F1 measure is lower with Local than without it. This is in agreement with the results obtained by Gionis et al. (2005).

On the other hand, the performances of Agglo, Balls and Furth are much more similar

[11] Similar results were obtained using the Minor ensemble generation strategy.



(a) F1 by k_{max}

(b) Correlation between optimal k_{max} and $\hat{k}$

Figure 3.5: Optimal values of k_{max} per collection
(Major strategy, $R = 50$, Agglo+L method)

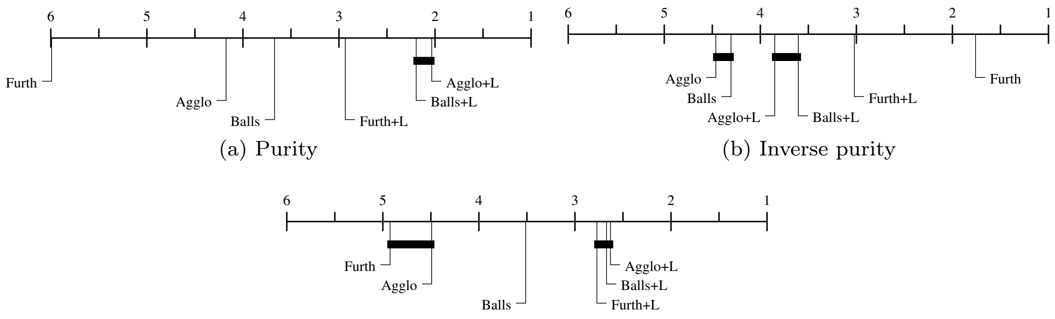
		-			+Local		
		Pur	IPur	F1	Pur	IPur	F1
Apw	Agglo	76.8	70.0	**73.2**	79.5	70.1	**74.5**
	Balls	78.5	67.5	**72.4**	79.6	70.1	**74.5**
	Furth	52.8	88.8	**66.1**	80.7	70.1	**75.0**
Efe	Agglo	74.7	51.9	**61.1**	75.4	52.6	**62.0**
	Balls	75.3	48.7	**59.0**	75.5	51.6	**61.2**
	Furth	59.4	72.5	**65.2**	75.4	52.6	**61.9**
Lat	Agglo	68.9	77.5	**72.9**	73.3	78.6	**75.6**
	Balls	67.7	80.0	**73.0**	68.4	77.0	**72.3**
	Furth	52.4	90.3	**66.2**	66.9	75.3	**70.7**
Reu	Agglo	84.4	89.9	**87.1**	85.5	90.1	**87.8**
	Balls	85.9	85.2	**85.5**	85.6	89.8	**87.7**
	Furth	71.9	91.3	**80.4**	85.1	90.0	**87.5**
Smt	Agglo	90.3	87.8	**89.0**	93.4	92.1	**92.7**
	Balls	87.8	82.9	**85.3**	93.4	92.1	**92.8**
	Furth	71.7	96.6	**82.3**	89.7	93.3	**91.2**
Swb	Agglo	28.4	97.4	**43.8**	28.6	96.9	**44.1**
	Balls	28.0	94.7	**42.9**	31.0	97.0	**46.7**
	Furth	17.2	96.1	**28.9**	26.4	96.7	**41.3**

Table 3.3: Comparison of ensemble combination methods for document clustering
(Major strategy, $R = 50$, $k_{max} = 10$)

between them. Without Local, Furth tends to obtain better inverse purity than the other two methods, at the expense of worse purity and, often, F1. However, after applying Local, the differences become much dimmer, and only in Lat and Swb do they exceed a 1%-gap between the lowest and the highest F1 scores—but never going beyond the 5%-difference.

The Bergmann-Hommel significance tests for each metric, displayed in Figure 3.6, confirm these trends. As seen in plot (a), all differences in purity are significant, except that between Agglo+L and Balls+L. The two methods, as well as their Local-less counterparts, do not show significant differences in terms of inverse purity, but plot (b) shows how they fall behind Furth and Furth+L. Overall, plot (c) depicts how the thee Local-enhanced methods obtain the best F1 values with no significant differences between them, but with a significant improvement with respect to the Local-less ones.

At the light of the outcome of the significance tests, among the best performing methods we will use Agglo+L for the rest of our experiments—being the algorithm which achieves the best



(a) Purity

(b) Inverse purity

(c) F1

Figure 3.6: Bergmann-Hommel tests for the proposed combination methods

purity and F1 score.

3.5.3.3 Overall Comparison

The third and last group of experiments provide an overall comparison of the performance of both individual and ensemble-based methods. Their results can be seen in Table 3.4.

Concerning individual methods, we can see how, firstly, methods Geo and Info tend to find clusterings with more purity than inverse purity, whereas Hi tends to favour inverse purity above purity. Globally, the best results are obtained with Geo in all collections but Efe, followed by Hi and, last, Info, which in some cases shows a significantly lower performance (e.g., collections Apw and Smt).

Regarding the ensemble methods, Major outperforms Minor in almost all collections and measures. Only in the Swb collection do the results of the former fall significantly behind those of the latter, because of an underestimation of the number of clusters which causes an exceedingly low value of purity. In the rest of the cases, the measures obtained by Major are always a few points above those of Minor, short of purity in the Lat collection—which, however, Minor obtains at the expense of a much larger decrease in inverse purity and F1.

Overall, the best two methods are Major and Geo, which obtain the best values of F1 in most of the collections, followed after a small gap by Minor. Methods Hi and Info fall much behind the performance of the other approaches, mostly because of their bad results in terms of purity and inverse purity, respectively. In the case of the former, this low purity is caused by its too shifted trade-off for inverse purity; but in the latter, the low inverse purity does not come with high values of purity—so the conclusion is that the approach is flawed at some point.

It is also interesting to note that, even if only one of the clusterings in the ensemble (namely, that produced by Geo) consistently provides good quality clusterings across all collections, the results of Minor are either better or only slightly below those of Geo in most of the cases. Only on the Apw and Swb collections do the values of F1 for Minor fall more than 1% below those of Geo.

The Bergmann-Hommel significance tests for each metric are displayed in Figure 3.7. In this case, the relatively low number of collections makes the tests less informative than those in the previous section—only the differences in purity between Hi and each one of Major, Minor and Geo, as well as that in F1 between Major and Info, are statistically significant.

Finally, it is worth to notice that the number of clusters in the output clustering, k, often differs considerably from the number of categories in the collection, $\hat{k}$, without this having an impact in the clustering quality in terms of the considered metrics. This is caused by two factors:

- On the one hand, and as mentioned in Section 3.5.1, the Apw, Efe, Lat and Reu collections contain a few large categories and many small ones. In these cases, the descent in purity caused by merging small categories with a large one will be small. This explains, for instance, why Geo obtains a good purity value in Reu, in spite of generating a clustering with a number of clusters $k = 6$ instead of the correct number of categories $\hat{k} = 10$.

		Pur	IPur	F1	k	$\hat{k}$
Apw	Geo	78.2	72.7	**75.3**	10	**11**
	Info	72.3	56.1	**63.1**	8	
	Hi	63.2	88.3	**73.7**	3	
	Minor	74.1	69.7	**71.8**	19.0	
	Major	79.5	70.1	**74.5**	60.6	
Efe	Geo	73.5	52.2	**61.1**	12	**6**
	Info	63.9	52.7	**57.8**	5	
	Hi	59.7	63.3	**61.4**	4	
	Minor	69.8	52.6	**60.0**	14.0	
	Major	75.4	52.6	**62.0**	69.0	
Lat	Geo	77.7	59.3	**67.3**	14	**8**
	Info	75.4	60.9	**67.4**	7	
	Hi	66.4	68.0	**67.2**	6	
	Minor	79.0	59.3	**67.7**	40.0	
	Major	73.3	78.6	**75.6**	27.2	
Reu	Geo	84.4	92.5	**88.2**	6	**10**
	Info	77.0	75.7	**76.4**	6	
	Hi	73.0	85.8	**78.9**	4	
	Minor	85.0	89.4	**87.1**	13.0	
	Major	85.5	90.1	**87.8**	18.2	
Smt	Geo	91.6	79.8	**85.3**	6	**4**
	Info	89.4	58.4	**70.6**	9	
	Hi	70.6	96.8	**81.6**	3	
	Minor	92.6	89.8	**91.2**	18.0	
	Major	93.4	92.1	**92.7**	20.6	
Swb	Geo	68.9	93.7	**79.4**	15	**22**
	Info	37.9	90.7	**53.4**	8	
	Hi	15.2	91.9	**26.1**	3	
	Minor	53.0	88.9	**66.4**	22.0	
	Major	28.6	96.9	**44.1**	10.4	

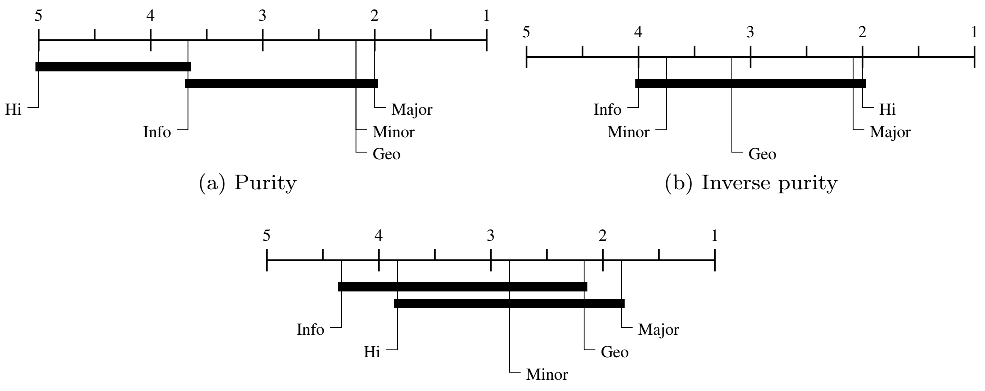
Table 3.4: Overall comparison of methods for document clustering

- On the other hand, the ensemble method tends to detect a large number of small clusters. These clusters will correspond in most cases to *outliers*—i.e., objects which "*[appear] to deviate markedly from other members of the sample in which [they occur]*" (Grubbs, 1969). Because of the small size of the outlier clusters, their separation from the other objects to whose category they belong does not produce a drop in inverse purity. This explains why the clustering obtained by Major on Apw and Efe has a much larger number of clusters than those of other methods ($k = 60.6$ and $k = 69.0$, respectively), yet obtains comparable values of inverse purity.

  In fact, given the attention deserved to the problem of *outlier detection* (Hodge and Austin, 2004), we believe the identification of such outlier clusters should not be seen as a drawback, but rather as a feature, of the combination algorithm.

3.6 Conclusions

In this chapter, we have introduced the problem of clustering, and reviewed some of the most prominent works for the particular task of unsupervised clustering.



(a) Purity

(b) Inverse purity

(c) F1

Figure 3.7: Bergmann-Hommel tests for the proposed clustering methods

In order to perform a comparison between individual and ensemble methods for unsupervised document clustering, as well as of ensemble generation strategies, we have formalized the core concepts related to clustering, and presented a number of approaches to solve the problem. We have then been able to carry out a number of experiments to assess the performance of the different methods over several real-world document datasets. At the light of the obtained results, we believe there is a number of relevant conclusions that can be drawn.

The main conclusion of our work is that ensemble methods do perform competitively for unsupervised document clustering. In particular, combination of large ensembles obtained through randomization of a supervised clustering algorithm (what we have named the Major strategy) outperform individual approaches (Geo, Info and Hi), as well as a combination of a reduced ensemble, containing few but better informed unsupervised clustering algorithms (Minor strategy).

However, Major depends on the tuning of two parameters: the number of clusterings in the ensemble, R, and the maximum number of clusters per clustering, k_{max}. The experiments have assessed the robustness of the strategy with respect to the former, but also stated its sensitivity on the latter. Nevertheless, using a fixed rule-of-thumb value of $k_{max} = 10$ has produced clusterings whose quality is above that of all other compared methods.

As mentioned in the introduction, the conclusions of our work on unsupervised document clustering have significantly influenced the path of further research, especially in the task of minority clustering. Chapter 5 will delve into this problem, and will provide insights of how this influence has crystallized.

4

Collaborative Learning

> "They're trying to kill me," Yossarian told him calmly.
> "No one's trying to kill you," Clevinger cried.
> "Then why are they shooting at me?" Yossarian asked.
> "They're shooting at everyone," Clevinger answered. "They're trying to kill everyone."
> "And what difference does that make?"
>
> Joseph Heller
> *Catch-22*

This chapter presents an empirical validation of the sequential and collaborative schemes of clustering and IE pattern learning combination. With this goal, we start by reviewing an existing manually-seeded weakly supervised approach, and then make a two-fold proposal to, on the one hand, reduce its supervision using clustering techniques, and, on the other, replace the used pattern formalism by a higher-coverage and more flexible one.

Section 4.1 starts by giving a motivation of the ideas behind our work, and Section 4.2 states the problem of relation detection as we consider it in this and following chapters. The proposed approaches to solve the problem are described in detail in Section 4.3. To determine their validity, we have carried out two different evaluations. Section 4.4 presents the first one, indirect and through the task of text categorization; Section 4.5 presents the second one, direct and on an actual relation detection task. Finally, Section 4.6 draws conclusions of our work.

AFTER DEVELOPING and evaluating a small panorama of unsupervised clustering methods in the previous chapter, we are now in a position to use clustering techniques to tackle the problem of learning for IE. Following the polytomy presented in Section 1.4, in this chapter we propose and evaluate two approaches to pattern learning for relation detection—a sequential one and a collaborative one—which incorporate clustering as a central component.

4.1 Collaborative Clustering and Pattern Learning

The natural scheme for collaborative and sequential combination of clustering and pattern learning—i.e., the interleaving of *document* clustering and pattern acquisition—is based on the assumption that there does exist a mapping from extraction patterns to document categories (or clusters). In particular, the patterns must extract entities, relations or events which are specific to scenarios occurring in a single domain. The *one-domain-per-pattern* requirement has been previously assumed by a number of authors (e.g., Yangarber et al., 2000; Stevenson, 2004; Surdeanu et al., 2006), in particular, to justify indirect IE pattern evaluation through text classification: "*we can judge the quality of [a] pattern set based on the quality of the documents that the patterns match*" (Yangarber, 2003).

Even though this assumption can hold in the case of domain-specific relations (such as those in the MUC scenarios that the mentioned works consider), it may on the contrary become unrealistic when trying to recover more general binary relations, which can be transverse to any categorization of the documents containing them. The goal of this chapter is hence not only the validation of a particular approach to combine clustering and pattern learning, but also of the whole framework which depends on this assumption.

In order to have an unsupervised collaborative approach—on which to perform our validation—at our disposal, we have decided to build one based on top of the previously reviewed weakly supervised system of Surdeanu et al. (2006). Recalling Section 2.2.3, this approach requires a document collection, unannotated but for a small number of seed documents—i.e., documents which have been judged relevant to each one of the domains present in the collection. These seed documents are then used to bootstrap a co-training loop between an iterative clustering algorithm and a decision-list learner, which use the words and patterns present in the documents, respectively, to cluster them. After a number of iterations, patterns which have been selected as features by the decision list learner are returned—as they will hopefully express entities, relations or events specific to their domain. In particular, predicate-argument patterns are used, as proposed by Yangarber et al. (2000).

In the work of Surdeanu et al., this co-training approach is compared with a straight bootstrapping of the decision-list learner on the given manual seeds, using self-learning (à la Yarowsky, 1995). This *sequential* approach, and the co-training or *collaborative* one, are depicted in Figures 4.1a and 4.1b, respectively. We believe this architecture can be enhanced to build an unsupervised system by replacing the initial step by an unsupervised document clustering process, thus lifting its dependency on a manual set of seeds. The resulting sequential and collaborative schemes are graphically represented in Figures 4.1c and 4.1d, respectively.

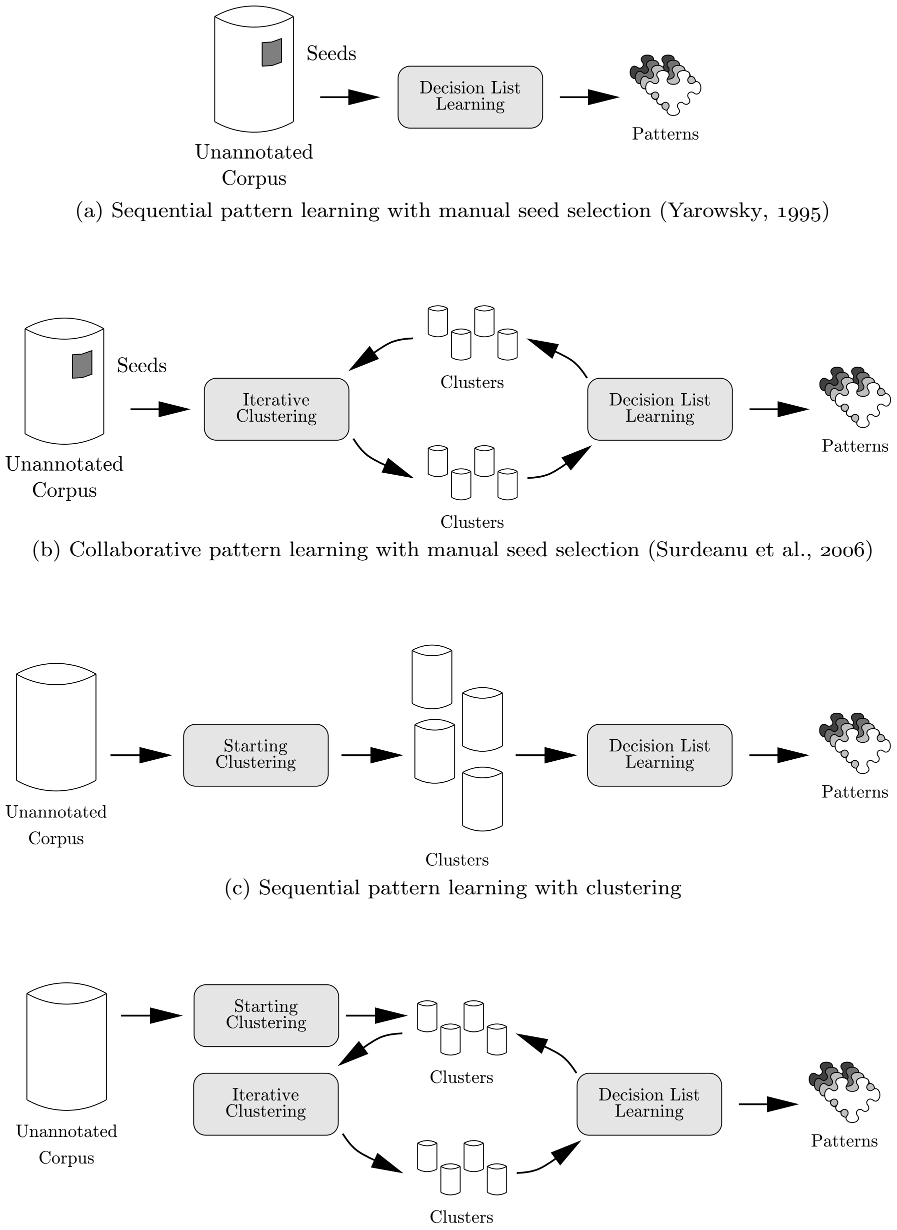
Moreover, the use of predicate-argument tuples restricts the capturable relations to those expressed using verbal grammatical constructions. Given that, for instance, only 21.2% of the relation mentions annotated on the ACE-2005 corpus follow a *verbal* lexical condition, this restriction effectively cramps the approach, lowering its recall. Our proposal becomes two-fold, as we additionally propose to replace the predicate-argument tuples by a more expressive formalism, based on conjunctions of binary features.

4.2 Problem Definition

As already presented in Section 1.1.1, the problem of *relation detection* is that of identifying the tuples of entities occurring in a document which are linked by some kind of relation at the instance level. In particular, in this work we have focused on binary relations between pairs of entities.

Our approaches to relation detection are all based on the transformation of the task into a binary decision problem: "*Given a pair of entities E_1 and E_2 which co-occur in a sentence, are they related or unrelated?*". This framework is depicted in Figure 4.2. Even if it suffers from the obvious drawback that it is unable to capture relations between entity mentions occurring in different sentences, such inter-sentence relation mentions are rare, and can therefore be disregarded without a significant loss in detection recall. We thus believe the scheme to be powerful enough for our purposes.

Following ACE terminology (see Section 2.1.2), we will be evaluating the task at the *mention* level. The problem will hence be determining if there is a relation expressed between two entity mentions (i.e., references in a linguistic fragment)—as opposed to the task of determining whether such relation exists between the actual entities (considered as the elements outside the discourse to which the mentions refer). The distinction is important because, occasionally, a pair of entities may be linked at the entity level, but not at the mention level. Those are cases in which the linguistic context surrounding the mentions does not allow inferring the existence of the particular relation between them.



(a) Sequential pattern learning with manual seed selection (Yarowsky, 1995)

(b) Collaborative pattern learning with manual seed selection (Surdeanu et al., 2006)

(c) Sequential pattern learning with clustering

(d) Collaborative pattern learning with clustering

Figure 4.1: Approaches for decision-list-based IE pattern learning

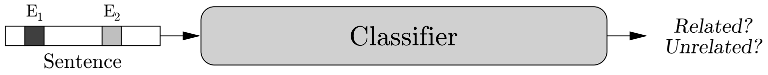
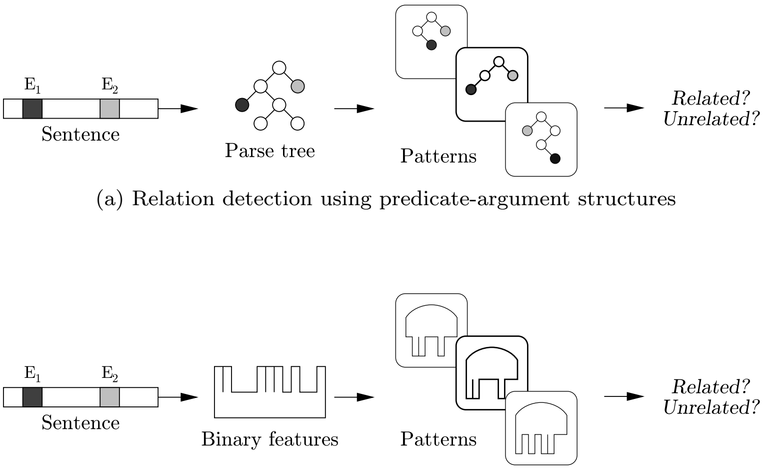


Figure 4.2: Relation detection as a binary decision problem



(a) Relation detection using predicate-argument structures

(b) Relation detection using binary feature conjunctions

Figure 4.3: Approaches for relation detection

4.3 Our Approaches

In the approaches in this chapter, the relatedness classification problem is solved using a pattern base, whose patterns are to be matched against the context in which the considered pair occurs. For the case of predicate-argument structures, this matching is determined from properties of the sentence parse tree (Figure 4.3a); whereas for binary feature conjunctions, it is a function of the set of predefined binary features which are active in the context (Figure 4.3b). In the case in which at least one of the patterns matches, we consider the pair of entities to be related; otherwise, they are deemed unrelated. The process of sequential or collaborative clustering and pattern learning has as a goal the construction of one such pattern base.

Next sections detail each one of the components involved in the learning process, starting with candidate pattern generation in Section 4.3.1. Next Section 4.3.2 describes the collaborative and sequential learners we have employed. Finally, Sections 4.3.3 and 4.3.4 describe the considered clustering and decision-list learning algorithms, respectively.

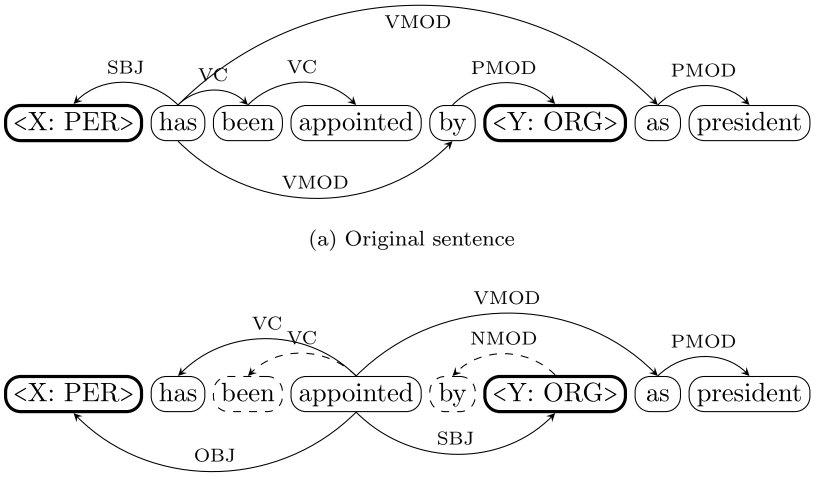
4.3.1 Candidate Pattern Generation

The first step of both approaches is the generation of the candidate patterns. Being unsupervised, the procedure considers *all* sentences in the corpus, and gathers *all* entity pairs which co-occur in it. All possible patterns within the considered formalism that join the pair are added to form the candidate set. However, given that relations are usually expressed using short-distance constructions[1], in the learning step only those pairs whose distance, measured in tokens between them, is less than a certain threshold are considered[2].

Additionally, during pattern generation a *bag-of-patterns* representation of the documents is built, analogous to the bag-of-words one (see Section 3.4.1.1) but using the candidate pattern set as dictionary. In this view, a document will hence be represented by the *frequency* of each one of the candidate patterns—i.e., the number of entity pairs in the document which are linked by the pattern.

[1] For instance, in more than 95% of the annotated relation mentions in the ACE-2003, ACE-2004 and ACE-2005 corpora the two entity mentions are separated by less than 11 tokens.

[2] It is important to note that this threshold is only used during the learning step, and not when classifying new entity pairs.



(a) Original sentence

(b) Normalized sentence

```
   sv:ORG:appoint
  svo:ORG:appoint:PER
svoio:ORG:appoint:PER:as:president
 svio:ORG:appoint    :as:president
   vo    :appoint:PER
 voio    :appoint:PER:as:president
  vio    :appoint    :as:president
```

(c) Generated predicate-argument structures

Figure 4.4: Passive voice and perfect tense normalization

Due to computational issues, this conceptually simple scheme cannot be directly applied to the two considered formalisms used in our experiments. Next two Sections 4.3.1.1 and 4.3.1.2 describe each one of them in detail, and discuss the restrictions and changes that have been needed in order to render the process computationally feasible.

4.3.1.1 Predicate-Argument Structures

Predicate-argument structures are tuples whose slots contain, for a given verbal predicate, which word acts as predicate verb, and which words or entities are the predicate arguments. The structures considered by Surdeanu et al. (2006) involve *subjects* (`s`), *objects* (`o`) and *indirect objects* (`io`; actually, any prepositional complement) for a given verb (`v`). For this last argument type, the preposition linking the argument to the verb also takes part in the pattern.

The candidate generation step only needs to consider entity pairs where both entities are direct arguments of the same verbal predicate. However, also following Surdeanu et al., a number of syntactic normalizations are applied to the dependency parse tree of the sentences before pattern generation and matching. In our experiments, the normalization process follows a number of handwritten rules which account for perfect and continuous verbal tenses, modal verbs and passive constructions. The goal of these transformations is to place the verb which carries the semantics of the predicate, rather than the aspect, in its main position, as well as to take into account the subject-object inversion of the passive voice. An example of the effect of such normalizations is represented in Figure 4.4.

After parse-tree normalization, for each verb in the sentence all possible structures are generated. In them, if the corresponding arguments are an entity, the slots are filled with the entity type; otherwise, the word lemma is used. Figure 4.4c contains the patterns that are generated from the sample sentence of Figure 4.4a.

We will refer to the feature set generated by these predicate-argument structures as `p:a`.

	w:t	c:l	w:t+c:l
Structure-based			
Distance between the pair is %d words	●	●	●
Distance between the pair is %d chunks	·	●	●
Left/rightmost entity is of type %t	●	●	●
Word-based			
Word %d positions before/after the left/rightmost entity. . .			
. . . has POS tag %t	●	·	●
Chunk-based			
Chunk %d positions before/after that containing the left/rightmost entity. . .			
. . . has type %t	·	●	●
. . . has a head with lemma %l	·	●	●

Table 4.1: Feature patterns used by feature sets

Word	**<X: PER>**	has	been	appointed	by	**<Y: ORG>**	as	president
Position	left	after:left:1 before:right:4	after:left:2 before:right:3	after:left:3 before:right:2	after:left:4 before:right:1	right	after:right:1	after:right:2
Tr.+Val.	type=PER	tag=VBZ	tag=VBN	tag=VBN	tag=IN	type=ORG	tag=IN	tag=NN
Features	dist=5, left/type=PER, after:left:1/tag=VBZ, before:right:4/tag=VBZ, after:left:2/tag=VBN, before:right:3/tag=VBN, after:left:3/tag=VBN, before:right:2/tag=VBN, after:left:4/tag=IN, before:right:1/tag=IN, right/type=ORG, after:right:1/tag=IN, after:right:2/tag=NN							

Figure 4.5: Binary features generated by the `w:t` patterns for a sample sentence

4.3.1.2 Binary Feature Conjunctions

The alternate formalism we propose represents detection patterns as conjunctions of binary features, which capture lexical and syntactic properties of the context in which pairs of entities co-occur.

The generation of the binary features is performed according to a number of *feature patterns* defined a priori, and which consider traits such as the POS and lemmas of the linguistic elements (words, chunks. . .) in a window around the considered pair. Each feature will be composed of (at most) three parts:

- a **position**, which is the direction (`before`/`after`) and distance from the involved element to one (`left`/`right`) of the two entities in the pair;
- a **trait**, which is the property (`tag`, `lemma`. . .) of the element that the feature regards;
- the **value** of the *trait* of the element in that *position*.

Some structural features, such as the distance between the entities, consist of only the trait and value parts, but in all other cases, this triplet scheme is followed.

We have considered three feature sets for our experiments[3]:

w:t A syntax-oriented feature set which, in addition to the entity types and distance, only considers the POS tags of the words in the context.

c:l A lexicon-oriented feature set which, in addition to the entity types and distance, only considers the lemmas of the heads of the chunks in the context.

[3] This formalism will be revisited in Section 6.2.1, where more feature patterns will be considered.

w:t+c:l The union of the previous two feature sets.

The specific patterns used to generate them are listed in Table 4.1. Following Hassan et al. (2006), before pattern generation, entities are replaced by a single token with the entity type as POS. An example of the nature of the features that would be active for a given sentence can be found in Figure 4.5.

Given that the number of patterns which can be generated from a certain context grows exponentially with the number of active features in it, an explicit enumeration of them, as done for `p:a` structures, is no longer possible. Instead, the process of candidate generation is split into three steps:

1. In a first pass over the corpus, the vectors of features which are active in each entity pair context are generated and collected.
2. Next, a frequent-itemset-mining algorithm is used to find all maximal conjunctions of features whose frequency is above a certain threshold.
3. Finally, the frequent itemsets found in the previous step are matched against the features active in each context, in order to build the bag-of-patterns representation of the documents.

Among the different algorithms that have been proposed for frequent-itemset mining, we have implemented the one of Bayardo (1998) for our experiments.

4.3.2 Combination Schemes

As mentioned in previous sections, the combination of clustering and pattern learning using the components proposed in this chapter can be achieved using two different schemes: a sequential one and a co-training collaborative one. Next Sections 4.3.2.1 and 4.3.2.2 describe each one of them, respectively.

4.3.2.1 Sequential Combination (Self-Training)

The simpler sequential combination approach closely follows the self-training bootstrapping scheme proposed by Yarowsky (1995), represented procedurally in Algorithm 4.1.

The algorithm first applies a clusterer, and then learns a decision-list classifier L, bootstrapping it from the obtained clustering. The former uses only bag-of-words features, whereas the latter uses the bag-of-patterns representation of the document collection $\mathcal{X}$ (built from the generated pattern candidates).

The first step of the procedure is thus performing the starting clustering step (line 1). As the list is built in an incremental fashion, the next step is its initialization to an empty set (line 2). The main loop (line 3–5) then consists in performing a rule acquisition step to update the decision list (line 4) and refining the clustering using the newly found rules (line 5). This loop is iterated until convergence (line 3).

Pattern acquisition is a byproduct of decision-list learning: in a final step, the features which have been used as antecedent in the decision list rules are collected and returned (line 6).

We will denote the results obtained using a given seed set and the sequential learning scheme using the /S suffix appended to the seeds' name.

4.3.2.2 Collaborative Combination (Co-Training)

The considered co-training scheme is a standard one, following Blum and Mitchell (1998), and interleaves the training of two classifiers on different views of the dataset. The specific procedure for the task at hand is shown procedurally in Algorithm 4.2.

In this case, the algorithm is simultaneously training a probabilistic clustering model Θ and a decision-list L. The former regards the considered document collection $\mathcal{X}$ using only bag-of-words features, whereas the latter only considers the bag-of-patterns view (built from the generated pattern candidates).

Similarly to the sequential algorithm, the procedure begins by finding the starting clustering (line 1) and initializing the decision list to an empty set (line 2). The main loop (line 3–7) is split in two halves. The first one consists in using an algorithm to fit a clustering model from the

Algorithm 4.1 Sequential clustering and decision-list learning

```
Input: A word-based view of the dataset $\mathcal{X}^w$
Input: A pattern-based view of the dataset $\mathcal{X}^p$
Output: A set of extraction patterns $P$
Find the starting clustering $\Pi$
    $\Pi \leftarrow \text{cluster}_{\text{st}}(\mathcal{X}^w)$
Initialize the decision list $L$
    $L \leftarrow \varnothing$
While $\neg$ converged do
    Perform a decision-list learning step, and append the obtained rules to the decision list $L$
        $L \leftarrow L \cup \text{DL}(\Pi, \mathcal{X}^p)$
    Apply the decision list to update clustering $\Pi$
        $\Pi \leftarrow \text{update}(\Pi, \mathcal{X}^p ; L)$
Return the set of patterns that have been used as antecedent in the decision list rules $P$
    $P = \bigcup_{(p_a \rightarrow \pi_c) \in L} p_a$
```

Algorithm 4.2 Collaborative clustering and decision-list learning

```
Input: A word-based view of the dataset $\mathcal{X}^w$
Input: A pattern-based view of the dataset $\mathcal{X}^p$
Output: A set of extraction patterns $P$
Find the starting clustering $\Pi$
    $\Pi \leftarrow \text{cluster}_{\text{st}}(\mathcal{X}^w)$
Initialize the decision list $L$
    $L \leftarrow \varnothing$
While $\neg$ converged do
    Perform a clustering step to obtain a model $\Theta$
        $\Theta \leftarrow \text{cluster}(\Pi, \mathcal{X}^w)$
    Apply the clustering model to update clustering $\Pi$
        $\Pi \leftarrow \text{update}(\Pi, \mathcal{X}^w ; \Theta)$
    Perform a decision-list learning step, and append the obtained rules to the decision list $L$
        $L \leftarrow L \cup \text{DL}(\Pi, \mathcal{X}^p)$
    Apply the decision list to update clustering $\Pi$
        $\Pi \leftarrow \text{update}(\Pi, \mathcal{X}^p ; L)$
Return the set of patterns that have been used as antecedent in the decision list rules $P$
    $P = \bigcup_{(p_a \rightarrow \pi_c) \in L} p_a$
```

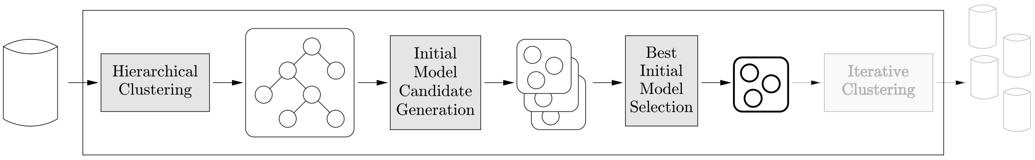

Hierarchical Clustering
Initial Model Candidate Generation
Best Initial Model Selection
Iterative Clustering

Figure 4.6: Hybrid unsupervised seed generation method (Geo.Seeds)

current clustering (line 4) and applying this model to update it (line 5). After this, the second one is shared with the sequential approach, and includes a rule acquisition step to update the decision list (line 6) and the refinement of the clustering using the newly found rules (line 7). The loop is, in this case, iterated until convergence of *both* learners. Finally, as in the sequential case, the patterns to be returned are gathered from the antecedents of the decision list rules (line 8).

To denote the results obtained using this collaborative scheme, we will use the /C suffix following the name of the used seeds.

4.3.3 Clustering

As previously mentioned, clustering is used in our approaches in two different phases: first, the documents are clustered before the co-training loop so as to substitute manual seeding; second, there is a clustering step within the main co-training loop, interleaved with decision-list learning.

Regarding the initial phase, we have considered the use of the Geo and Major[4] methods presented in the previous chapter (Sections 3.4.1.1 and 3.4.2.2, respectively) to obtain a starting clustering of the document collection. We have chosen these methods as they were the ones to obtain the best results, in terms of purity and F1, in our document clustering evaluation (Section 3.5).

Additionally, given that the pattern extraction loop only requires a set of seeds as input, we have considered an alternative Geo.Seeds method: a variation of Geo in which the final iterative clustering step is omitted, and which is depicted in Figure 4.6. Given that the initial model candidates are built from tight and separated groups of objects, we believe these groups can fulfill the requirements to act as a replacement of manual seeds.

With respect to the iterative clustering step, we have followed Surdeanu et al., and used the EM algorithm to fit a mixture of multinomial distributions. This is, again, the same algorithm we had already resorted to for the final iterative clustering step of Geo.

However, in order to improve the interleaving of the decision-list learning and clustering processes, Surdeanu et al. make a recommendation regarding the update of the clustering using the probabilistic model. As in classification EM (Celeux and Govaert, 1992), hard assignment is used at each iteration. However, instead of directly assigning objects to the component with the largest a posteriori probability of having generated them, the authors recommend preserving the cluster assignments from the previous iteration, only updating the clustering of those documents for which this most probable component has generated them with a probability beyond a given threshold (set in their—and our—experiments to 0.95).

4.3.4 Decision List Learning

The rule learning procedure follows the scheme of Yarowsky (1995). Formally, given a pattern dictionary $\Omega^p = (p_1 \dots p_z)$, the algorithm requires the bag-of-patterns representation of the document collection, $\mathcal{X}^p = \{\vec{x}_1 \dots \vec{x}_n\}$—where each x_{iw} in $\vec{x}_i = (x_{i1} \dots x_{iz})$ contains the number of occurrences of pattern p_w in document x_i—and a hard clustering $\Pi = \{\pi_1 \dots \pi_k\}$; and outputs a decision list with simple rules of the type $p_a \rightarrow \pi_c$.

Each learning iteration is then a two-step process in which all *antecedent* patterns p_a are scored with respect to each potential *consequent* cluster π_c, and then the best ranked pattern-cluster pairs are acquired as rules. Among the different scoring functions proposed by Surdeanu et al., we have used those of Riloff (1996) and Collins and Singer (1999). Both criteria start by finding the *precision*

[4] The best-performing settings of $k_{max} = 10$ and $R = 50$, with the Agglo+L combination algorithm, have been used (see Section 3.5.3).

of the patterns over each cluster:

$$\mathrm{prec}(p_a\,;\pi_c) = \frac{\mathrm{freq}(p_a \wedge \pi_c)}{\mathrm{freq}(p_a)} = \frac{|\{x_i \in \pi_c \mid x_{ia} > 0\}|}{|\{x_i \in \mathcal{X} \mid x_{ia} > 0\}|}$$

The patterns for which the precision is above a certain threshold are then scored using a criterion-specific formula, whereas those falling below are assigned a zero score. In the case of Collins and Singer's criterion, the score is the absolute frequency of the pattern in the whole collection:

$$\mathrm{score}_{Co}(p_a\,;\pi_c) = \begin{cases} \mathrm{freq}(p_a) & \text{if } \mathrm{prec}(p_a\,;\pi_c) > T_{Co} \\ 0 & \text{otherwise} \end{cases}$$

whereas for Riloff's the product of precision and log-frequency of the pattern over the cluster is used:

$$\mathrm{score}_{Ri}(p_a\,;\pi_c) = \begin{cases} \mathrm{prec}(p_a\,;\pi_c) \cdot \log \mathrm{freq}(p_a \wedge \pi_c) & \text{if } \mathrm{prec}(p_a\,;\pi_c) > T_{Ri} \\ 0 & \text{otherwise} \end{cases}$$

We have chosen these particular criteria as they were the ones reported to obtain the best results by the authors. The suggested values of $T_{Co} = 0.95$ and $T_{Ri} = 0.5$ for the respective precision thresholds have also been preserved.

The selected rules are appended to build a decision-list classification model (Rivest, 1987). In order to classify a document x' using this model, all rules are checked and the consequent of the highest-precision matching rule is returned as prediction.

In order to allow the bootstrapping of the learner (sequential combination) and the interleaving of the decision-list learning and clustering processes (collaborative combination), Surdeanu et al. recommend selecting the rules for different clusters independently, taking the 3 highest scored rules in each one of them at every acquisition iteration.

We will refer to the results obtained with Collins and Singer's and Riloff's criteria using the :Co and :Ri suffixes, respectively, appended to the seeds' name and combination scheme.

4.4 Text Categorization Evaluation

The first of the two evaluations we have carried out on the proposed approaches aims at comparing the performance of the clustering-seeded pattern learning method with that of the manually seeded one. With this purpose, we have replicated the evaluation presented by Surdeanu et al. (2006). In their work, the authors use an indirect evaluation scheme, applying the obtained decision list classifier for text categorization, and evaluating the performance of their method on this task.

The *text categorization* task "*(a.k.a. text classification, or topic spotting), [is] the activity of labeling natural language texts with thematic categories from a predefined set*" (Sebastiani, 2002). Text categorization has been a typical testbed for ML methods, and several comprehensive comparisons of their performance, as well as surveys on the topic, have been published elsewhere (e.g., Yang, 1999; Sebastiani, 2002).

Given that the core of our pattern acquisition approaches is the learning of a decision list for document clustering, it is quite straightforward to apply the obtained classifier on a new set of documents, and then assess the quality of the obtained patterns by their performance on the categorization of this test data. Nevertheless, it is important to bear in mind that, as mentioned in the chapter introduction, evaluation of IE is based on the one-domain-per-pattern assumption.

Next sections give more details about the way this indirect evaluation has been performed. The data collections we have used are described in Section 4.4.1. Section 4.4.2 describes, in turn, the protocol and metrics used. Finally, Section 4.4.3 discusses the obtained results.

4.4.1 Data

For the text categorization evaluation, we have used a subset of the real-world English datasets to which we had previously resorted for evaluation of clustering approaches (Section 3.5.1). Specifically, we have used the Apw, Lat, Reu and Smt collections. The choice comes from the fact that these are the datasets used by Surdeanu et al. in their evaluation. The preprocessing applied to the documents is the same as in the previous chapter.

4.4.2 Protocol

In order to assess the quality of the text categorizations produced by the compared methods, the standard metrics of micro-averaged precision, recall and F1 have been used. Their usage for text categorization is common (e.g., Lewis, 1991) and, similarly to those considered for clustering, they are based on comparing the system output to a gold standard. However, given that the decision list classifier may not be able to assign a category to all test documents, they take into account the fact that some among them may remain unclassified.

An extra consideration which has to be taken into account comes from the fact that, in difference from manual seeds, there need not be a direct correspondence between the clusters found by the clustering methods and the categories in the gold standard. Because of this, the evaluation requires a cluster-to-category mapping to be found. For our purposes, we have allowed clusters to be mapped to the category with which they have the most documents in common.

Formally, given the partial partition of the test set produced by the decision list classifier $\Pi = \{\pi_1 \dots \pi_k\}$ (where $\bigcup \pi_c \subseteq \mathcal{X}$) and the gold partition $\hat{\Pi} = \{\hat{\pi}_1 \dots \hat{\pi}_{\hat{k}}\}$, the *cluster-to-category mapping* from Π to $\hat{\Pi}$ is thus defined as the function $\varphi : \Pi \to \hat{\Pi}$ such that

$$\varphi(\pi_c) = \arg\max_{\hat{\pi}_d \in \hat{\Pi}} |\pi_c \cap \hat{\pi}_d|$$

The cluster-to-category mapping is not necessarily injective, and hence not necessarily invertible. However, an inverse *category-to-clusters mapping* $\bar{\varphi}$ can also be defined, as the union of the category inverses under φ:

$$\bar{\varphi}(\hat{\pi}_d) = \bigcup_{\varphi(\pi_c)=\hat{\pi}_d} \pi_c$$

Once these two mappings are found, we can calculate micro-averaged precision, recall and F1, which are defined as:

Precision (Prc) measures the degree to which the objects assigned to a cluster belong to the category to which the cluster is mapped. The precision of the cluster is the fraction of its size which the objects in this category represent. The overall micro-averaged precision of a categorization is the average of all cluster precisions, weighted by cluster size.

$$\begin{aligned}
\text{Prc}(\pi_c\,;\hat{\Pi}) &= \frac{|\varphi(\pi_c) \cap \pi_c|}{|\pi_c|} \\
\text{Prc}(\Pi\,;\hat{\Pi}) &= \frac{\sum_{\pi_c \in \Pi} |\pi_c| \cdot \text{Prc}(\pi_c\,;\hat{\Pi})}{\sum_{\pi_c \in \Pi} |\pi_c|}
\end{aligned}$$

Recall (Rec) measures the degree to which the objects from a category are assigned to clusters mapped to that category. The recall of the category is the fraction of its size which the objects in these clusters represent. The overall micro-averaged recall of a categorization is the average of all category recalls, weighted by category size.

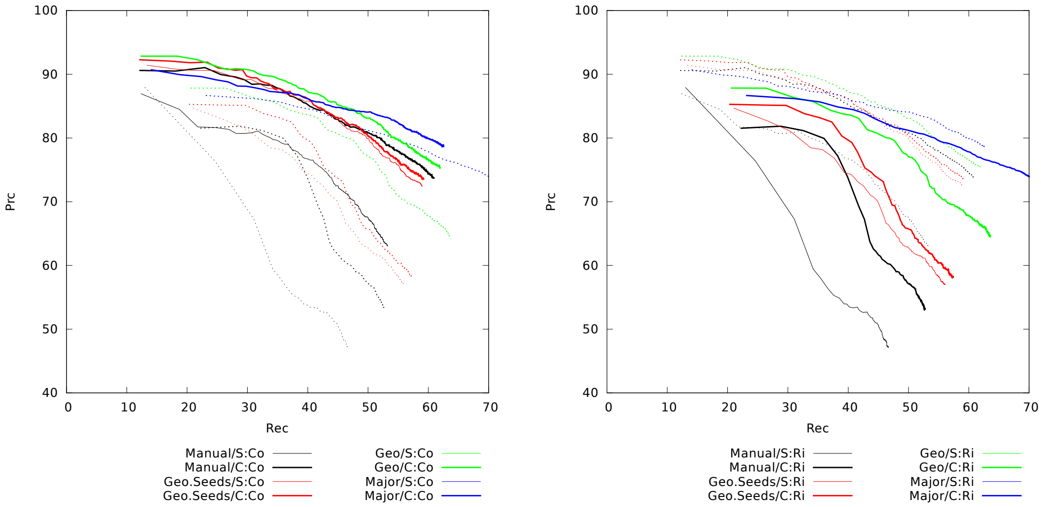
$$\begin{aligned}
\text{Rec}(\Pi\,;\hat{\pi}_d) &= \frac{|\bar{\varphi}(\hat{\pi}_d) \cap \hat{\pi}_d|}{|\hat{\pi}_d|} \\
\text{Rec}(\Pi\,;\hat{\Pi}) &= \frac{\sum_{\hat{\pi}_d \in \hat{\Pi}} |\hat{\pi}_d| \cdot \text{Rec}(\Pi\,;\hat{\pi}_d)}{\sum_{\hat{\pi}_d \in \hat{\Pi}} |\hat{\pi}_d|}
\end{aligned}$$

F1 tries to be a global performance score, and is calculated as the harmonic mean of precision and recall:

$$\text{F}_1(\Pi\,;\hat{\Pi}) = \frac{2 \cdot \text{Prc}(\Pi\,;\hat{\Pi}) \cdot \text{Rec}(\Pi\,;\hat{\Pi})}{\text{Prc}(\Pi\,;\hat{\Pi}) + \text{Rec}(\Pi\,;\hat{\Pi})}$$

As customary, the values for these metrics will always be expressed as percentages.

The evaluation is performed using 5-fold cross-validation. For each fold, the clustering and pattern learning steps are performed on the training partition, and the obtained decision-list classifier is then applied on the test one. Text categorization performance is then measured on the test data. The sole exception is the Reu collection, in which a training and test partition was already defined,



(a) Apw collection, Co criterion

(b) Apw collection, Ri criterion

Figure 4.7: Categorization precision-recall curves per collection (I)

and a single evaluation fold has been carried out. The assignment of documents to training and test sets for each one of the folds, as well as the selection of those to be used as manual seeds, was the same used by Surdeanu et al.[5].

Precision, recall and F1 measures are regarded as functions of the number of co-training iterations. When finding mean performance across folds, it is the values for the same number of iterations which are averaged. Additionally, for the Major clustering method, 5 runs of the method have been considered for each fold, and the averages across all folds and runs are presented.

It is important to note that Surdeanu et al. did not propose any terminating condition for the acquisition procedure. As hypothesis testing requires the determination of a cutoff point, we have opted to report the results for the iteration where the Best value of F1 is achieved. Similarly to the protocol used for clustering evaluation (Section 3.5.2), Bergmann and Hommel hypothesis tests are applied on these results to assess their statistical significance.

Finally, given that Surdeanu et al. used only predicate-argument structures (`p:a`), the experiments in this section are all performed within this pattern family. We postpone to the relation detection evaluation, to be discussed in Section 4.5, a comparison of this and the competing binary feature conjunction formalism.

4.4.3 Results

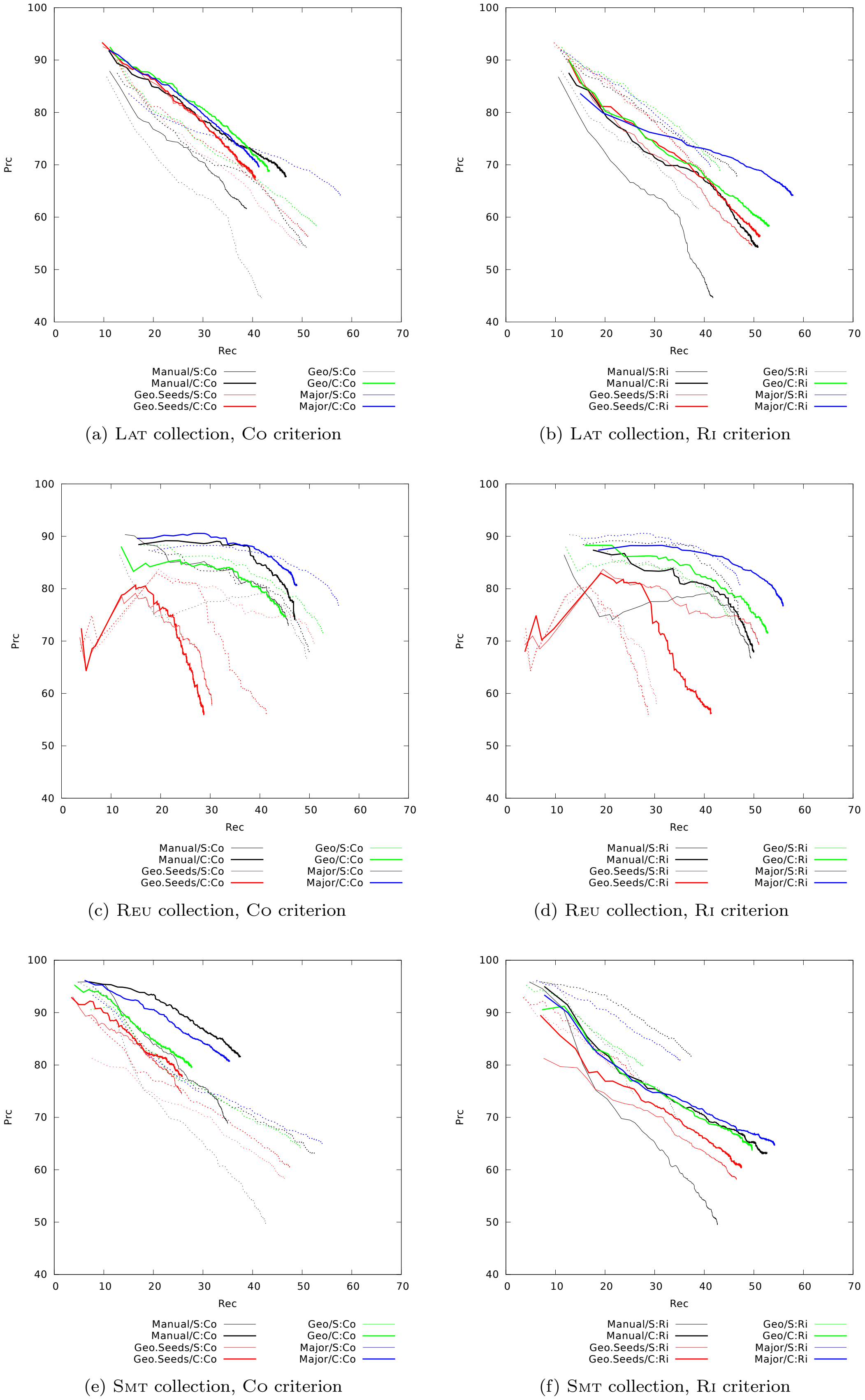
4.4.3.1 Precision-Recall Curves

Figures 4.7 and 4.8 contain the precision-recall curves for the proposed approaches over the four considered document collections. The left column shows the results obtained with the Co selection criterion; and the right one, those with Ri. In all plots, the corresponding curves for the opposite criterion are depicted in dotted lines to ease the comparison.

Overall, the plots put into relief three main trends in the results: first, the improvement produced by the use of a collaborative instead of a sequential combination scheme; second, the difference in precision-recall trade-offs of the two rule selection criteria Co and Ri; and third, the competitive performance of clustering-seeded approaches with respect to the manually seeded one.

Delving more into the detail of the first phenomenon, the plots show how, for Manual and Geo.Seeds, the collaborative process positively improves the results obtained by the sequential one. Only in Reu, the curve of Geo.Seeds/C is, for larger values of recall, below that of Geo.Seeds/S in terms of precision. Regarding Geo and Major, the fact that a complete clustering solution is being fed as input to the co-clustering loop limits the influence of the decision-list learner, and the

[5] We would like to thank Surdeanu, Turmo, and Ageno for providing us these and all the code and data we required to be able to faithfully replicate their results.



(a) LAT collection, CO criterion

(b) LAT collection, RI criterion

(c) REU collection, CO criterion

(d) REU collection, RI criterion

(e) SMT collection, CO criterion

(f) SMT collection, RI criterion

Figure 4.8: Categorization precision-recall curves per collection (II)

partition is dominated by the starting clustering step. As a result, in these cases co-clustering is rendered equivalent to the much simpler sequential combination, and no differences are observable between the results of the sequential and collaborative approaches. Nevertheless, in all cases pattern learning via co-clustering can be said to be more or equally effective than the sequential scheme. This is in complete agreement with the results previously reported, using MANUAL seeds, by Surdeanu et al. (2006).

Regarding the two different rule selection criteria, the general observed tendency is that CO shows a stronger preference for precision than RI, which is more geared towards extraction recall. Hence, for the same seeds, the curves for RI appear below and to the right of those for CO. Only the results on the REU collection using GEO.SEEDS show a different picture—RI there obtains *both* larger precision and recall values, and its curves dominate those of CO. These results are consistent with the differences in parameter setting recommended by the authors of the two criteria. As mentioned in Section 4.3.4, Collins and Singer recommend a precision threshold of $T_{Co} = 0.95$, whereas Riloff sets it to $T_{Ri} = 0.5$. Therefore, in order to obtain a more confident classifier, the former prunes a large number of lower-precision candidate rules—which, on the contrary, will be eventually accepted by the latter and allow it to achieve a larger recall.

Finally, considering the differences between seeding strategies, the results of the clustering-seeded approaches, even within a sequential scheme, are seen in the plots to match those of the manually seeded ones in terms of precision and recall. This is particularly true on the APW and LAT collections using any of the GEO.SEEDS, GEO and MAJOR starting clustering methods; and on all collections using MAJOR and, to a lesser degree, GEO.

Overall, the patterns obtained using clustering-seeded approaches achieve a level of precision beyond 90 or 80%, depending on the selection criterion, on their most confident documents; and one around 70% for values of recall larger than 50%. This is clearly good news: the results certify the validity of the clustering-based seeding strategy as a replacement to manual seeds. With the former, we have been able to reduce the level of supervision of the process, yet obtained a system whose results are comparable, or even better—for the text categorization task—than those of the more supervised one which requires the latter.

4.4.3.2 Hypothesis Tests

Table 4.2 contains the values of precision, recall and F1 obtained at the BEST iteration for each collection, seed and combination strategy. The results for GEO/C:CO and MAJOR/C:CO are shaded as a visual reminder that they are equal to those of GEO/S:CO and MAJOR/S:CO, respectively[6].

The figures here do nothing but confirm the conclusions drawn from the observation of the precision-recall curves. In particular, it is interesting to note how, effectively, for the same collections and seed sets, the approaches using the RI criterion tend to solutions with higher recall—and, in fact, with a better balance of the two measures. Also, it can be seen how collaborative combination improves the precision and, more often than not, the recall of their sequential equivalents. Overall, the F1 score achieved by the former is always higher than that of the latter, except for the case of the REU collection when using GEO.SEEDS. Therein, the drop in recall and, especially, precision between GEO.SEEDS/C:RI and GEO.SEEDS/S:RI is remarkable. This can also be told from the difference of the right ends of the corresponding curves in Figure 4.8d.

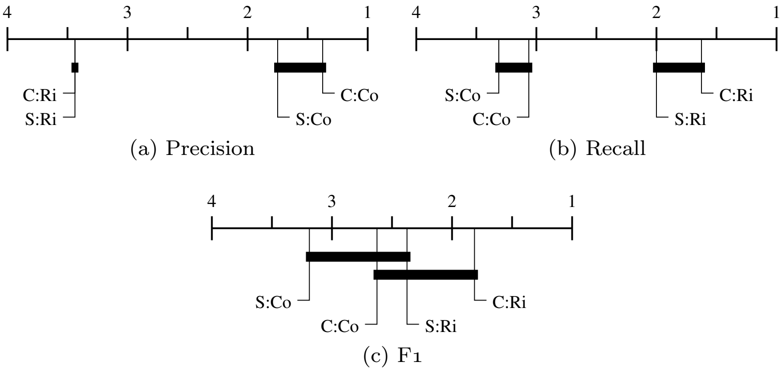
The statistical significance of all these comparisons can be confirmed by the corresponding Bregman-Hommel tests. The first group of them tries to assess the differences in performance of combination strategies and selection criteria over all collections and seed sets, and its outcomes are collected in Figure 4.9. Collaborative approaches obtain better results in all metrics than their sequential homologues (except for precision when using the RI criterion), but the differences are not statistically significant. However, the differences between CO- and RI-based approaches are significant, favouring the former in the case of precision, and the latter in the case of recall. Overall, it is the collaborative C:RI which achieves the best F1 scores (Figure 4.9c), followed by its counterpart S:RI and then the CO-based approaches. The difference between C:RI and S:CO is statistically significant. As the two strategies using the RI criterion obtain better average ranks than those with CO, we will henceforth focus our discussion on the results obtained using this first criterion. However, it can be checked how the results are, in broad outlines, similar enough in both cases.

[6] For parsimony, equivalent results obtained using a simpler approach should be preferred.

		S:Co			C:Co		
		Best			Best		
		Prc	Rec	F1	Prc	Rec	F1
Apw	Manual	63.49	53.01	**57.78**	73.74	60.94	**66.72**
	Geo.Seeds	72.60	58.92	**65.04**	73.63	59.16	**65.60**
	Geo	75.57	61.87	**68.03**	75.57	61.87	**68.03**
	Major	78.74	62.52	**69.69**	78.74	62.52	**69.69**
Lat	Manual	61.63	38.82	**47.63**	68.35	46.47	**55.33**
	Geo.Seeds	67.09	40.63	**50.57**	67.62	40.57	**50.67**
	Geo	68.86	43.39	**53.21**	68.86	43.39	**53.21**
	Major	69.76	41.23	**51.81**	69.76	41.23	**51.81**
Reu	Manual	74.15	45.54	**56.43**	77.44	46.80	**58.34**
	Geo.Seeds	59.72	30.18	**40.09**	57.53	28.53	**38.14**
	Geo	74.76	45.15	**56.30**	74.76	45.15	**56.30**
	Major	80.69	47.42	**59.73**	80.69	47.42	**59.73**
Smt	Manual	69.33	34.86	**46.39**	81.64	37.53	**51.40**
	Geo.Seeds	74.72	25.68	**37.98**	77.97	25.77	**38.49**
	Geo	79.63	27.75	**40.72**	79.63	27.75	**40.72**
	Major	80.75	35.33	**49.10**	80.75	35.33	**49.10**

		S:Ri			C:Ri		
		Best			Best		
		Prc	Rec	F1	Prc	Rec	F1
Apw	Manual	50.95	44.81	**47.68**	56.30	51.16	**53.61**
	Geo.Seeds	59.77	53.89	**56.67**	59.31	56.61	**57.93**
	Geo	66.01	62.67	**64.30**	66.01	62.67	**64.30**
	Major	72.82	71.57	**72.19**	72.82	71.57	**72.19**
Lat	Manual	59.87	35.08	**44.23**	56.56	49.23	**52.64**
	Geo.Seeds	55.58	49.00	**52.08**	56.40	51.18	**53.66**
	Geo	58.39	53.00	**55.56**	58.39	53.00	**55.56**
	Major	64.22	57.85	**60.87**	64.22	57.85	**60.87**
Reu	Manual	69.28	48.92	**57.35**	73.16	47.98	**57.95**
	Geo.Seeds	71.57	50.26	**59.05**	57.04	41.22	**47.86**
	Geo	72.81	52.30	**60.87**	72.81	52.30	**60.87**
	Major	78.03	55.50	**64.86**	78.03	55.50	**64.86**
Smt	Manual	52.34	41.43	**46.25**	63.20	52.61	**57.42**
	Geo.Seeds	58.40	46.46	**51.73**	60.67	47.43	**53.21**
	Geo	64.59	49.55	**56.02**	64.59	49.55	**56.02**
	Major	65.29	53.86	**59.03**	65.29	53.86	**59.03**

Table 4.2: Comparison of pattern acquisition strategies, evaluated on text categorization

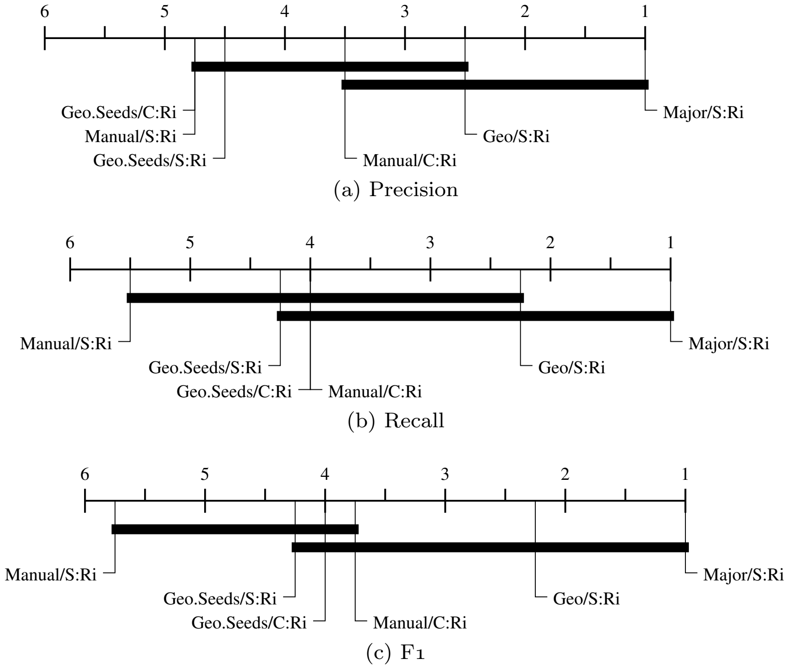


(a) Precision

(b) Recall

(c) F1

Figure 4.9: Bergmann-Hommel tests for text categorization (all criteria)



(a) Precision

(b) Recall

(c) F1

Figure 4.10: Bergmann-Hommel tests for text categorization (Ri criterion)

The second group of Bergmann-Hommel tests, represented in Figure 4.10, were performed to establish the significance of the observed differences within the approaches using the Ri criterion. It is interesting to note how the results obtained using Major/S and Geo/S seeds achieve the first and second, respectively, best average ranks in all three metrics, always exceeding the results of Manual/C. Actually, Major/S has an average rank of 1—i.e., it obtains the best results in *all* cases. This again confirms the validity of the clustering-seeded strategy, and how it is not only competitive but even able to outperform the manually seeded one. Regarding the sequential and collaborative combinations using Geo.Seeds, they fall between Manual/C and Manual/S in terms of performance.

Nevertheless, similarly to the case of our clustering evaluation (Section 3.5.3), the reduced number of available data collections prevents the Bergmann-Hommel tests from detecting but a small number of statistically significant differences. Only that in the precision between using seeds Major/S and all other sources except Manual/C and Geo/S, and that in the recall and F1 between Major/S and Manual/S are judged so.

		2003		2004	
		Docs.	Words	Docs.	Words
<bn>	Broadcast news	147	38,298	220	69,547
<nw>	Newswire	105	67,100	223	101,109
<ts>	Telephone speech	-	-	8	14,937
<all>	TOTAL	252	105,398	451	185,593

		2005		TOTAL	
		Docs.	Words	Docs.	Words
<bc>	Broadcast conversation	60	46,587	60	46,587
<bn>	Broadcast news	226	62,820	593	170,665
<nw>	Newswire	106	54,766	434	222,975
<ts>	Telephone speech	39	48,901	47	63,838
<un>	Usenet groups	49	42,084	49	42,084
<wl>	Weblogs	119	42,316	119	42,316
<all>	TOTAL	599	297,474	1,302	588,465

Table 4.3: ACE subcollection sizes

4.4.3.3 Conclusions

Overall, the results of the indirect evaluation look promising, as clustering-seeded approaches match or outperform manual ones. In particular, the acquisition process using the Ri rule selection criterion and the seeds provided by the Major and Geo clustering methods allows the obtainment of patterns which, when applied on a text categorization task, exceed the results of all other compared approaches in terms of precision, recall and F1.

However, the fact that this is an *indirect* evaluation, and that the approaches—and the framework itself—are hence still in need of *direct* validation on a real extraction task, must be remembered. The evaluation in next section is specifically devised to fill in this gap.

4.5 Relation Detection Evaluation

Once the feasibility of replacing manual seeding by a clustering process has been ascertained, it becomes necessary to carry out a direct evaluation of the considered approaches on an actual IE task—in particular, the relation detection task.

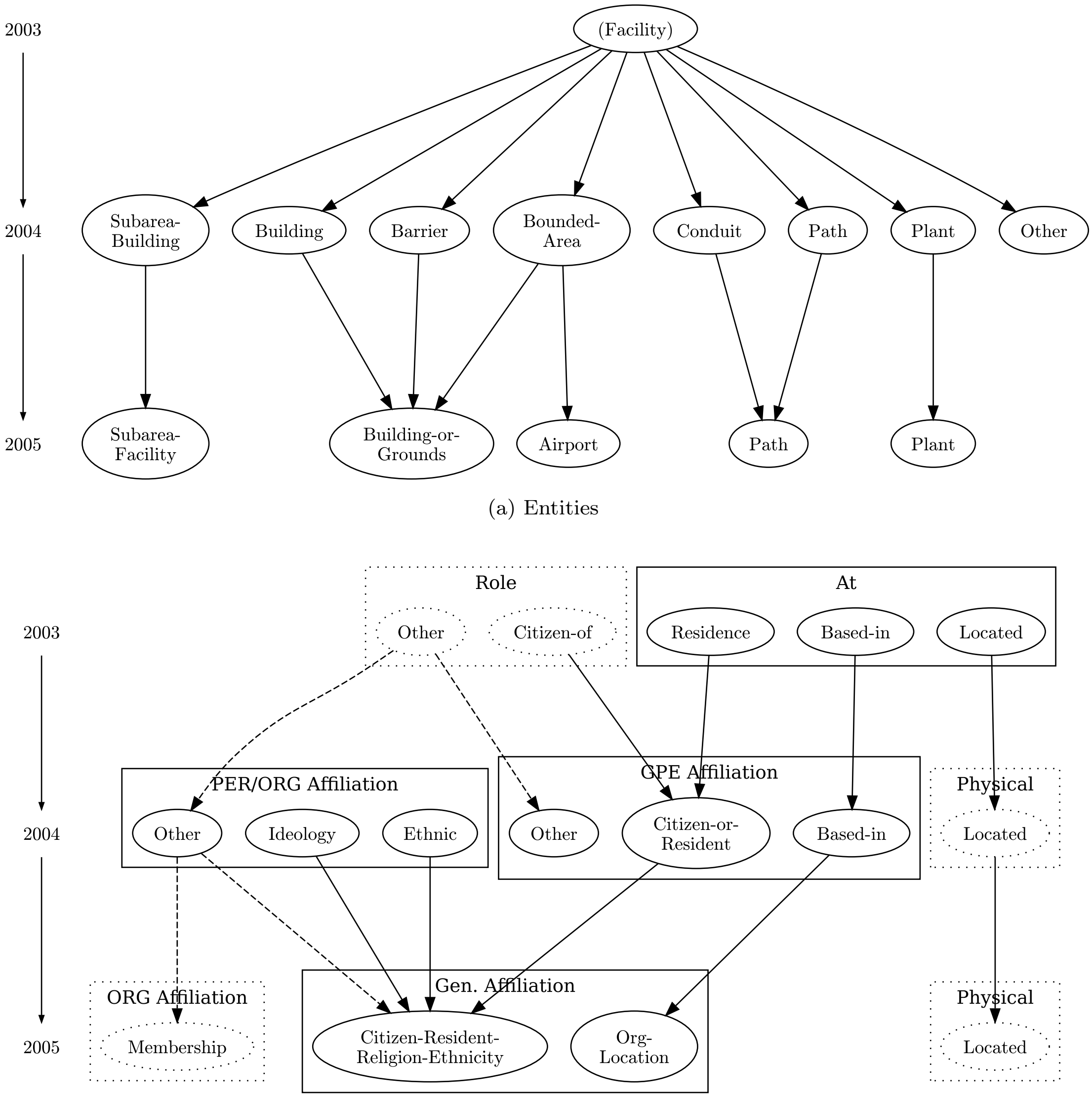
The details of the evaluation are given in the following sections. Section 4.5.1 describes the employed data collection, and Section 4.5.2, the protocol and metrics. Section 4.5.3 discusses the obtained results.

4.5.1 Data

In order to evaluate the proposed approaches on a relation detection task, we have used the English training data provided by the organizers of the ACE-2003, ACE-2004 and ACE-2005 evaluations. The collection consists of 1,302 documents coming from a variety of heterogeneous sources (broadcast news and conversations, newswire text, telephone speech, and Usenet and weblog posts) and contains a total of around 588k words. Table 4.3 contains a more detailed description of the number of documents and words (after preprocessing with our tools) in each one of the considered ACE subcollections[7].

The data contain annotations regarding the entities, relations, events and mentions of them present in the documents. Given that we are focusing on the relation detection task, we have

[7] The ACE-2005 dataset was used for the ACE-2005, ACE-2007 and ACE-2008 competitions, making the considered collection embrace 6 years of IE evaluation.



(a) Entities

(b) Relations

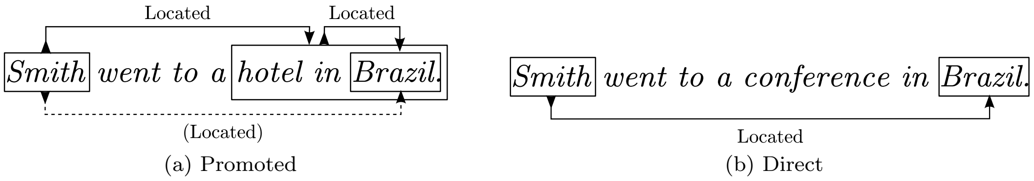
Figure 4.11: Entity and relation types in the ACE 2003–2005 evaluations (fragment)

disregarded event annotations. Moreover, in order to isolate our evaluation from issues related to entity detection, we have used the annotated gold entity mentions for our experiments. In total, the corpus contains 100,932 entity mentions and 19,160 relation mentions between them.

The type and subtypes of the entities and relations annotated—and hence to be extracted—in the different ACE evaluations kept changing from one installment of the series to the next one (see ACE, 2003, 2004, 2005a). Figure 4.11 contains diagrams of two fragments of the entity type and subtype hierarchies, as they evolved through ACE-2003, ACE-2004 and ACE-2005[8]. An arrow between two subtypes expresses the transformation of one into the other during the change of annotation scheme[9]. Both hierarchies are complex and, especially in the case of relations, were the object of significant restructuring from year to year. Nevertheless, it is interesting to note that, despite possible changes in type or subtype, entities and relations which were or were not annotated in one evaluation usually remained so the following year. The only significant change was the introduction of the *Vehicle* and *Weapon* types in the 2004 evaluation, together with relation types in which entities of these types might be involved. As in the detection task the concrete

[8] Appendix C contain the diagrams for the complete hierarchies.

[9] The correspondences are established by ourselves after comparison of the annotation guidelines, and do not necessarily correspond to the official view of the ACE organizers.


Located
Located
Smith went to a hotel in Brazil.
(Located)
Smith went to a conference in Brazil.
Located

(a) Promoted (b) Direct

Figure 4.12: Promoted and direct relations in the ACE corpus (adapted from ACE, 2005b)

relation type and subtype are irrelevant, we believe we can use the three datasets as a single, and homogeneous enough, collection.

It is also interesting to observe how, a number of the relation types in ACE can be considered *generic*—i.e., they express knowledge which may be of interest across different and unrelated domains: such as location, part-of, employment or social relations, just to name a few.

Impact of unannotated relations The main problem associated to the use of the ACE data for evaluation is that, by following the ACE annotation guidelines, we will be considering as non-related those entity pairs which are linked by some relation not included in them. Given the unsupervised nature of the approaches considered in this thesis, it is likely that instances of relations beyond the ACE types be discovered—and by using the ACE data as gold, we will judge them as false positives.

In addition, following the guidelines, those relations which are *promoted* through taggable entities should also be omitted from the annotation: "*if [...] one of the Entity Mentions to be used as an argument is embedded in some other (Simple) Entity Mention, then that Entity Mention is not accessible and the potential Relation is not taggable*" (ACE, 2005b). Similarly to the previous case, this will cause these relations to be judged as false positives if detected by the systems.

As an example (also adapted from the guidelines), there does exist a relation between "*Smith*" and "*Brazil*" in the sentence shown in Figure 4.12a. However, "*a hotel in Brazil*" is itself annotated as an entity, and there exist relations between "*Smith*" and "*hotel*", and between "*hotel*" and "*Brazil*"—and, as a result, the previous relation is considered promoted and hence not annotated as such. The interaction of this criterion with those of entity annotation causes that for the similar sentence shown in Figure 4.12b, on the contrary, the relation between "*Smith*" and "*Brazil*" be annotated. In this case, "*conference*" does not belong to any taggable entity type, and the relation is direct—and hence annotated.

Despite these drawbacks, the corpus provides us a convenient gold standard, created following a consistent set of criteria. Moreover, assuming that all methods will have a similar tendency to detect these unannotated relations, the issue will have a uniform impact across them, and will not affect the assessment of their relative performance—even if it will alter the absolute performance measurements. For these reasons, we have decided to stick to the ACE data *as is* in order to compare the performance of the several methods. However, the issue of unannotated relations must be kept in mind and, after presenting the considered performance metrics in next Section 4.5.2, we will briefly discuss the impact it may have in them.

4.5.2 Protocol

So as to numerically assess the system performance on relation detection, we have used once more corresponding versions of the standard metrics of precision, recall and F1. This choice is common in binary classification problems in which the class distribution is considerably unbalanced, such as information retrieval (Raghavan et al., 1989) or gene identification (Bockhorst and Craven, 2004). They have also been previously used in evaluations of IE tasks and, in particular, relation detection (Zelenko et al., 2003; Bunescu and Mooney, 2005).

For the purpose of formalization, it will be convenient to regard relation detection as the determination of a binary partition Π of the set of contexts of co-occurring entity mention pairs, between related and unrelated ones. We will also refer to these two sets as the *positive* and the *negative* sets, and use the notation π_+ and π_- for them, respectively. For evaluation, we will require the *gold partition* $\hat{\Pi}$, which will contain the set of *real positive* $\hat{\pi}_+$ and *real negative* $\hat{\pi}_-$ contexts.

We can then define:

Precision (Prc) measures the degree to which the contexts classified positive are really positive, and is defined as the ratio between *true positives* and the sum of *true positives* and *false positives*:

$$\mathrm{Prc}(\Pi\,;\hat{\Pi}) = \frac{|\pi_+ \cap \hat{\pi}_+|}{|\pi_+ \cap \hat{\pi}_+| + |\pi_+ \cap \hat{\pi}_-|} = \frac{|\pi_+ \cap \hat{\pi}_+|}{|\pi_+|}$$

Recall (Rec) measures the degree to which the positive contexts are classified so, and is defined as the ratio between *true positives* and the sum of *true positives* and *false negatives*:

$$\mathrm{Rec}(\Pi\,;\hat{\Pi}) = \frac{|\pi_+ \cap \hat{\pi}_+|}{|\pi_+ \cap \hat{\pi}_+| + |\pi_- \cap \hat{\pi}_+|} = \frac{|\pi_+ \cap \hat{\pi}_+|}{|\hat{\pi}_+|}$$

F1 tries again to be a global performance score, and is calculated as the harmonic mean of precision and recall:

$$\mathrm{F}_1(\Pi\,;\hat{\Pi}) = \frac{2 \cdot \mathrm{Prc}(\Pi\,;\hat{\Pi}) \cdot \mathrm{Rec}(\Pi\,;\hat{\Pi})}{\mathrm{Prc}(\Pi\,;\hat{\Pi}) + \mathrm{Rec}(\Pi\,;\hat{\Pi})}$$

As usual, percentages will be used to express the values of these metrics.

Additionally, to evaluate the performance of the pattern learning, isolating it from any suitable stopping criterion, we have also included information about Receiver Operator Characteristic (ROC) curves, more specifically, the Area Under the ROC Curve (AUC, Fawcett, 2006). The relation of dominance between ROC curves has been proved equivalent to that of precision/recall curves (Davis and Goadrich, 2006), and they are less sensitive to variances of the class skew. After having been employed for decades in other fields like signal detection, psychology and medicine, Spackman (1989) first introduced the use of ROC for evaluation of ML systems.

If we have a certain parameter $t \in T$ which determines a decision threshold or a stopping criterion (e.g., the number of learning iterations in our methods), we can regard the system partition Π as a function of this parameter t. The positive and negative sets will hence be functions $\pi_+(t)$ and $\pi_-(t)$. We can define the *true positive rate* $\mathrm{tpr}(t)$ and *false positive rate* $\mathrm{fpr}(t)$ as:

$$\mathrm{tpr}(t) = \frac{|\pi_+(t) \cap \hat{\pi}_+|}{|\hat{\pi}_+|} \qquad\qquad \mathrm{fpr}(t) = \frac{|\pi_+(t) \cap \hat{\pi}_-|}{|\hat{\pi}_-|}$$

It can be noted how the true positive rate is equivalent to recall for a fixed value of the parameter t. The ROC curve is then defined as the set of points $\{(\mathrm{tpr}(t), \mathrm{fpr}(t)) \mid t \in T\}$. The area under this curve will be:

$$\mathrm{AUC} = \int_T \frac{\mathrm{tpr}(t+dt) + \mathrm{tpr}(t)}{2} \left(\mathrm{fpr}(t+dt) - \mathrm{fpr}(t)\right) \mathrm{d}t$$

Despite its apparent complexity, for classifiers in which the threshold value for a prediction can be obtained easily, the ROC curve and the area under it can be estimated efficiently (Fawcett, 2006). In our case, it is easy to provide the iteration number in which the most confident pattern matching a context was learnt, so AUC values are simple to obtain.

One must keep in mind that, because of the unannotated relations present in the gold standard (Section 4.5.1), the values for these metrics are biased. Thus, whereas the measured values of recall will be close to the actual ones (one can expect a similar fraction of true positives and false negatives to be considered false positives and true negatives, respectively), the obtained precision, F1 and AUC values will be pessimistic estimations, and would be higher if the gold standard were extended to include those relations beyond the ACE hierarchy (a number of instances will be considered false positives instead of true ones). However, as mentioned previously, this will not affect the comparisons of relative performance between different approaches—only their absolute values.

Last, in order to quantify the degree to which document clustering can be used to acquire knowledge about relations—with independence of any particular employed method—we have decided to use normalized mutual information as defined by Strehl and Ghosh (2002, §2.2). For each related entity pair in $\hat{\pi}_+$, we have considered a random variable T_+ which contains the annotated subtype in the ACE hierarchy, and a random variable Π_+ which contains the cluster to which the containing document belongs[10]. The normalized mutual information $\bar{\mathrm{I}}(T_+\,;\Pi_+)$ between T_+ and Π_+ is then defined as the quotient of the mutual information of the variables and the square root of the product of their entropies[11]:

$$\bar{\mathrm{I}}(T_+\,;\Pi_+) = \frac{\mathrm{I}(T_+\,;\Pi_+)}{\sqrt{\mathrm{H}(T_+)\cdot\mathrm{H}(\Pi_+)}}$$

Strehl and Ghosh state that, for all pairs of random variables X and Y, $0 \leq \bar{\mathrm{I}}(X,Y) \leq 1$, with $\bar{\mathrm{I}}(X,Y) = 1$ if and only if $X = Y$. Given that its range is thus constrained, values of normalized mutual information can be compared across different random variable pairs. In particular, they can be compared across different sets of seeds.

Strictly regarding the experimentation protocol, we will report the results obtained for each of the four considered pattern families (`p:a`, `w:t`, `c:l` and `w:t+c:l`), and each of the seeding strategies (Geo.Seeds, Geo and Major). For Geo.Seeds, the sequential and collaborative approaches are compared—for Geo and Major, only the former is included, following the observation in the text classification evaluation that there is no difference in behaviour between the two. Only the results obtained with the Ri selection criterion of Riloff are reported, given that it clearly outperformed Co with respect to F1 values in the previous evaluation.

Additionally to the sequential and collaborative pattern learning approaches, a baseline Frequent method has been included in the evaluation. It selects the patterns in decreasing order of absolute frequency in the dataset. Moreover, in order to obtain an upper-bound of the system performance, we have considered an Optimal pattern set, incrementally built by selecting, at every step, the pattern which provides the largest F1 value[12].

Regarding the particular case of predicate-argument structures (`p:a`), given that the task is the extraction of binary relations between two entities, only those patterns which contain at least two slots filled with an entity have been generated (extra slots may be filled with either entities or non-entities).

Finally, and similarly to the previous section, we have followed a 5-fold cross-validation scheme for the evaluation, with 5 runs for the Major seeds. Again, clustering and pattern learning are performed on the training partition; relation detection is evaluated on the test partition; and precision, recall and F1 values for a given feature set, method, and number of iterations are averaged.

4.5.3 Results

Figure 4.13 contains the precision-recall curves for the four different pattern feature sets, using the different learning approaches. The plots use a logarithmic scale in both the precision and recall axes.

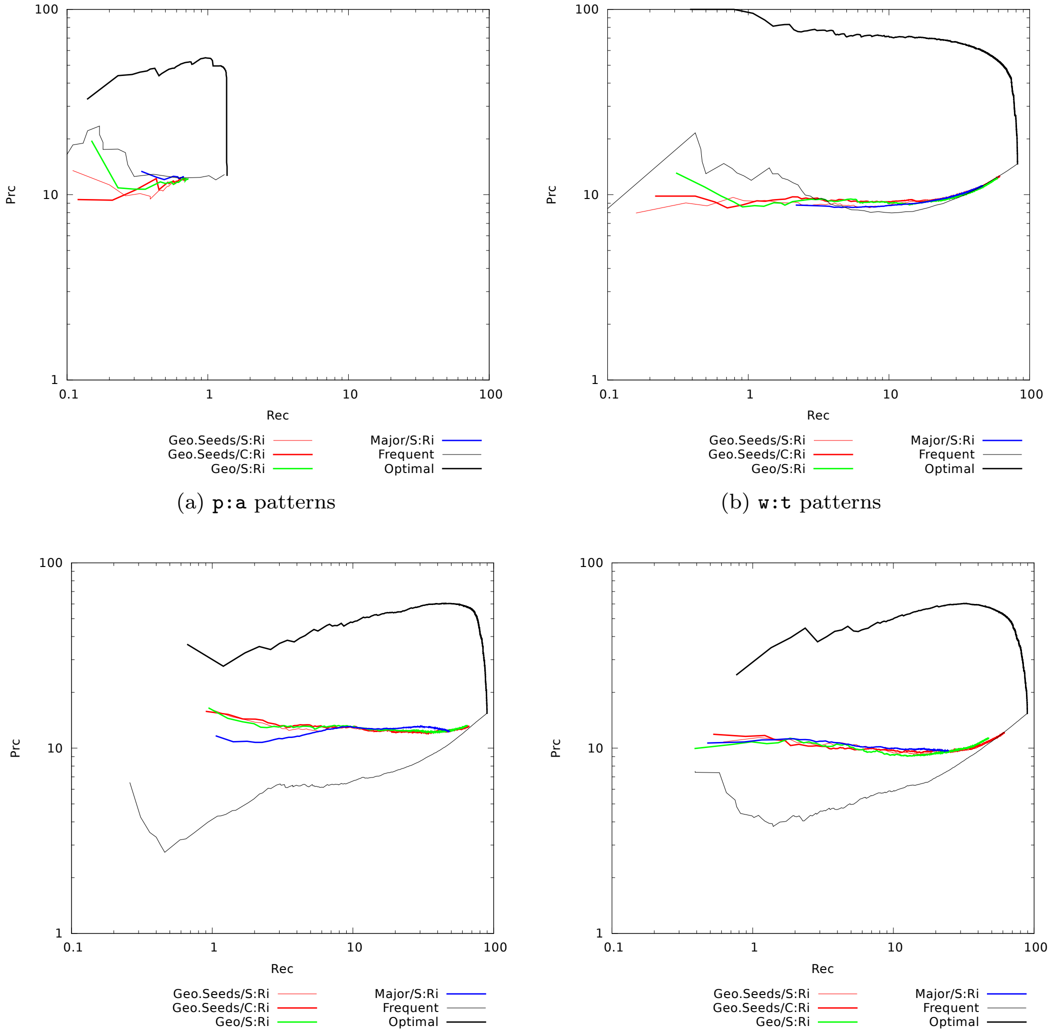
The first obvious impression is that the landscape that these plots draw is radically different from that sketched at the light of the results of the text categorization evaluation. Overall, one could say that the performance is dramatically poor. A number of major problems can be identified:

- The precision of all the obtained approaches is low: the curves do not surpass the 20% threshold, even at their lowest-recall ends. For the case of feature sets `p:a` and `w:t`, the performance of clustering-based approaches falls even below that of the simple Frequent baseline.
- The usual precision-recall trade-off is lost, as the curves exhibit a flat slope, rather than the expected decreasing one.
- In the case of `p:a`, the recall is also extremely low. This formalism is only capable of capturing 1.37% of the relations present in the corpus—a figure much lower than even the 21.2% fraction of mentions with a lexical condition annotated as *verbal*. The presence of linguistic phenomena like coordination or ellipsis, or the location of the verb in a position other than the sentence or clause root, cause the potential relation mentions therein expressed to be missed, and harshly cut down the recall of the detection process.

[10] In the particular case of Geo.Seeds, we have only considered related pairs inside the seed documents, so Π_+ will always be defined.

[11] For the definitions of entropy and mutual information, see Definitions A.4 and A.5 in Appendix A.3.

[12] The name Optimal is a misnomer, as the obtained pattern set may not *strictly* the global optimum. However, the outlined greedy procedure will converge to a local optimum, which we believe will be useful enough as an upper bound of the system performance.



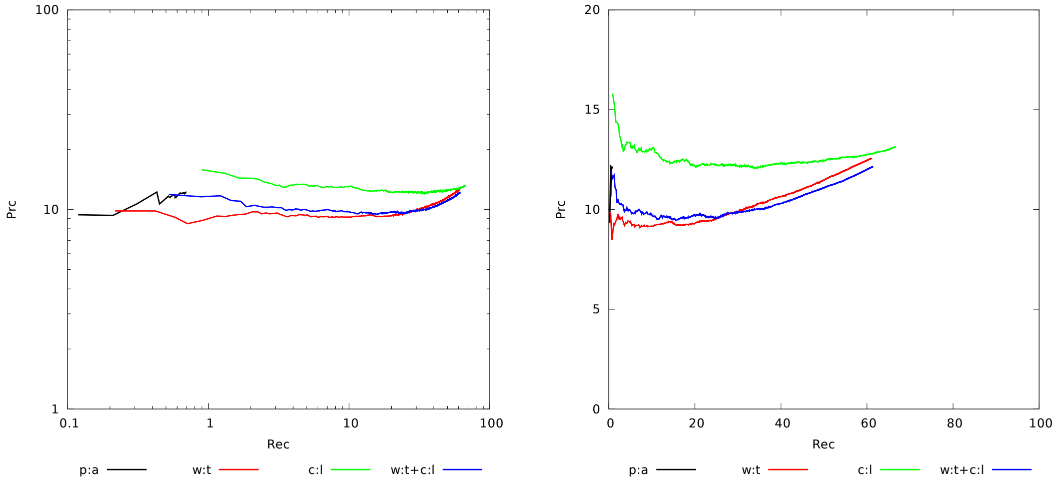
(a) `p:a` patterns

(b) `w:t` patterns

(c) `c:l` patterns

(d) `w:t+c:l` patterns

Figure 4.13: Relation detection precision-recall curves (logarithmic plots)



(a) Logarithmic plot

(b) Linear plot

Figure 4.14: Relation detection precision-recall curves (Geo.Seeds/C:Ri method)

		S:Ri				C:Ri			
			Best				Best		
		AUC	Prc	Rec	F1	AUC	Prc	Rec	F1
`p:a`	Geo.Seeds	**0.501**	11.92	0.62	**1.17**	**0.501**	12.15	0.69	**1.30**
	Geo	**0.501**	12.15	0.72	**1.35**	**0.501**	12.15	0.72	**1.35**
	Major	**0.501**	12.47	0.66	**1.26**	**0.501**	12.47	0.66	**1.26**
	Frequent	**0.503**	12.82	1.31	**2.38**	-	-	-	-
	Optimal	**0.503**	39.81	1.36	**2.64**	-	-	-	-
`w:t`	Geo.Seeds	**0.596**	12.41	59.45	**20.54**	**0.600**	12.56	61.01	**20.84**
	Geo	**0.592**	12.28	58.65	**20.31**	**0.592**	12.28	58.65	**20.31**
	Major	**0.563**	11.18	46.20	**18.00**	**0.563**	11.18	46.20	**18.00**
	Frequent	**0.655**	14.66	81.82	**24.87**	-	-	-	-
	Optimal	**0.840**	50.02	64.40	**56.31**	-	-	-	-
`c:l`	Geo.Seeds	**0.633**	13.01	65.36	**21.69**	**0.635**	13.13	66.54	**21.93**
	Geo	**0.633**	13.24	64.66	**21.98**	**0.633**	13.24	64.66	**21.98**
	Major	**0.596**	12.60	44.83	**19.67**	**0.596**	12.60	44.83	**19.67**
	Frequent	**0.657**	15.42	89.72	**26.32**	-	-	-	-
	Optimal	**0.888**	55.49	71.19	**62.37**	-	-	-	-
`w:t+c:l`	Geo.Seeds	**0.593**	12.06	61.07	**20.14**	**0.596**	12.14	61.29	**20.26**
	Geo	**0.568**	11.32	47.43	**18.22**	**0.568**	11.32	47.43	**18.22**
	Major	**0.525**	9.70	24.49	**13.85**	**0.525**	9.70	24.49	**13.85**
	Frequent	**0.653**	15.39	89.43	**26.26**	-	-	-	-
	Optimal	**0.877**	50.25	65.20	**56.76**	-	-	-	-

Table 4.4: Comparison of pattern acquisition strategies, evaluated on relation detection

Figure 4.14 contains, with both logarithmic and linear axis scales, the precision-recalls plots for all feature sets using the Geo.Seeeds/C:Ri method, which is the one to obtain the best peak F1 values. Even if the results remain low, it is interesting to note how that, for any given level of recall, the best results in terms of precision (and, hence, the overall F1) are obtained with the `c:l` feature set: patterns using `c:l+w:t` features obtain a lower performance, comparable to that of `w:t` alone. Again this is a consequence of the ugly duckling theorem of Watanabe (1985): in unsupervised learning settings, the addition of new features does not always come with an improvement of the recognition performance (see Section 2.4). Overall, and despite the bad results, it is clear that there *is* an improvement on the usage of feature-conjunction patterns with respect to predicate-argument structures, and we will hence favour the former in following experiments.

Finally, Table 4.4 contains the exact precision, recall and F1 values at the Best iteration, as well as the AUC values, for all feature sets and methods. The figures confirm the observations from Figures 4.13 and 4.14: how precision at the Best iteration is, for both sequential and collaborative approaches and features, around the 12%–15% level; or how recall using feature set `p:a` is extremely low.

The achieved AUC values are also poor. In the case of `p:a`, due to its low recall, the values barely exceed the lower bound random classification value of 0.500. Regarding the feature-conjunction patterns, they climb a few more points, but remain always surpassed by the baseline Frequent pattern selection strategy—which also outperforms them in terms of F1. Only the results of Geo.Seeds/C:Ri using `c:l` patterns are comparable, albeit still lower, to this Frequent baseline. In all cases, the upper bound performance of Optimal is completely out of reach.

The results of this evaluation can be considered devastating, and seem to clearly contradict those of our previous text categorization evaluation. Nevertheless, it must be kept in mind that indirect evaluations are not perfect substitutes of direct ones, and their results may hence disagree—a reason why the latter should be preferred if they are possible.

Moreover, one of the keys towards understanding the observed behaviour may be found in

	$\mathrm{H}(\Pi_+)$	$\mathrm{H}(T_+)$	$\mathrm{I}(\Pi_+\,;T_+)$	$\bar{\mathrm{I}}(\Pi_+\,;T_+)$
Geo.Seeds	2.078	3.335	0.591	0.224
Geo	2.192	3.359	0.390	0.144
Major	3.144	3.359	0.671	0.206

Table 4.5: Average relation-cluster entropies and mutual information for the different seeds

Table 4.5, which contains the normalized mutual information between the relation types and the clustering—which is used as seeds for the pattern learning process—obtained using each one of the three methods.

As seen therein, the mutual information between document clusters and relation types is particularly low: the normalized values are found in the 0.14–0.22 range. This means that correlation between the category (cluster) and relation distributions is small, and that the type of relations that are defined in the ACE evaluations—and which we are trying to detect—are likely to be mostly transverse to text categories. Relation mentions for a single relation type are possibly spread across documents from multiple clusters—and, in turn, the same spreading may also be happening for the linguistic constructions which express this relations. Overall, everything points towards the fact that the one-domain-per-pattern assumption is certainly broken—and that, hence, the proposed combination of document clustering and pattern learning is not suitable for the task at hand.

This lack of correlation between relation types and clusters also accounts for the divergence between the text categorization evaluation and this one. The obtained patterns may be suitable for detecting hypothetical relations specific of the domains found by clustering algorithms, which is what the text categorization evaluation measured—but may be useless when trying to identify generic and non-domain-specific ones, which is what the relation detection assesses.

4.6 Conclusions

In this chapter, we have explored sequential and collaborative schemes for the combination of document clustering and IE pattern learning, with the ultimate goal of building unsupervised pattern learning approaches.

We have in particular explored bootstrapping-based schemes, which had been successfully used for weakly supervised systems, and replaced the manual supervision present in them by a document clustering process. The results over an indirect text categorization evaluation show the effectiveness of this replacement, as clustering-seeded approaches match or surpass manually seeded ones on the task.

However, and to our disappointment, when evaluated on an actual relation detection task, the proposed approaches obtain more than modest results, even being outperformed by a frequency-based baseline approach. Nevertheless, the low correlation between the sought relation types and the domains detected by text clustering algorithms points towards the violation of the one-domain-per-pattern assumption as the culprit of these results, and suggests that combination of document clustering and pattern learning may not be the most suitable framework for detection of generic and transverse relations.

At the same time, we have explored a pattern formalism based on conjunctions of binary features, as an alternative to the previously used predicate-argument structures. We have also proposed the application of frequent-itemset mining algorithms to cope with the combinatorial explosion they introduce, and render its use computationally feasible. Despite the overall bad performance on the task, we believe that these patterns represent an improvement over predicate-argument structures and that, hence, their use should be preferred.

At the light of these results, we believe that, among the three combination schemes sketched in Section 1.4, the sequential and collaborative ones may not be suitable to solve our problem– and that, hence, exploration of the remaining one becomes necessary. A joint combination may be useful in decoupling pattern learning from document clustering, and in building unsupervised systems which extract useful generic relations. Nevertheless, the clustering setting which occurs in this joint learning process differs from classical clustering settings, such as the document clustering ones we have been facing so far. It is for this reason that we will wholly devote next Chapter 5 to this alternative—and often disregarded—clustering task, before proceeding to develop our joint clustering and pattern learning approach in Chapter 6.

5 Minority Clustering

"*In almost all textbooks, even the best, this principle is presented so that it is impossible to understand.*" (K. Jacobi, Lectures on Dynamics, 1842-1843). I have not chosen to break with tradition.

Vladimir Igorevich Arnold
Mathematical Methods of Classical Mechanics

This chapter presents our experiments on the task of minority clustering. Inspired by the success of ensemble methods in clustering problems (confirmed by our experiments in previous Chapter 3) we have developed a novel ensemble approach to the problem, Ewocs, *and evaluated its different components to assess its validity.*

Section 5.1 introduces the task of minority clustering, putting into relief its differences with respect to the usual clustering task and to other similar problems. Section 5.2 gives an overview of related work in the area. Next, Section 5.3 contains a description of Ewocs*— including the derivation of a minority clustering algorithm whose properties are theoretically proved under a set of conditions. The obtained algorithm has a number of components which allow different implementations: Sections 5.4 and 5.5 give details on the specific weak clustering algorithms and threshold score determination methods we have used, respectively. Section 5.6 contains the details and results of an empirical evaluation of the proposed approaches. Finally, Section 5.7 draws conclusions of our work.*

Part of this work is described in (Gonzàlez and Turmo, 2009).

In the last two chapters we have explored the problem of clustering, focusing on the task of unsupervised clustering in particular, and developing a clustering-based method for IE pattern learning. In the presented setting, which is by far the most common, it was assumed that all objects belonged to some cluster. Even the diverse surveys that have so far reviewed the vast literature on clustering methods (the previously mentioned ones of Dubes and Jain, 1980; Jain et al., 1999; Xu and Wunsch, 2005), have all focused on this standard task, which can be named *all-in clustering.*

5.1 Minority Clustering

There is a number of situations in which the data are known not to fit neatly within this all-in assumption. In such cases, we know there is a fraction of data which are neither similar to one another nor to the data within the clusters. Often, these data will correspond to a certain form of *noise* and should hence be separated from the sought regular clusters, which constitute the *signal*. Within this alternative setting, a number of different tasks can be identified according to the characteristics of the data and the aim of the task itself.

In one of these tasks, the all-in clustering goal is preserved, but the data are known to contain a small fraction of noise. This has been called the *robust clustering* task (Davé and Krishnapuram, 1997). To solve it, some authors have proposed changes to standard clustering methods to make them more robust to the presence of noise. The replacement of the centroid calculation in k-means by that of medoids in the k-medoids or partitioning about medoids (PAM; Kaufman and Rousseeuw, 2005, Ch. 2) algorithm, or the use of mixtures of Student t distributions instead of Gaussian ones (Peel and McLachlan, 2000) are examples of work in this direction.

In other approaches to the task, robustness is increased by explicitly incorporating a noise cluster, often with different properties from the regular signal clusters. For instance, distance-based methods have been extended to incorporate an ideal noise prototype, "*a universal entity such that it is always at the same distance from every point in the data-set*" (Davé, 1991); and model-based clustering methods have been proposed which incorporate, among a mixture of otherwise Gaussian components, an extra one with a Poisson (Banfield and Raftery, 1993) or uniform (Guillemaud and Brady, 1997; Biernacki et al., 2000, §4.2.3) distribution to account for noise.

A last family of approaches is that of algorithms specifically devised for robust clustering, such as BIRCH (Zhang et al., 1996) or DBSCAN (Ester et al., 1996).

It is worth noticing that there is a number of related tasks which share this setting, such as *one-class classification* or *learning* (Moya et al., 1993; Schölkopf et al., 2001; Tax and Duin, 2004) and *outlier detection* (Hodge and Austin, 2004; Chandola et al., 2009). In both cases, there is also a dataset containing both signal and a fraction of noise objects. However, the focus of these tasks shifts away from that of clustering, becoming the estimation of a model which covers the signal objects in the former, and the detection of the objects that significantly deviate from the rest in the latter.

Nevertheless, there is still another setting to be considered. In some cases, there will only be a minority of signal objects, standing against the majority of noise. Most often, the signal objects will be embedded within the noise ones, becoming respectively *foreground* and *background* objects, and the distinction between the former and the latter must be done on grounds of density criteria. In the literature, this task has been compared to "*clustering needles in a haystack*" (Ando, 2007), and has received names such as *one-class clustering* (Crammer and Chechik, 2004), *density-based clustering* (Gupta and Ghosh, 2006) or *minority detection* (Ando and Suzuki, 2006). As a catchall term, in this paper we will refer to this setting and task as *minority clustering*.

Even if this new task is related to the previously presented ones, the reversal of the signal-to-noise ratio can make existing approaches unsuitable. For instance, Crammer and Chechik (2004) give insights into why existing one-class classification approaches, which are tailored to finding large-scale structures, may be unable to identify small and locally dense regions embedded in noise. Empirical comparisons have also stated the low performance exhibited by all-in and robust clustering methods in the minority clustering task (Gupta and Ghosh, 2006).

However, to the best of our knowledge, all the methods proposed so far require as an input the distribution of the foreground clusters or both the foreground clusters and the background noise, either in the form of a probability distribution or, equivalently, of a divergence metric[1]. This can become a significant issue when facing large amounts of data coming from a new and unexplored domain, whose distribution may be completely unknown.

5.1.1 Ensemble Minority Clustering

As mentioned in Chapter 3, in the context of supervised learning, combination methods have been successfully used to overcome the limitations of individual algorithms. They provide a way to obtain distribution-free learners able to perform competitively across a wide spectrum of learning

[1] A Bregman divergence induces a probability distribution of the exponential family (Banerjee et al., 2005)

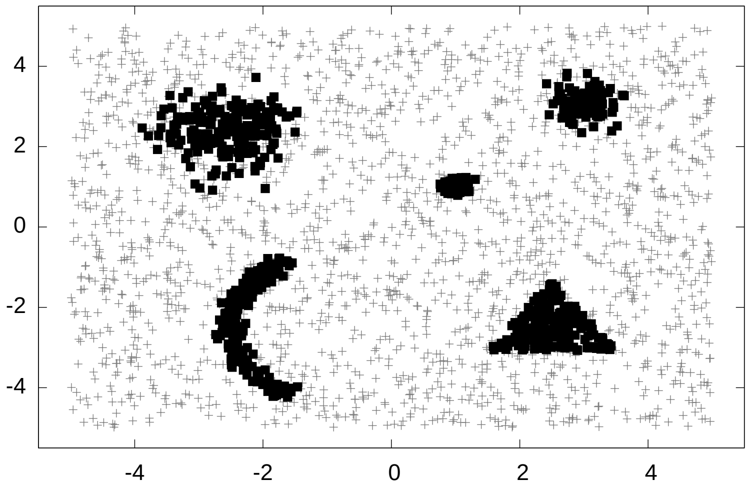


Figure 5.1: Sample Toy minority clustering dataset

problems, even from the combination of the outputs of weak learning algorithms (Freund and Schapire, 1995). Section 3.2.2 also described how, more recently, a number of combination methods have appeared for all-in clustering (e.g., Strehl and Ghosh, 2002; Topchy et al., 2003, 2004; Gionis et al., 2005)—and how, among them, Topchy et al. (2003) introduced the idea of using an ensemble of weak, almost random, clusterings to obtain a high-quality consensus clustering.

In this chapter, we make a three-fold proposal:

- First, we propose an unsupervised minority clustering approach, Ensemble Weak minOrity Cluster Scoring (Ewocs), based on weak-clustering combination. In it, a number of weak clusterings is generated, and the information coming from each one of them is combined to obtain a score for each object. A threshold separating foreground from background objects is then inferred from the distribution of these scores. We have been able to find a theoretical proof of the properties of the proposed method, and we consider a number of criteria by which the threshold value can be determined.
- Second, we propose Random Bregman Clustering (Rbc), a weak clustering algorithm based on Bregman divergences, for use within Ewocs ensembles; as well as an extension of the Random Splitting (RSplit) weak clustering algorithm of Topchy et al. (2003).
- Third, we propose an unsupervised procedure to determine a set of suitable scaling parameters for a Gaussian kernel, to be used within Rbc.

We have implemented a number of approaches built from the proposed components, and evaluated them on a collection of datasets. The results of the evaluation show how approaches based on Ewocs are competitive with respect to—and even outperform—other minority clustering approaches in the state of the art, in terms of F1 and AUC measures of the obtained clusterings.

5.2 Related Work

One of the first works to identify the minority clustering task in opposition to that of one-class classification is that of Crammer and Chechik (2004). The authors formalize the problem in terms of the Information Bottleneck principle (IB; Tishby et al., 1999), and provide a sequential algorithm to solve this one-class IB problem. Given a Bregman divergence (Bregman, 1967) as a generalized measure of object discrepancy, and a fixed radius value, the OC-IB method outputs a centroid for a single dense cluster. The foreground cluster consists of the objects which fall inside the Bregmanian ball of given radius centered around the given centroid. More recently, Crammer et al. (2008) propose a different algorithm for the same model, based in rate-distortion theory and the Blahut-Arimoto algorithm (Blahut, 1972; Arimoto, 1972), and extend it to allow for more than one cluster.

In a different direction, Gupta and Ghosh (2005) reformulate the problem in terms of cost, defined as the sum of divergences from the cluster centroid to each sample within it, and extend the OC-IB method to avoid local minima. A triad of methods (HOCC, BBOCC and Hyper-BB) is proposed. However, the requirement of an a priori determination of the cluster radius (or equivalently, size) is not removed, and the output remains a single ball-shaped cluster.

To overcome this second limitation, Gupta and Ghosh (2006) propose Bregman Bubble Clustering (BBC), as a generalization of BBOCC to several clusters. However, the number of such clusters must still be given a priori, as well as the desired joint cluster size. The authors also propose a soft clustering version of BBC, as well as a unified framework between all-in Bregman clustering (Banerjee et al., 2005) and BBC, in all their hard and soft versions.

The work of Ando and Suzuki (2006) is similar to previous ones in that it also uses the Information Bottleneck principle as a criterion to identify a single minority cluster. However, the method is more general in the sense that it allows arbitrary distributions, not only those induced by Bregman divergences, as foreground and background. Ando (2007) extends this last proposal, allowing multiple foreground clusters, and also provides a unifying framework of which not only the task of minority clustering, but also those of outlier detection and one-class learning, are particular cases.

5.3 Ewocs

This section presents our Ensemble Weak minOrity Cluster Scoring (Ewocs) algorithm to solve the task of minority clustering.

First Section 5.3.1 defines our setting for the task of minority clustering. Section 5.3.2 presents, from a theoretical point of view, the scoring scheme that lies at the core of our method. Sections 5.3.3 and 5.3.4 then study the conditional probability distributions of the assigned scores: the first one on a single dataset; the second, across multiple dataset samplings. Next, Section 5.3.5 introduces the concept of *consistent clustering*, and shows how, when using clustering functions from a consistent family, an inequality on the score expectations for foreground and background objects can be established. This inequality will allow us to obtain as a corollary, in Section 5.3.6, a generic algorithmic procedure for minority clustering, based on the proposed scores. Finally, it is also possible to obtain a clustering model using this algorithm: its construction and application is described in the last Section 5.3.7.

5.3.1 Task Setting

Assume we have a finite set of $\hat{k}$ generative distributions or *sources* $\Psi = \{\psi_1 \dots \psi_{\hat{k}}\}$, with a priori probabilities $\{\alpha_1 \dots \alpha_{\hat{k}}\}$. Assume we also have a dataset $\mathcal{X} = \{x_1 \dots x_n\}$ of size n, which has been sampled from Ψ. Each object x_i will be generated by one of the sources in Ψ, and we can hence consider a set $\mathcal{Y}$ of hidden variables, with each $y_i \in \Psi$ containing the source which generated the corresponding x_i.

Without loss of generality, we will name the first of those sources, ψ_1, the *background source*; and the objects generated by it, the *background objects*. The rest of sources and objects shall be named the *foreground sources* (whose set will be denoted as Ψ^+) and the *foreground objects*, respectively.

In the setting we are interested in, we can make two assumptions which can be stated as follows:

Density Foreground sources are *dense*, i.e., objects generated by the same foreground source are more similar to each other than to those generated by the background source.

Locality Foreground sources are *local*, i.e., objects generated by different foreground sources are as similar to each other as they are to those generated by the background source

These assumptions are similar to those in other works, for instance, those of *atypicalness* and *local distribution* defined by Ando (2007).

We can now recall the definitions of *hard* and *soft clustering* presented in Section 3.3, as they remain valid in this setting, and assume we have a (possibly infinite) family of *clustering functions* F. From them, a sequence of functions $(f_1 \dots)$ are independently drawn at random, with a certain probability density. When applied to the dataset, each f_r will produce a soft[2] clustering $\Pi_r = \{\pi_{r1} \dots \pi_{rk_r}\}$ with a number k_r of clusters.

5.3.2 Per-Clustering Scoring

After clustering function f_r is applied, the *cluster size* and *object scores* can be calculated from the output clustering Π_r.

Definition 5.1 (Cluster size)
*The **size** of cluster π_{rc} is the sum of the grade of membership to the cluster of all objects in the dataset:*

$$\text{size}(\pi_{rc}) = \sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc}) \tag{5.1}$$

Definition 5.2 (Object score)
*The **score** of an object x_i by clustering function f_r is*

$$s_{ri} = \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc}) \tag{5.2}$$

i.e., the sum of the sizes of the output clusters, weighted by the grade of membership of x_i to each one of them.

An additional concept will turn out to be of much importance later.

Definition 5.3 (Co-occurrence vector)
*The **co-occurrence vector** for object x_i and clustering function f_r is $\vec{c}_{ri} = [c_{ri1} \dots c_{rin}]^T$, where each component c_{rij} is*

$$c_{rij} = \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) \tag{5.3}$$

Remark Using the co-occurrence vector, the score of object x_i by clustering function f_r can be written as

$$\begin{aligned} s_{ri} &= \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc}) \\ &= \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \sum_{x_j \in \mathcal{X}} \text{grade}(x_j, \pi_{rc}) \\ &= \sum_{\pi_{rc} \in \Pi_r} \sum_{x_j \in \mathcal{X}} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) \\ &= \sum_{x_j \in \mathcal{X}} \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{grade}(x_j, \pi_{rc}) \\ &= \sum_{x_j \in \mathcal{X}} c_{rij} \end{aligned}$$

From its definition, we can infer that the co-occurrence vector will satisfy the following property:

Proposition 5.4
The values of the entries c_{rij} in the co-occurrence vector belong to the interval $[0, 1]$.

Proof By the properties of fuzzy pseudopartitions[3], and hence of soft clusterings, we know that

$$\forall x_i : \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) = 1$$

[2] The result is also valid for hard clustering families, being a particular case of soft clustering (see Definition 3.2).

[3] See Definition A.3 in Appendix A.2.

The product of two of these terms, which will also be equal to 1, can be expressed as

$$\begin{aligned}
1 &= \left(\sum_{\pi_{rc}\in\Pi_r} \mathrm{grade}(x_i,\pi_{rc})\right)\cdot\left(\sum_{\pi_{rc}\in\Pi_r} \mathrm{grade}(x_j,\pi_{rc})\right)\\
&= \sum_{\pi_{rc},\pi_{rc'}\in\Pi_r} \mathrm{grade}(x_i,\pi_{rc})\cdot\mathrm{grade}(x_j,\pi_{rc'})\\
&= \sum_{\pi_{rc}\in\Pi_r} \mathrm{grade}(x_i,\pi_{rc})\cdot\mathrm{grade}(x_j,\pi_{rc}) + \sum_{\substack{\pi_{rc},\pi_{rc'}\in\Pi_r\\ \pi_{rc}\neq\pi_{rc'}}} \mathrm{grade}(x_i,\pi_{rc})\cdot\mathrm{grade}(x_j,\pi_{rc'})\\
&= c_{rij} + \triangledown c_{rij}
\end{aligned}$$

Given that the grade of membership is by definition non-negative[4], all pairwise products of grades will also be non-negative—and, being sums of pairwise products, both c_{rij} and $\triangledown c_{rij}$ will at their turn be non-negative too: $0 \leq c_{rij}, \triangledown c_{rij}$.

Finally, given that c_{rij} and $\triangledown c_{rij}$ are two non-negative terms adding up to 1, it is clear that neither of them can exceed this value: $c_{rij}, \triangledown c_{rij} \leq 1$. Hence, as we wanted to prove, $0 \leq c_{rij} \leq 1$ ■

Rather than considering a single application of one clustering function $f_r \in F$ on $\mathcal{X}$, we will mainly be concerned with aggregating the results over a number R of repetitions of the process. In this context, we can define:

Definition 5.5 (Average co-occurrence vector)
The sequence of ***average co-occurrence vectors*** *for object x_i is $(\vec{c}^{\star}_{1i}\ldots)$, where each component of $\vec{c}^{\star}_{Ri} = [c^{\star}_{Ri1}\ldots c^{\star}_{Rin}]^T$ is*

$$c^{\star}_{Rij} = \frac{1}{R}\sum_{r=1}^{R} c_{rij} \tag{5.4}$$

Definition 5.6 (Average score)
The sequence of ***average scores*** *of object x_i is $(s^{\star}_{1i}, s^{\star}_{2i}\ldots)$, where each $s^{\star}_{Ri}$ is*

$$s^{\star}_{Ri} = \frac{1}{R}\sum_{r=1}^{R} s_{ri} \tag{5.5}$$

Remark Using average co-occurrence vectors, the average score of object x_i can be expressed as

$$s^{\star}_{Ri} = \frac{1}{R}\sum_{r=1}^{R} s_{ri} = \frac{1}{R}\sum_{r=1}^{R}\sum_{x_j\in\mathcal{X}} c_{rij} = \sum_{x_j\in\mathcal{X}}\frac{1}{R}\sum_{r=1}^{R} c_{rij} = \sum_{x_j\in\mathcal{X}} c^{\star}_{Rij}$$

It is interesting to note that

Proposition 5.7
The s_{ri} are linear transformations of $\vec{c}_{ri}$, and the $s^{\star}_{Ri}$ are linear transformations of $\vec{c}^{\star}_{Ri}$.

Proof Using an all-ones vector,

$$\begin{aligned}
s_{ri} &= \vec{1}^T\cdot\vec{c}_{ri} = \begin{bmatrix} 1 & 1 & \cdots & 1\end{bmatrix}\cdot\begin{bmatrix} c_{ri1} & c_{ri2} & \cdots & c_{rin}\end{bmatrix}^T = \sum_{x_j\in\mathcal{X}} c_{rij}\\
s^{\star}_{Ri} &= \vec{1}^T\cdot\vec{c}^{\star}_{Ri} = \begin{bmatrix} 1 & 1 & \cdots & 1\end{bmatrix}\cdot\begin{bmatrix} c^{\star}_{Ri1} & c^{\star}_{Ri2} & \cdots & c^{\star}_{Rin}\end{bmatrix}^T = \sum_{x_j\in\mathcal{X}} c^{\star}_{Rij}
\end{aligned}$$

[4] See Definition A.2 in Appendix A.2.

5.3.3 Dataset-Conditioned Distribution

The dataset $\mathcal{X}$ and clustering function f_r uniquely determine the values for the co-occurrence vectors $\vec{c}_{ri}$, and hence for all other values considered in the previous Section. However, as the selection of f_r is not deterministic, the c_{rij} can be regarded as random variables, and their conditional distribution across clustering functions, given a certain dataset $\mathcal{X}$, can be considered.

As the selection of each f_r is independent from the others, the values of the c_{rij} for different r will also be. The $\vec{c}_{ri}$ for different r will hence be independent and identically distributed random vectors, with a common expectation vector $\vec{\mu}_i$ and covariance matrix Σ_i. We will refer to each element, μ_{ij}, of $\vec{\mu}_i$ as the *affinity* of x_i and x_j.

Definition 5.8 (Object affinity)
The ***affinity*** *of objects x_i and x_j is the conditional expectation of c_{rij} given $\mathcal{X}$,*

$$\mu_{ij} = E[c_{rij} \mid \mathcal{X}] \tag{5.6}$$

Remark Being the expectations of the c_{rij}, with $c_{rij} \in [0,1]$, the affinities μ_{ij} will also fall in the $[0,1]$ interval.

We can additionally define

Definition 5.9 (Object expected score)
The ***expected score*** *of object x_i is the conditional expectation of s_{ri} given $\mathcal{X}$,*

$$\mu_i = E[s_{ri} \mid \mathcal{X}] \tag{5.7}$$

It is then easy to successively prove that

Proposition 5.10
The value of the expected score μ_i of object x_i is

$$\mu_i = E[s_{ri} \mid \mathcal{X}] = \sum_{x_j \in \mathcal{X}} \mu_{ij} \tag{5.8}$$

Proof As s_{ri} is the sum of the c_{rij}, its conditional expectation is

$$\mu_i = E[s_{ri} \mid \mathcal{X}] = E[\sum_{x_j \in \mathcal{X}} c_{rij} \mid \mathcal{X}] = \sum_{x_j \in \mathcal{X}} E[c_{rij} \mid \mathcal{X}] = \sum_{x_j \in \mathcal{X}} \mu_{ij}$$

Remark Being the sum of $n = |\mathcal{X}|$ terms within the interval $[0,1]$, the value of μ_i will fall in the interval $[0,n]$. In order to make scores across differently-sized datasets comparable, we will also consider a **normalized expected score** $\bar{\mu}_i$, defined as $\bar{\mu}_i = \mu_i / n$.

Proposition 5.11
As the number of repetitions R increases, the conditional distributions of the average co-occurrence vectors $\vec{c}^{\star}_{Ri}$ approach a multivariate Gaussian distribution with expectation $\vec{\mu}_i$ and covariance matrix Σ_i / R.

Proof As the c_{rij} are independent and identically distributed for different r, by the multivariate central limit theorem we know that the sequence

$$\sqrt{R}\left(\frac{1}{R}\sum_{r=1}^{R}\vec{c}_{ri} - \vec{\mu}_i\right) = \sqrt{R}\,(\vec{c}^{\star}_{Ri} - \vec{\mu}_i)$$

converges in distribution to a multivariate Gaussian distribution with expectation $\vec{\mu}_i$ and covariance matrix Σ_i. Hence, for large enough R,

$$\begin{aligned} \sqrt{R}\,(\vec{c}^{\star}_{Ri} - \vec{\mu}_i) &\approx \mathcal{N}(0, \Sigma_i) \\ \vec{c}^{\star}_{Ri} - \vec{\mu}_i &\approx \mathcal{N}(0, \Sigma_i / R) \\ \vec{c}^{\star}_{Ri} &\approx \mathcal{N}(\vec{\mu}_i, \Sigma_i / R) \end{aligned}$$

Proposition 5.12
As the number of repetitions R increases, the conditional distributions of the average scores $s^\star_{Ri}$ approach a Gaussian distribution with expectation μ_i.

Proof Being linear transformations of random vectors $\vec{c}^\star_{Ri}$ approaching a multivariate Gaussian distribution, the $s^\star_{Ri}$ also approach a Gaussian distribution

$$s^\star_{Ri} = \vec{1}^T \cdot \vec{c}^\star_{Ri} \quad \approx \quad \mathcal{N}(\vec{1}^T \cdot \vec{\mu}_i, (\Sigma^\star_{Ri})^2)$$

with a certain variance $(\Sigma^\star_{Ri})^2$. The conditional expectation of these variables hence converges to

$$\lim_{R\to\infty} E[s^\star_{Ri} \mid \mathcal{X}] = \vec{1}^T \cdot \vec{\mu}_i = \sum_{x_j\in\mathcal{X}} \mu_{ij} = \mu_i$$

5.3.4 Sampling Distribution

We can now proceed to consider the distribution of the scores across multiple samplings of the dataset $\mathcal{X}$. In particular, we will first focus on the distribution of the affinity μ_{ij} between objects x_i and x_j, conditioned to their being respectively generated by a certain pair of sources ψ_s and ψ_t—a measure which we shall name the *affinity* of the two sources, ζ_{st}.

Definition 5.13 (Source affinity)
*The **affinity** of sources ψ_s and ψ_t is the conditional expectation of the object affinity μ_{ij}, given that $y_i = \psi_s$ and $y_j = \psi_t$, across all datasets $\mathcal{X}$ sampled from Ψ:*

$$\zeta_{st} = E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t]$$

A particular case of affinity is that of $\psi_t = \psi_s$, which we shall name the *self-affinity* ζ_{ss} of source ψ_s.

We can now also consider the conditional expectation of the normalized expected scores $\bar{\mu}_i$ for objects from source ψ_s.

Definition 5.14 (Source normalized expected score)
*The **normalized expected score** of a source ψ_s is the conditional expectation of the normalized expected score $\bar{\mu}_i$ of objects x_i generated by ψ_s, across all datasets $\mathcal{X}$ sampled from Ψ:*

$$\zeta_s = E[\bar{\mu}_i \mid y_i = \psi_s]$$

This newly defined score satisfies that:

Proposition 5.15
The value of the normalized expected score ζ_s for a source ψ_s is

$$\zeta_s = \sum_{\psi_t\in\Psi} \alpha_t \cdot \zeta_{st}$$

Proof The value of $\bar{\mu}_i$ is

$$\bar{\mu}_i = \frac{1}{n}\mu_i = \frac{1}{n}\sum_{x_j\in\mathcal{X}} \mu_{ij}$$

The conditional expectation of $\bar{\mu}_i$ across samplings of $\mathcal{X}$ for which $|\mathcal{X}| = n$ can then be found as

$$\begin{aligned} E[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}| = n] &= E\left[\frac{1}{n}\sum_{x_j\in\mathcal{X}} \mu_{ij} \,\middle|\, y_i = \psi_s, |\mathcal{X}| = n\right] \\ &= \frac{1}{n}E\left[\sum_{x_j\in\mathcal{X}} \mu_{ij} \,\middle|\, y_i = \psi_s, |\mathcal{X}| = n\right] \end{aligned}$$

Assuming the $x_j \in \mathcal{X}$ are independent and identically distributed, and using the law of total expectation, this can be expressed as

$$\begin{aligned}
E[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}| = n] &= \frac{1}{n} \sum_{x_j \in \mathcal{X}} E\left[\mu_{ij} \,|\, y_i = \psi_s, |\mathcal{X}| = n\right] \\
&= \frac{1}{n} \sum_{x_j \in \mathcal{X}} \sum_{\psi_t \in \Psi} P(y_j = \psi_t) \cdot E\left[\mu_{ij} \,|\, y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n\right] \\
&= \frac{1}{n} \sum_{x_j \in \mathcal{X}} \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n] \\
&= \frac{1}{n} \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n] \cdot \sum_{x_j \in \mathcal{X}} 1 \\
&= \frac{1}{n} \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n] \cdot n \\
&= \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n]
\end{aligned}$$

Finally, assuming independence of normalized expected scores and source affinities with respect to dataset size n, and plugging the definition of the latter into the above formula, we obtain the desired result:

$$\begin{aligned}
E[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}| = n] &= \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t, |\mathcal{X}| = n] \\
\zeta_s = E[\bar{\mu}_i \mid y_i = \psi_s] &= \sum_{\psi_t \in \Psi} \alpha_t \cdot E[\mu_{ij} \mid y_i = \psi_s, y_j = \psi_t] = \sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{st}
\end{aligned}$$

5.3.5 Consistent Clustering

We will now impose some conditions on the used clustering families, with respect to how they preserve the density and locality of the sources in Ψ. We will start by considering the *detectability* of a source by a clustering family:

Definition 5.16 (Source detectability)
Given a set of sources Ψ and a clustering family F, a foreground source $\psi_s \in \Psi^+$ is ***detectable by*** *F if and only if its normalized expected score ζ_s is larger than that ζ_1 of the background source ψ_1.*

Proposition 5.17 (Detectability criterion)
Given a set of sources Ψ and a clustering family F, a foreground source $\psi_s \in \Psi^+$ is detectable by F if and only if:

$$\alpha_s \cdot (\zeta_{ss} - \zeta_{1s}) > \alpha_1 \cdot (\zeta_{11} - \zeta_{s1}) + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot (\zeta_{1t} - \zeta_{st})$$

Proof From the definition of detectability and Proposition 5.15,

$$\begin{aligned}
\zeta_s &> \zeta_1 \\
\sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{st} &> \sum_{\psi_t \in \Psi} \alpha_t \cdot \zeta_{1t} \\
\alpha_s \cdot \zeta_{ss} + \alpha_1 \cdot \zeta_{s1} + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot \zeta_{st} &> \alpha_s \cdot \zeta_{1s} + \alpha_1 \cdot \zeta_{11} + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot \zeta_{1t} \\
\alpha_s \cdot (\zeta_{ss} - \zeta_{1s}) &> \alpha_1 \cdot (\zeta_{11} - \zeta_{s1}) + \sum_{\substack{\psi_t \in \Psi^+ \\ \psi_t \neq \psi_s}} \alpha_t \cdot (\zeta_{1t} - \zeta_{st})
\end{aligned}$$

Remark This arrangement of the terms in the difference $\zeta_s - \zeta_1$ is intended to capture the degree to which the clustering family captures the *density* and *locality* properties of the data in the minority clustering setting:

- For *dense* sources, self-affinity should be much larger than affinity to the background source. Therefore, the value of the left-side term should be large.
- For *local* sources, affinity to the background source and to other foreground sources should not be much different than their affinity to the background source itself. Therefore, the value of the right-side term should be small.

If a clustering family respects the density and locality of all foreground sources in a set, all of them will be detectable. In this case, the family is said to be consistent with the source set:

Definition 5.18 (Clustering family consistency)
Given a set of sources Ψ, a clustering family F is ***consistent with*** *Ψ if and only if all foreground sources $\psi_s \in \Psi^+$ are detectable by F.*

The importance of detectable sources and consistent families lies in the fact that:

Theorem 5.19
Given a dataset $\mathcal{X}$ sampled from a set of sources Ψ and a consistent clustering family F, for a sufficiently large number of repetitions R, the expected value of the average score $s^\star_{Ri}$ of objects x_i generated by a foreground source $\psi_s \in \Psi^+$ is larger than the expected value of the average scores $s^\star_{Rj}$ of objects x_j generated by the background source ψ_1.

Proof Using $n = |\mathcal{X}|$, replacing the definitions of the different used quantities, and applying properties of the expectation, we know that, if ψ_s is detectable,

$$\begin{aligned} \zeta_s &> \zeta_1 \\ n \cdot \zeta_s &> n \cdot \zeta_1 \\ n \cdot E[\bar{\mu}_i \mid y_i = \psi_s] &> n \cdot E[\bar{\mu}_j \mid y_j = \psi_1] \end{aligned}$$

Assuming independence on the size of the dataset $\mathcal{X}$,

$$\begin{aligned} n \cdot E[\bar{\mu}_i \mid y_i = \psi_s, |\mathcal{X}'| = n] &> n \cdot E[\bar{\mu}_j \mid y_j = \psi_1, |\mathcal{X}'| = n] \\ n \cdot E[\mu_i/n \mid y_i = \psi_s, |\mathcal{X}'| = n] &> n \cdot E[\mu_j/n \mid y_j = \psi_1, |\mathcal{X}'| = n] \\ n \cdot E[E[s^\star_{Ri} \mid y_i = \psi_s, \mathcal{X}', |\mathcal{X}'| = n]]/n &> n \cdot E[E[s^\star_{Rj} \mid y_j = \psi_1, \mathcal{X}', |\mathcal{X}'| = n]]/n \\ E[s^\star_{Ri} \mid y_i = \psi_s, \mathcal{X}', |\mathcal{X}'| = n] &> E[s^\star_{Rj} \mid y_j = \psi_1, \mathcal{X}', |\mathcal{X}'| = n] \end{aligned}$$

which, assuming independence again, leads to

$$E[s^\star_{Ri} \mid y_i = \psi_s, \mathcal{X}] > E[s^\star_{Rj} \mid y_j = \psi_1, \mathcal{X}]$$

5.3.6 Algorithm

A corollary of this last Theorem 5.19 is

Corollary 5.20
Given a dataset $\mathcal{X}$ sampled from a set of sources Ψ, and using a clustering family F which is consistent with Ψ, we can devise an algorithmic procedure to obtain a minority clustering of $\mathcal{X}$.

Proof Given a dataset $\mathcal{X}$, we can apply a sequence of clustering functions f_r, drawn from F, and find the average score $s^\star_{Ri}$ for each object $x_i \in \mathcal{X}$. The expected value of the average scores of the background objects will be lower than that of the foreground ones. If a suitable threshold value is determined, we will be able to discriminate most foreground and background objects according to their score. ■

Algorithm 5.1 Ensemble Weak minOrity Cluster Scoring (Ewocs)

```
Input: A dataset $\mathcal{X}$
Input: A consistent clustering family $F$
Input: An ensemble size $R$
Output: A hard minority clustering $\Pi$ of $\mathcal{X}$
1: Initialize the accumulated scores of all objects $x_i$ to zero,
       $s_i^+ = 0$
2: For $r = 1$ to $R$ do
3:     Draw a clustering function $f_r$ at random from $F$,
           $f_r \in F$
4:     Apply $f_r$ to obtain clustering $\Pi_r$,
           $\Pi_r = f_r(\mathcal{X})$
5:     Find cluster sizes,
           $\text{size}(\pi_{rc}) = \sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc})$
6:     Update the accumulated scores of each object,
           $s_i^+ \leftarrow s_i^+ + s_{ri} = s_i^+ + \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc})$
7: Find the final average scores of each object,
       $s_{Ri}^\star = \frac{s_i^+}{R}$
8: Determine a threshold $s_{th}^\star$ separating the scores,
       $s_{th}^\star = \textbf{find\_threshold}(s_{R1}^\star \dots s_{Rn}^\star)$
9: Create the foreground and background clusters, $\pi_f$ and $\pi_b$,
       $\pi_f = \{x_i \mid s_{Ri}^\star \geq s_{th}^\star\}$
       $\pi_b = \{x_i \mid s_{Ri}^\star < s_{th}^\star\}$
10: Return The minority clustering $\Pi = \{\pi_b, \pi_f\}$
```

The resulting algorithm, which we have named Ensemble Weak minOrity Cluster Scoring (Ewocs), is described in Algorithm 5.1.

The first step of Ewocs is the initialization of an auxiliary array, which will contain the accumulated scores s_i^+ of all objects, to zero (line 1). The main loop is then entered (lines 2–6). The number of iterations of this loop, R, determines the ensemble size and is a user-supplied parameter. Larger values of R are expected to yield better results, but at the expense of a larger computational cost.

At each iteration, a clustering function f_r is drawn at random from family F (line 3) and is then applied to dataset $\mathcal{X}$ to obtain a clustering Π_r (line 4). The size of each cluster π_{rc} in clustering Π_r is then found (line 5), and then the score of each object, as defined in Equation 5.2, is found and added to the accumulated score s_i^+ (line 6).

When the main loop is over, the final average score of each object, $s_{Ri}^\star$ is found from the final accumulated score s_i^+ and the ensemble size R (line 7). From the distribution of these scores $s_{Ri}^\star$, a threshold value $s_{th}^\star$ which separates the scores of the foreground and the background objects is inferred (line 8). At this point, the only steps that remain are separating the objects according to their scores into a foreground and a background cluster (line 9) and returning the resulting clustering (line 10).

Algorithm 5.2 Classification using an Ewocs clustering model

```
Input: An Ewocs minority clustering model $\mathcal{M}^E = (\{\mathcal{M}_r\}, \{\text{size}(\pi_{rc})\}, s^+_{th})$
Input: An object $x_x$
Output: The cluster $\pi_x \in \{\pi_b, \pi_f\}$ to which $x_x$ would belong
1: Initialize the accumulated score of the object $x_x$ to zero,
       $s^+_x = 0$
2: For $r = 1$ to $R$ do
3:     Apply the clustering model $\mathcal{M}_r$ to obtain the grade of membership of $x_x$ to each $\pi_{rc}$
           $(\text{grade}(x_x, \pi_{r1}) \dots \text{grade}(x_x, \pi_{rk_r})) = \mathcal{M}_r(x_x)$
4:     Update the accumulated score of the object
           $s^+_x \leftarrow s^+_x + s_{rx} = s^+_x + \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_x, \pi_{rc}) \cdot \text{size}(\pi_{rc})$
5: Find the final average score of the object
       $s^\star_{Rx} = \frac{s^+_x}{R}$
6: Assign the object to the foreground or background cluster, $\pi_f$ or $\pi_b$, according to the relation
   between its average score and the separating threshold
       $\pi_x = \begin{cases} \pi_f & \text{if } s^\star_{Rx} \geq s^\star_{th} \\ \pi_b & \text{if } s^\star_{Rx} < s^\star_{th} \end{cases}$
7: Return The object cluster $\pi_x$
```

The obtained Ewocs algorithm has a number of components which allow different implementations: neither the consistent clustering function family F (line 3) nor the method for the determination of the threshold score separating foreground and background objects (line 8) are specified. As mentioned in the introduction, the following two sections, 5.4 and 5.5, give insights into each one of these two issues, respectively.

5.3.7 Clustering Model

As mentioned in Section 3.3.4, most clustering algorithms provide, in addition to an output clustering, a clustering model which allows the (hard or soft) assignment of new objects to the obtained clusters. In the case of Ewocs, if the functions in the used family F provide models together with the clusterings when applied to dataset $\mathcal{X}$, these individual models can be extended to obtain an aggregated minority clustering model.

More specifically, if the application of $f_r \in F$ to $\mathcal{X}$ produces clustering Π_r and clustering model $\mathcal{M}_r$, after Algorithm 5.1, an Ewocs minority clustering model $\mathcal{M}^E$ can be constructed, containing:

- the inner clustering models $\mathcal{M}_r$,
- the size of each cluster π_{rc} in the clusterings Π_r,
- and the threshold value $s^\star_{th}$ which separates foreground and background objects.

The process of classifying a new object x_x using the obtained model $\mathcal{M}$ is described in Algorithm 5.2. It follows the main steps of the previous Algorithm 5.1, but replacing the application of new clustering functions $f_r \in F$, by that of the previously obtained clustering models $\mathcal{M}_r$ (line 3). After all models have been applied, the average score of the object is found (line 5), and the object is deemed to belong to the foreground or background cluster according to whether its score exceeds the previously found threshold (line 6).

5.4 Weak Clustering

As stated in Section 5.3.5, the theoretical properties of the Ewocs algorithm depend only on the condition of the used clustering family being consistent. We believe that the requirements for being consistent, according to Definition 5.18, should be fairly loose—and that, hence, the Ewocs algorithm is suitable for use with weak clustering algorithms.

In this context, a clustering function family F is a clustering algorithm which includes elements of randomness. Each sequence of random values will determine a member function of the family. From a conceptual point of view, drawing a function f_r from the family F will hence correspond to drawing a sequence of random values to be later used by the algorithm. From a computational one, it can correspond, for instance, to choosing a seed value for the algorithm's internal random number generator.

The two weak clustering algorithms that are used in the work of Topchy et al. are based on either splitting the dataset using random hyperplanes, or on clustering projections of the data on random subspaces. We found the first of them particularly convenient for our purposes, and extended it. Section 5.4.1 reviews this our extension of the random splitting algorithm.

However, even if these methods have been proved to produce clusterings useful for combination within an ensemble, they both perform linear mappings of the data and, hence, are based on the notion of linear separation. Although non-linearly separable clusters can be successfully identified by linear separators, non-linear weak separators have not been thoroughly explored. Besides, linear methods depend on the data being expressible as feature vectors, and hence cannot directly deal with structured objects such as sequences or trees.

Our proposal in this direction is a new weak clustering algorithm based on Bregman divergences, which allows non-linear splitting boundaries and, through the use of kernels, can deal with structured data. This proposed Random Bregman Clustering is described in Section 5.4.2.

Later, Section 5.6.4.2 will provide an estimation of the consistency of the proposed clustering families over a number of datasets. The results shall provide an empirical assessment of the suitability of these two families for use within Ewocs.

5.4.1 Random Splitting

The random splitting algorithm presented in Topchy et al. (2003) performs only binary bisections of the objects in the dataset. Our Random Splitting algorithm (RSplit) is a generalization of this algorithm, which allows an arbitrary number of clusters k.

For this algorithm we require the objects in dataset $\mathcal{X}$ to be expressible as z-dimensional real vectors (i.e., $\mathcal{X} \subset \mathbb{R}^z$). To account for multiple clusters, we have adopted the same representation of hyperplanes as in the Multi-Class Support Vector Machines of Crammer and Singer (2001): each splitting hyperplane is defined by a weight vector $\omega_c = (\omega_{c1} \dots \omega_{cz})$ and an offset δ_c, and objects belong to the cluster (class in the original formulation) from whose hyperplane they are separated by the largest margin.

Similarly to Topchy et al., in a clustering ensemble setting, the number of clusters k does not need to be given a priori, but is rather drawn at random between 2 and a user-supplied value k_{max}.

This idea leads to the simple procedure described in Algorithm 5.3. The algorithm takes three sequential steps. The first of them is the selection of the effective number of clusters k (line 1). Any discrete distribution between 2 and k_{max}, such as the uniform distribution, can be used. For each cluster π_c, random weights ω_c and offsets δ_c (line 2) are then generated. Again, we have stuck to the uniform distribution from all the possible continuous distributions within the $[-1 \dots 1]$ range.

Once these values are generated, the margin of each object x_i with respect to the hyperplanes is found as the dot product between the object x_i and the hyperplane's weight vector ω_c, shifted by the latter's offset δ_c. Each object is assigned to the cluster induced by the hyperplane to which its margin is maximal (line 3). The resulting clustering can then be returned (line 4).

The time complexity of this algorithm is dominated by the calculation of the margin in step (line 3), and is hence in the order of $O(k_{max} \cdot z \cdot |\mathcal{X}|)$.

We will henceforth refer to this algorithm as RSplit, and to its application within Ewocs as Ew-RSplit.

Algorithm 5.3 Random Splitting (RSPLIT)

```
Input: A dataset $\mathcal{X}$
Input: A maximum number of clusters $k_{max}$
Output: A hard all-in clustering $\Pi$ of $\mathcal{X}$
Draw a number of clusters $k$ at random from the range $\{2 \dots k_{max}\}$
    $k \in \{2 \dots k_{max}\}$
Generate a weight vector $\omega_c = [\omega_{c1} \dots \omega_{cz}]$ and an offset $\delta_c$ at random for each $c \in \{1 \dots k\}$
    $\omega_{c1} \dots \omega_{cz}, \delta_c \in [-1 \dots 1]$
Assign each object $x_i$ to the cluster $\pi_c$ whose hyperplane gives the largest margin
    $\pi_c = \{x_i \in \mathcal{X} \mid \arg\max_q \omega_q \cdot x_i + \delta_q = c\}$
Return The clustering $\Pi = \{\pi_1 \dots \pi_k\}$
```

5.4.2 Random Bregman Clustering

As stated in the introduction to Section 5.4, two desirable properties of weak clustering algorithms, but to which few attention has been devoted so far, are, first, the ability to find non-linear boundaries in vectorial data, and, second, the possibility to deal with non-vectorial and/or structured data. Kernel methods have a long story of successes across a wide spectrum of machine learning tasks (Shawe-Taylor and Cristianini, 2004) and, specifically, they are known for their capability to address both of these issues. The use of kernel functions allows to separate non-linearly separable classes, even with linear methods (Freund and Schapire, 1999); and kernels have been devised and successfully applied for non-vectorial objects such as word sequences (Cancedda et al., 2003) or parse trees (Collins and Duffy, 2002).

It is interesting to note that kernel functions induce a distance metric between objects. As proved by Mercer (1909), any kernel function K_ϕ is equivalent to an inner product in a high-dimensional space, onto which there will exist a certain mapping ϕ. Hence, if $\phi(x)$ and $\phi(y)$ are, respectively, the images of two objects x and y in this space, $K_\phi(x,y) = \phi(x) \cdot \phi(y)$. Their squared Euclidean distance on the mapped space, $D_\phi(x,y)$, can then be found as:

$$\begin{aligned} D_\phi(x,y) &= \|\phi(x) - \phi(y)\|^2 \\ &= (\phi(x) - \phi(y)) \cdot (\phi(x) - \phi(y)) \\ &= \phi(x) \cdot \phi(x) + \phi(y) \cdot \phi(y) - 2 \cdot \phi(x) \cdot \phi(y) \\ &= K_\phi(x,x) + K_\phi(y,y) - 2K_\phi(x,y) \end{aligned} \tag{5.9}$$

This transformation is the basis for existing kernel-based all-in clustering algorithms, such as kernel k-means (Girolami, 2002). In our case, given that these squared Euclidean distances will be, by construction, Bregman divergences, we can join Mercer kernel theory and that of Bregman clustering and devise a weak all-in clustering procedure. The idea is to randomly select a number of objects which can act as *seeds* for the clustering, and then define clusters according to the divergence from these seeds of the objects in the dataset. The resulting Random Bregman Clustering (RBC) method is described in Algorithm 5.4.

RBC is thus a seed-based algorithm, and in this sense is similar to the MAJOR clustering generation strategy presented in Section 3.4.2.2. Given dataset $\mathcal{X}$, a Bregman divergence D and a maximum number of clusters k_{max}, the first step of RBC is selecting the effective number of clusters in the clustering, k (line 1). Any discrete distribution between 2 and k_{max}, such as the uniform distribution, can be used. A subset $\hat{\mathcal{X}}$ of size k is then selected at random from $\mathcal{X}$ (line 2). We shall name this subset the *seed subset*, and each one of their members will be a *seed*. Each seed will induce a cluster in the output clustering.

The output clustering is constructed following the theoretical framework provided by Bregman clustering (Banerjee et al., 2005). First, the distance of each object $x_i \in \mathcal{X}$ to the seeds $\hat{x}_c \in \hat{\mathcal{X}}$ is found. If a hard clustering is desired, each object is then assigned to the cluster induced by its

Algorithm 5.4 Random Bregman Clustering (Rbc)

```
Input: A dataset $\mathcal{X}$
Input: A Bregman divergence $D$
Input: A maximum number of clusters $k_{max}$
Output: A (hard or soft) all-in clustering $\Pi$ of $\mathcal{X}$
1: Draw a number of clusters $k$ at random from the range $\{2 \dots k_{max}\}$
       $k \in \{2 \dots k_{max}\}$
2: Select a subset $\hat{\mathcal{X}}$ of $k$ seeds from $\mathcal{X}$
       $\hat{\mathcal{X}} = \{\hat{x}_1 \dots \hat{x}_k\} \subset \mathcal{X}$
3: If Hard clustering desired then
4:     Assign each object $x_i$ to the cluster $\pi_c$ induced by its nearest seed $\hat{x}_c$,
           $\pi_c = \{x_i \in \mathcal{X} \mid \arg\min_{\hat{x}_q \in \hat{\mathcal{X}}} D(\hat{x}_q, x_i) = \hat{x}_c\}$
5: Else
6:     Find membership grade for each object $x_i$ and cluster $\pi_c$,
           $\text{grade}(x_i, \pi_c) = \frac{e^{-D(\hat{x}_c, x_i)}}{\sum_{q=1}^{k} e^{-D(\hat{x}_q, x_i)}}$
7: Return The clustering $\Pi = \{\pi_1 \dots \pi_k\}$
```

nearest seed (line 4). If, instead, a soft clustering is desired, the grade of membership of each object to each cluster is proportional to the exponential of the negated divergence from the seed of the latter to the former (line 6). In both cases, the only remaining step is then returning the resulting (hard of soft) clustering (line 7).

The construction of the hard clustering is hence equivalent to a single assignment step of Bregman hard clustering; and that of the soft clustering is equivalent to a single expectation step of Bregman soft clustering, with a uniform *a priori* probability of membership to all clusters.

The time complexity of the Rbc algorithm is dominated by the clustering construction step (line 4 or 6), and, as long as the kernel computation does not depend on the maximum number of clusters k_{max} or on the size of the dataset $|\mathcal{X}|$, it is in the order of $O(k_{max} \cdot |\mathcal{X}|)$. This is comparable to the cost of RSplit, so the increase in expressiveness of the algorithm does not come at the expense of an increase in computational complexity. The algorithm hence remains inexpensive, and suitable for use in a weak clustering ensemble.

We will henceforth refer to the hard and soft versions of this algorithm as HRbc and SRbc, respectively, and to their application within Ewocs as Ew-HRbc and Ew-SRbc.

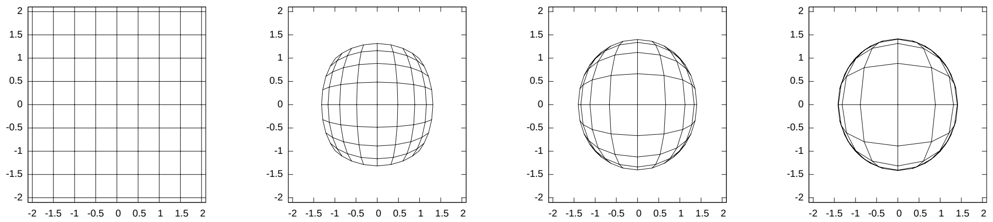
We have explored the use of two different families of Bregman divergences at the core of the Rbc algorithm presented above. The first one is the well-known Mahalanobis distance, whereas the second one is that of Gaussian-kernel-based distances. The following Sections 5.4.2.1 and 5.4.2.2, respectively, describe each one of them in more detail.

5.4.2.1 Mahalanobis Distance

Mahalanobis distance was first introduced by Mahalanobis (1936), and can be regarded as a version of standard Euclidean distance normalized for a particular dataset. Being μ the mean vector of the considered dataset, and Σ its covariance matrix, the squared Mahalanobis distance $D_M(x,y)$ between two objects x and y is:

$$D_M(x,y) = (x-\mu)^T \Sigma^{-1} (y-\mu) \tag{5.10}$$

Mahalanobis distance is a common choice for clustering tasks, and it has specifically been reported to give the best results within previous approaches to minority clustering (Gupta and Ghosh, 2005,



(a) Lineal (b) Gaussian, $\gamma = 0.5$ (c) Gaussian, $\gamma = 1.0$ (d) Gaussian, $\gamma = 2.0$

Figure 5.2: Comparison of Euclidean distances induced by different kernels

2006).

We will henceforth refer to the Mahalanobis distance as Mah.

5.4.2.2 Gaussian-Kernel Distance

Gaussian kernels are used in a wide variety of classification and clustering tasks. Given that they have been successfully applied in non-parametric (i.e., distribution-free) clustering algorithms, such as mean shift (Fukunaga and Hostetler, 1975; Cheng, 1995), they are a sensible choice for use within unsupervised clustering ensembles.

The Gaussian kernel $K_\phi(x,y)$ between two objects x and y is defined as the exponential of the negated squared Euclidean distance between them, with two additional scaling parameters α and γ:

$$K_\phi(x,y) = \alpha \cdot e^{-\gamma \|x-y\|^2} \tag{5.11}$$

By Equation 5.9, their induced squared Euclidean distance mapped space, $D_\phi(x,y)$, can be found as:

$$\begin{aligned} D_\phi(x,y) &= K_\phi(x,x) + K_\phi(y,y) - 2K_\phi(x,y) \\ &= \alpha + \alpha - 2\alpha \cdot e^{-\gamma\|x-y\|^2} \\ &= 2\alpha(1 - e^{-\gamma\|x-y\|^2}) \end{aligned} \tag{5.12}$$

A graphical comparison of standard Euclidean distance (which is induced by a linear kernel, the standard dot product) to Gaussian-kernel distance for $\alpha = 1.0$ and several values of γ can be found in Figure 5.2. In each subfigure, points in the grid are plotted at a distance to the origin which is proportional to the considered Euclidean distance induced by the kernel. Gaussian kernels locally map the Euclidean space around each point into a hypersphere of radius $\sqrt{2\alpha}$, and the rate at which neighbouring points are pushed apart towards the edge of the hypersphere increases with the value of parameter γ.

It is also interesting to note that, if this Gaussian-kernel distance is used in Rbc, small values of α lead to fuzzy boundaries between the clusters, whereas large values produce crisp ones. As a particular case, hard Rbc is equivalent to the limit of soft Rbc as $\alpha \to \infty$.

We will henceforth refer to the Gaussian-kernel-induced distance for certain α and γ as G(α,γ). In particular, the notation G(∞,γ) will be used within HRbc.

5.4.2.3 Unsupervised Tuning of Gaussian-Kernel Distance

The use of the presented Gaussian-kernel distance presents an obvious drawback, and that is the choice of the values for α and γ. As mentioned in the previous section, the choice of α and γ determine the degrees of fuzziness and locality of the output clustering, respectively. This can have a dramatic influence on the result of the clustering process, and different datasets will require different values of these parameters, not only to obtain quality clusterings, but even to avoid degenerate solutions. The determination of suitable values for α and γ hence can become a problematic issue, especially in unsupervised clustering settings.

Similar problems are to be addressed in all-in fuzzy clustering algorithms which depend on a parameter. Among them, the one that has received the most attention in the literature is the tuning of the *degree of fuzziness* parameter, traditionally referred to as m, of the fuzzy c-means algorithm (FCM; Bezdek, 1981). A series of works have proposed rules to find intervals inside which the optimal m might be found (Deer and Eklund, 2003; Yu et al., 2004), but only more recently procedures which provide a determinate value for m have been devised. In this direction, Okeke and Karnieli (2006) propose the use of grid search over m to minimize the sum of distances between the objects in the original dataset and their reconstructed images built using the clustering information. The value of m for which this sum of distances is minimal is selected.

A different approach is that of Schwämmle and Jensen (2010). In it, the authors study the behaviour of the cluster centroids as the degree of fuzziness m increases, and find that, at a certain point, the clustering degrades and the clusters start collapsing on each other. This phenomenon can be detected by watching the minimum distance between centroids: the moment the degradation starts, the first two clusters collapse and this distance becomes close to zero. It is interesting to note that, according to the authors, this happens however many clusters are used, even if the number does not match the actual one.

Given that "*a large fuzzifier value suppresses outliers in data sets*", the authors consider that maximum fuzziness should be sought, and hence propose selecting the largest m value for which the minimum centroid distance still remains above a predefined threshold ϵ (set so as to reduce floating-point errors).

We have adapted the approach of Schwämmle and Jensen to determine the optimal values of α and γ for Ew-SRbc. The method is particularly suitable to our needs: it does not depend on specific properties of FCM, nor requires knowledge of the exact number of clusters in the dataset. However, as the Ew-SRbc method does not provide centroids for the found signal clusters, we have instead tuned the parameters with the SoftBBC-EM algorithm of Gupta and Ghosh (2006). Given that the optimal divergence metric for clustering will be more dependant on the dataset than on the used algorithm, we believe that the parameters detected using SoftBBC-EM will provide, at least, competitive performance when used within Ew-SRbc.

For a given value of γ, the influence of α on the clustering is equivalent to that of m for FCM. When moving from $\alpha \to \infty$ to $\alpha \to 0$, the fuzziness of the clustering is increased from a completely crisp clustering to gradually fuzzier ones. At a certain point α_{th}, the clustering starts degrading, and each object is eventually assigned a uniform probability of belonging to any cluster.

On the flipside, for a given value of α, the influence of γ on the clustering gives rise to two turning points: for values larger than a certain γ_h, the distance between all pairs of objects tend to 2α; whereas for those smaller than a certain γ_l, they all tend to 0. Both phenomena degrade the clustering, and hence also lead to cluster collapse. However, there is an interaction between the values of α and γ: larger values of α force crisper decisions, and hence extend the feasible region for γ.

Hence, the (α, γ) plane will contain an approximately V-shaped curve on one of whose sides the value of the minimum centroid distance will fall below the floating-point-precision threshold ϵ. Following the criterion of Schwämmle and Jensen, we look for maximum fuzziness, and hence the algorithm should select the vertex of this curve. At this point, the value of α is the minimum one which still avoids degradation, and for it γ_h and γ_l have become equal.

We have empirically verified that such curves actually arise across a variety of datasets. For instance, Figure 5.3 shows a contour plot of the minimum centroid distance of the clusterings obtained using SoftBBC-EM on the Toy dataset. In it, the thicker curve denotes the contour level for a value of $\epsilon = 10^{-3}$, and the point at its vertex corresponds to the values of α and γ detected by the algorithm.

Given that the minimum centroid distance function has to be obtained by sampling, which introduces an amount of experimental noise, standard numerical methods for optimization cannot be used, and minimization is instead performed using a recursive logarithmic grid search algorithm. This allows us to exponentially increase the precision in the detection of the optimal point, without an exponential increase of the computational burden.

We will henceforth refer to the distance induced by this automatically tuned Gaussian kernel as G(Auto).

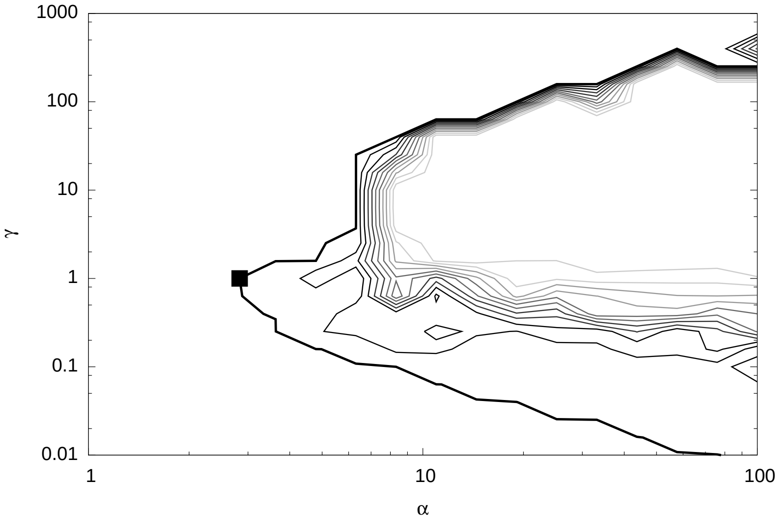


Figure 5.3: Contour plot of minimum centroid distance (Toy data)

5.5 Threshold Determination

The last step of the Ewocs algorithm is that of determining, from the sequence of scores $s^\star_1 \dots s^\star_n$ found by the ensemble clustering process[5], a threshold value $s^\star_{th}$ which separates foreground and background objects. We have considered a number of procedures to perform this decision. Next Sections 5.5.1 to 5.5.4 give descriptions of each one of the considered threshold determination criteria.

5.5.1 Best

A first criterion, which we have named Best, is that of selecting the threshold for which the performance of the method is maximal according to a given measure. The actual applicability of this criterion is limited, as performance measures depend on the availability of the gold truth. However, its output is informative as an upper bound of the performance of the other ones, and we have hence reported it for our experiments.

From the metrics that we have used for our evaluation, we have chosen for our experiments the Best cutoff point to be the one that maximizes the F1 measure, which will be defined in Section 5.6.3.

5.5.2 Size

A more realistic approach than Best is that followed by other works in minority clustering (Gupta and Ghosh, 2005, 2006): the number of foreground objects is assumed to be known a priori. After sorting the objects by their score, it is this number of highest-scored objects that are taken to form the foreground cluster, whereas the rest are considered background objects. The score of the object in the cutoff point is taken as threshold.

However, this threshold criterion, which we will refer to as Size, has the critical drawback of determining the number of foreground objects. Their proposers give no hints about how this quantity could be estimated, and we believe this limits its applicability for unsupervised minority clustering. We have nevertheless included it to allow a comparison to previous approaches which use it.

For our experiments, we have assumed that the exact number of foreground objects is known, and used this value. Hence, the results for Size should also be regarded as an upper bound.

[5] For the sake of simplicity, we will be omitting in this section the R subindex from $s^\star_{Ri}$, as we believe there is no risk of confusion with other than the final scores.

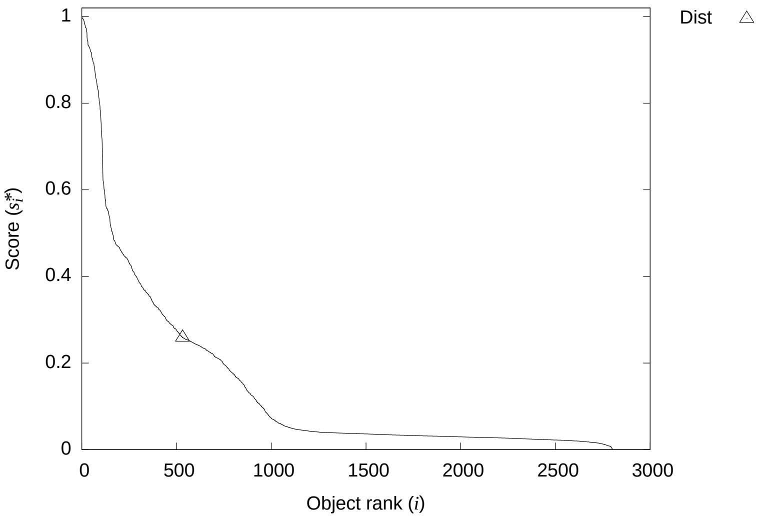


Figure 5.4: Accumulated score distribution (EW-SRBC on TOY data)

5.5.3 DIST

A third approach, more appropriate to unsupervised minority clustering, uses a simple heuristic to determine the threshold value for scores generated by EWOCS, and arises from the observation of the distribution of the sorted sequence of scores of the clustered objects. An example of such distribution appears in Figure 5.4, for a run of EW-SRBC on the TOY data in Figure 5.1.

As observed in the figure, a small number of instances are assigned high scores whereas a large number are assigned low ones, presumably corresponding to foreground and background objects, respectively. The score sequence follows thus the shape of a decreasing convex function. This phenomenon was recurrent across most of the tested datasets.

The cutoff point should try to separate these two regions. Intuitively, this point will lie in the region of maximum convexity of the curve, and hence close to the lower left corner of the plot. This idea leads to the criterion to which we will refer as DIST, and which, as an approximate but efficient way to determine the threshold, minimizes the distance from the origin in a normalized plot of the scores.

The first step in this criterion is hence sorting the objects $x_i \in \mathcal{X}$ by decreasing scores assigned to them by the EWOCS algorithm, so that, in the sequence $s_1^\star \dots s_n^\star$, $\forall i: \; s_i^\star \geq s_{i+1}^\star$. These scores are then linearly mapped to the range $[0 \dots 1]$, obtaining normalized versions $\bar{s}_i^\star$:

$$\bar{s}_i^\star = \frac{s_i^\star - \min s_j^\star}{\max s_j^\star - \min s_j^\star} \tag{5.13}$$

Then, the distance from the origin in the normalized plot is found for each object, and that at the minimum distance is selected as cutoff object x_{th}:

$$\begin{aligned} \mathbf{dist}(x_i) &= \sqrt{(\bar{s}_i^\star)^2 + (i/\max i)^2} && (5.14) \\ x_{th} &= \operatorname*{arg\,min}_{x_i \in \mathcal{X}} \mathbf{dist}(x_i) && (5.15) \end{aligned}$$

This object is the one marked as DIST in Figure 5.4. Its score, $s_{th}^\star$, is the one returned as threshold value.

5.5.4 NGAUSS

The theoretical analysis of the EWOCS method presented in Section 5.3 provides us a new approach to automatically determine the threshold score. In particular, we can much benefit from the result stated in Proposition 5.12: the conditional distributions of the average scores $s_i^\star$ approach a Gaussian distribution with expectation μ_i. If we assume that the value of μ_i depends mainly on the

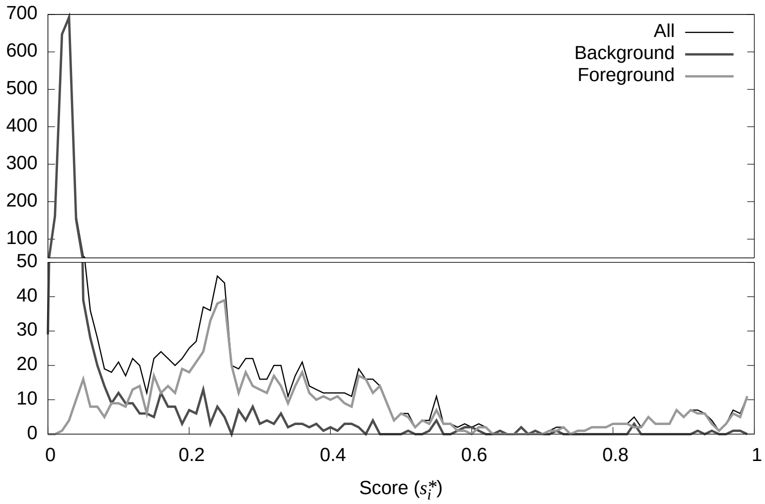


Figure 5.5: Score histogram (Ew-SRbc on Toy data)

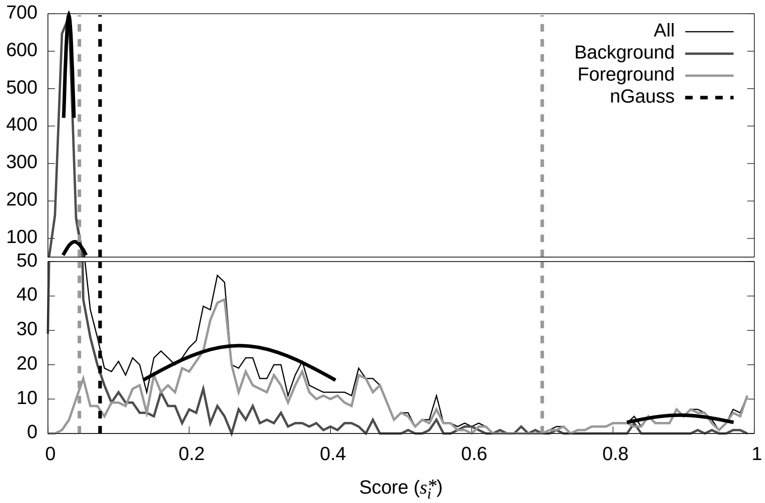


Figure 5.6: Score fitting and threshold detection using nGauss (Ew-SRbc on Toy data)

source ψ_s which produced x_i, we can try to approximate the overall distribution of average scores $s_i^\star$ by a mixture of Gaussian components, one for each one of the sources generating the dataset.

As an example, the histogram of scores generated by the same run of Ew-SRbc on the Toy data is shown in Figure 5.5. As well as the joint distribution of scores (labeled All), the separate histograms for objects from the foreground and background sources are also plotted. Two Gaussian peaks are easily identifiable around the scores of 0.05 and 0.25, and we could expect another minor Gaussian component to explain the probability mass around the score of 0.9.

The key to threshold selection is thus determining the number of mixtures, identifying them, and finding the boundaries between them. The cutoff points must lie at one of these boundaries. There is a wide spectrum of methods to solve this task, and among them we have chosen Expectation-Maximization (EM), being by far the most popular one. The determination of the number of mixtures reduces to discovering the number of clusters and hence to a model selection problem. Given that one-dimensional EM is fast, we have used the usual approach of running EM for increasing numbers of clusters and then using a model-selection criterion to select the best one (Fraley and Raftery, 1998). More specifically, we have used the Bayesian Information Criterion (BIC; Schwartz, 1978).

We will denote the criteria based in this Gaussian-mixture modelling as nGauss. A graphical

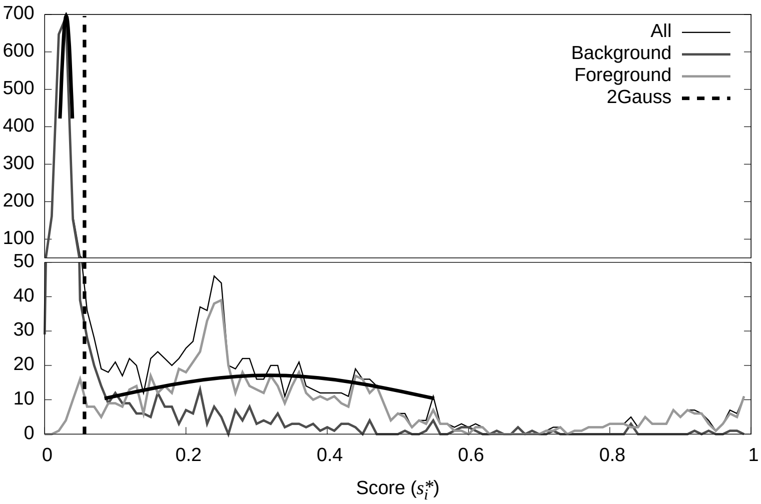


Figure 5.7: Score fitting and threshold detection using 2Gauss (Ew-SRbc on Toy data)

depiction of the modelling of the sample score distribution using Gaussian mixtures is shown in Figure 5.6. In it, the arcs denote the mean, variance and a priori probabilities of the identified components.

Proposition 5.12 states than only the mixture with the lowest mean should contain the background objects. However, it is empirically observed that the selection criterion often chooses models which split this source into several components (this can be observed, for instance, in Figure 5.6). It is hence necessary to separate the found components into those corresponding to the background source and those from the foreground ones. More specifically, if k components $\hat{\psi}_1 \ldots \hat{\psi}_k$ have been identified (sorted by increasing means $\hat{\mu}_1 > \ldots > \hat{\mu}_k$), for each $c \in \{1 \ldots k-1\}$, the possibility that $\hat{\psi}_1 \ldots \hat{\psi}_c$ contain background objects and $\hat{\psi}_{c+1} \ldots \hat{\psi}_k$ contain foreground ones needs to be considered.

The set of cutoff point candidates is hence built from the boundary scores for each $c \in \{1 \ldots k-1\}$, i.e., the scores $s_c^\star$ for which[6]

$$p(s_c^\star \in \hat{\psi}_1 \vee \ldots \vee s_c^\star \in \hat{\psi}_c \mid s_c^\star) = p(s_c^\star \in \hat{\psi}_{c+1} \vee \ldots \vee s_c^\star \in \hat{\psi}_k \mid s_c^\star)$$

Moreover, and as stated in Section 5.5.3, the small number of foreground instances are assigned high scores whereas the large number of background instances are assigned low scores. As a result, the variances of the scores of the former will differ significantly from those of the latter, being much larger.

This last fact provides us with a heuristic criterion to choose a single threshold score from the candidate set: being $\hat{\sigma}_1^2 \ldots \hat{\sigma}_k^2$ the variances of the found components $\hat{\psi}_1 \ldots \hat{\psi}_k$, we select the boundary score that maximizes the difference between the average component variances at both sides:

$$s_{th}^\star = \arg\max_{s_c^\star} \left| \frac{1}{c} \sum_{i=1}^{c} \hat{\sigma}_i^2 - \frac{1}{k-c} \sum_{i=c+1}^{k} \hat{\sigma}_i^2 \right|$$

We will refer to this criterion as nGauss+Var. As an upper bound of its performance, we will also consider a nGauss+Best criterion, which selects the boundary score $s_c^\star$ which maximizes the F1 measure. In Figure 5.6, the possible cutoff points are depicted by dashed vertical rules. The score selected as threshold by both nGauss+Best and nGauss+Var is emphasized in black.

A slightly different alternative to overcome the foreground and background component separation problem is that of simplifying the possible models and performing EM with only 2 clusters. In this case, there is no ambiguity in the choice of the background and foreground components, as there must be one of each. We have named this simplified Gaussian modeling approach 2Gauss. Figure 5.7 contains a representation of the modelling of the sample scores using 2Gauss.

[6] If several such scores exist for a given c, we have taken the largest value for which, in addition, the probability of the foreground mixtures is increasing.

Number of dimensions	2, 3, 5, 8
Data range	$[-2.0 \ldots +2.0]$
Number of background samples	5400 ... 12000
Number of foreground sources	3 ... 8
Number of foreground samples	700 ... 1800
Variance within foreground sources	0.125 ... 0.25
Minimum distance between foreground sources	0.75

Table 5.1: Parameter range for synthetic dataset generation

Finally, as a last and implementation-related detail, we have found that using the linearly mapped scores $\bar{s}_i^\star$ as defined in Equation 5.13 as input to the EM algorithm for model fitting, instead of the actual scores $s_i^\star$, reduces the floating point rounding error and improves the quality of the detected threshold.

5.6 Evaluation

In order to validate the proposed Ewocs algorithm and to assess the performance of Ewocs-based approaches, we have performed a series of experiments on synthetic data. In a preliminary stage, the consistency of the different used weak clustering algorithms has been empirically assessed. Later, a full-fledged comparison of the performance of Ewocs-based approaches to other methods in the state of the art has been carried out.

Next sections give details about the evaluation procedure. Section 5.6.1 describes the used datasets and Section 5.6.2 enumerates the different approaches to be evaluated or employed as reference. Next Section 5.6.3 describes the evaluation protocol, including the considered metrics, and, finally, Section 5.6.4 exposes and discusses the obtained results.

5.6.1 Data

The first dataset we have used for our experiments is the sample data plotted in Figure 5.1. It is a simple 2-dimensional dataset in which five foreground sources, with different shapes and variances, are scattered against a background filled with a uniform distribution. Even though evaluation on a single dataset such as Toy scarcely possesses any statistical significance, "*for a 2-dimensional dataset, graphical verification is an intuitive and reliable validation of clustering*" (Ando, 2007), and we believe this can be useful as an illustration of most of the concepts in our work.

For a more serious evaluation, we have prepared a number of synthetic datasets where foreground Gaussian sources are embedded within a set of uniformly distributed background objects. Several parameters, such as the number of sources, the number of foreground and background objects and the means and variances of the Gaussian sources, were chosen at random for each dataset. A summary of the ranges of these parameters can be found in Table 5.1. In total, 160 such datasets have been generated. We will refer to this collection as Synth.

Additionally, in order to perform the preliminary experiments on method consistency, for each dataset in Synth, 9 additional samplings using the same source parameters were generated. The whole 10-dataset groups have been used for consistency estimation.

5.6.2 Approaches

We have implemented the Ewocs algorithm using each one of the weak clusterers proposed in Section 5.4.

Ew-RSplit Ewocs using the RSplit algorithm of Section 5.4.1.

Ew-HRbc Ewocs using the hard Rbc algorithm, HRbc, of Section 5.4.2.

Ew-SRbc Ewocs using the soft Rbc algorithm, SRbc, of Section 5.4.2.

The notation Ew-RSplit/R×k (resp., Ew-HRbc/R×k and Ew-SRbc/R×k) will be used to refer to the results obtained by Ewocs with an ensemble of R clusterings, each one produced by RSplit (resp., HRbc and SRbc) with $k_{max} = k$.

In order to assess the performance of Ewocs-based approaches with respect to the state of the art, we have implemented three existing methods for minority clustering:

BBOCC as proposed by Gupta and Ghosh (2005). We have used Mah as divergence, and the actual number of foreground objects as the *desired clustering size* parameter.

BBCPress as proposed by Gupta and Ghosh (2006). Similarly to BBOCC, we have used Mah as divergence, and the actual number of foreground objects as the *desired clustering size* parameter. The number of clusters, however, has been assumed to be given a priori, and by BBCPress/k we will refer to the runs of this algorithm with a number of clusters k.

***k*MD** as proposed by Ando (2007). The implementation tries to mimic to the maximum extent that of the original paper: we have used Gaussian distributions for the foreground clusters, and a uniform distribution for the background cluster. The clusters have been initialized by selecting fixed-size sets of most similar points to a randomly chosen one. To refer to the runs of this algorithm with a certain parameter tuning, we will use the notation kMD/R×s_0–s_{min}, where R refers the number of cluster detection iterations, and s_0 and s_{min} refer to the *initial* and *required cluster size* parameters.

It is important to note that these methods, as well as, to our knowledge, all other existing minority clustering methods proposed so far, include critical elements of supervision, in the form of parameters such as the number of foreground objects, the number of foreground clusters, or the foreground cluster sizes.

Additionally, we have considered three pseudo-systems for reference, to give lower and upper bounds of the performance of the actual systems:

Random A random clusterer, which assigns foreground and background clusters according to a Bernoulli distribution. We have taken the one among such clusterers which assigns the labels according to the actual source size ratio in the data.

AllFG A blind clusterer, which assigns all objects to the foreground cluster.

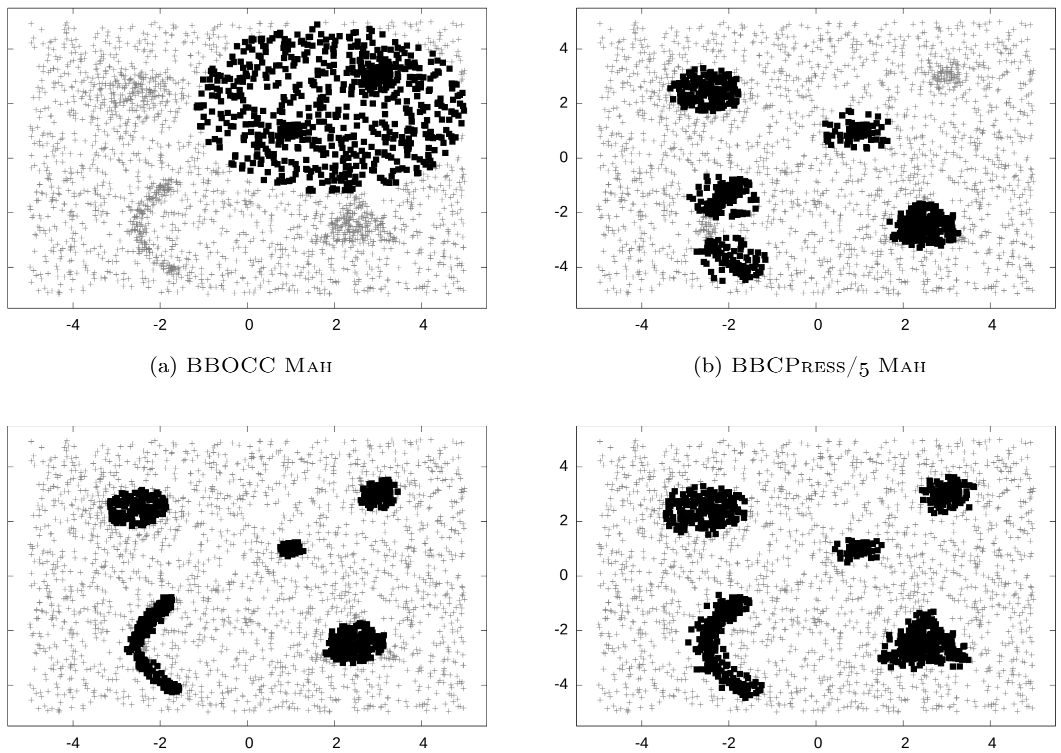
Convex An oracle clusterer for the Synth dataset, which detects as foreground objects those objects that lie within the convex hull of the actual foreground sources. As the foreground objects are embedded within the background ones, there is a number of the latter that fall within the region of the former and are hence, from the point of view of Gaussian distributions, indistinguishable from them. The output of this Convex clusterer will hence detect all foreground objects, but include some background ones (those in the union of each source's convex hull) as false positives.

5.6.3 Protocol

In the preliminary evaluation of clustering consistency, for each one of the 10 samplings of the datasets in the Synth collection, 25 runs of every weak clustering algorithm were performed, and the source affinities have been estimated from the co-occurrence matrices of these 250 clusterings. We have then reported the fraction of datasets with which the considered methods are consistent (Cons), as well as, more precisely, the fraction of sources which are detectable by them—both macro- (M-Det) and micro-averaged (μ-Det) by dataset.

On the flipside, in order to assess and compare the performance of the different approaches in the full minority clustering evaluation, we have used the same set of metrics as in our previous relation detection evaluation: namely, the well-known measures of precision (Prc), recall (Rec) and F1 (see Section 4.5.2). These metrics constitute a de facto standard for unsupervised classification tasks, and, in particular, have been previously employed for the evaluation of minority clustering (Ando, 2007). As in previous chapters, the use of percentages when printing values of these metrics is customary.

Additionally, to evaluate the performance of the scoring phase, isolating it from that of threshold selection, we have also included Area Under the ROC Curve (AUC) measures. To reduce the impact



(a) BBOCC Mah

(b) BBCPress/5 Mah

(c) kMD/100×800–50

(d) Ew-SRbc/100×100 + nGauss G(10,10)

Figure 5.8: System output for the compared methods (Toy data)

of randomness, we have carried out 5 different runs for each method, configuration and dataset, and reported the average measures.

Finally, as in previous chapter, statistical significance of the results is assessed using Bergmann and Hommel hypothesis tests, with their output represented in the style of Demšar.

5.6.4 Results

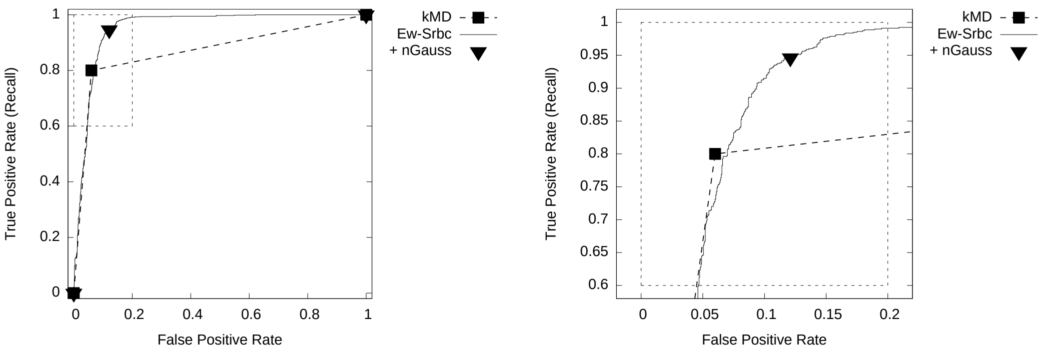
The first Section 5.6.4.1 presents the results of the full experiments on the Toy dataset. The next two sections, 5.6.4.2 and 5.6.4.3, detail the results obtained over the Synth collection—the former regarding the preliminary experiments on clustering consistency; the latter, those on the full minority clustering task.

5.6.4.1 Clustering on the Toy Dataset

A graphical depiction of the output of a representative subset of the compared approaches on the Toy dataset is shown in Figure 5.8. Even if these results are only given for illustrative purposes, a number of phenomena can already be observed in the different obtained clusterings.

For instance, we can see how the BBOCC method is unable to detect the multiple foreground sources and instead creates a single cluster covering two of them. Also, the BBCPress method, despite being given the correct number of sources, fails to recognize the half-moon-shaped one and instead splits it into two clusters, and rounds the triangle-shaped one. As a result, the top right source to be missed. The limitations of these two methods are well-known, and come from the fixed number and shape (hyperelliptical) of clusters they look for.

On the flipside, the kMD and Ew-SRbc methods are able to recognize the variously shaped foreground sources. It is interesting to note that, for this Toy dataset, kMD produces tighter clusters, favouring precision over recall, whereas for Ew-SRbc this tendency is reversed. The more systematical evaluation on Synth, presented in next section 5.6.4.3, will confirm this behaviour.



(a) Overall (b) Detail

Figure 5.9: ROC curves for *k*MD and Ew-SRbc (Toy data)

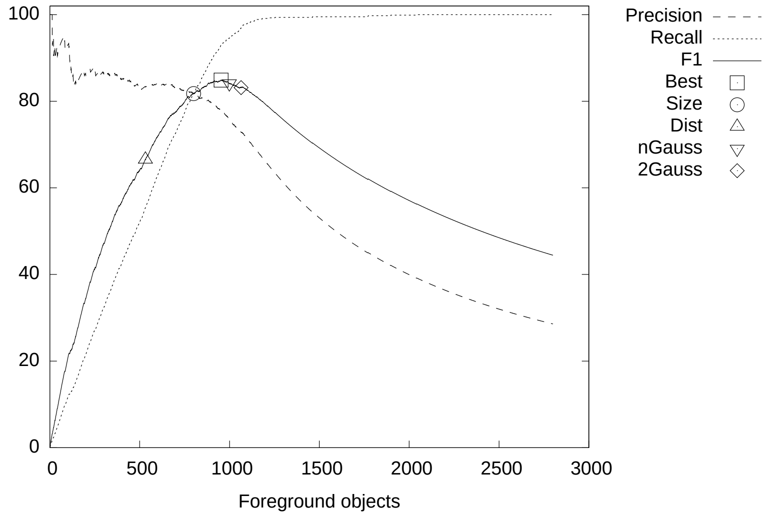


Figure 5.10: Precision, Recall and F1 curves, and cutoff point determined by different threshold detection criteria (Ew-SRbc on Toy data)

It is interesting to compare the ROC curves for both approaches, which are plotted in Figure 5.9. *k*MD does not provide an adjustable decision threshold; instead, its output is a fixed crisp boundary, and hence its ROC curve is composed of two straight segments. On the contrary, Ew-SRbc, as all other Ewocs-based approaches, assigns a continuous score to all objects, and the separation between foreground and background ones is based on a threshold. Hence, its ROC curve, as a function of this threshold, is much smoother. For this reason, even if the differences in precision, recall and F1 score between the two methods are small (see Subfigure 5.9b), the curve for *k*MD is in this case missing a large fraction of the AUC, which that of Ew-SRbc is able to enclose. The fact will also be relevant to the evaluation on Synth.

Regarding the proposed threshold determination approaches, Figure 5.10 shows the precision, recall and F1 curves for the output of Ew-SRbc on Toy, according to the number of objects clustered as foreground. The cutoff points for the different criteria are plotted above the F1 curve. For this particular case, nGauss+Var finds the same cutoff point as nGauss+Best, and they are both plotted as nGauss.

It is interesting to observe how the curve for precision remains quite high up to the break-even point with recall, capturing the foreground objects and those background objects which lie among them (and which prevent the value from reaching 100%), and then start to decrease as all foreground objects have been detected. Regarding recall, it increases regularly until it reaches almost 100%, only slightly after the best F1 point. We believe both behaviours are indicators of a good quality of the minority clustering process.

			2 Dimensions			3 Dimensions		
			Cons	M-Det	μ-Det	Cons	M-Det	μ-Det
RSplit	×2	-	81.82	96.10	94.48	100.00	100.00	100.00
	×50	-	78.79	95.82	95.71	100.00	100.00	100.00
HRbc	×100	Mah	93.94	99.13	98.77	100.00	100.00	100.00
		G(∞,2)	93.94	99.13	98.77	100.00	100.00	100.00
SRbc	×100	Mah	84.85	94.81	95.71	100.00	100.00	100.00
		G(10,10)	100.00	100.00	100.00	100.00	100.00	100.00

			5 Dimensions			8 Dimensions		
			Cons	M-Det	μ-Det	Cons	M-Det	μ-Det
RSplit	×2	-	100.00	100.00	100.00	100.00	100.00	100.00
	×50	-	100.00	100.00	100.00	100.00	100.00	100.00
HRbc	×100	Mah	100.00	100.00	100.00	100.00	100.00	100.00
		G(∞,2)	100.00	100.00	100.00	100.00	100.00	100.00
SRbc	×100	Mah	100.00	100.00	100.00	100.00	100.00	100.00
		G(10,10)	100.00	100.00	100.00	100.00	100.00	100.00

Table 5.2: Consistency of the proposed weak clustering algorithms (Synth data)

Additionally, the plot shows how the threshold values found by Size, nGauss and 2Gauss are quite close to the optimal one, Best. It is only the threshold found by Dist which falls somehow behind, trading in this case too much recall for precision. Again, to asses the actual relationship between the power of the different criteria, more systematical testing, such as the one on Synth, is required.

5.6.4.2 Consistency on the Synth Dataset Collection

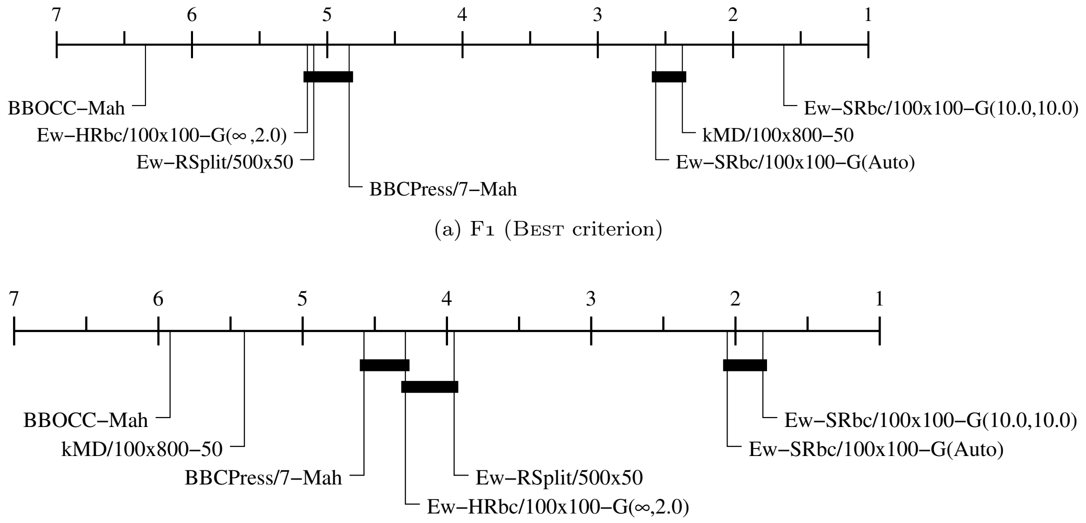
Table 5.2 contains the values of consistency and averaged source detectability of the different weak clustering algorithms, estimated over all Synth datasets. Given that more dimensional data will exhibit a larger degree of sparsity which may render the results not comparable with those of lower dimensional datasets, we have opted to present the results segregated by the number of dimensions in the datasets.

As seen in the table, our hypothesis that weak clustering algorithms are consistent with data generated by dense and local sources seems clearly corroborated by the empirical evidence coming from these experiments. We have found the property to hold in *all* tested datasets for 3-, 5- and 8-dimensional data. Only for 2-dimensional datasets, the algorithms, especially RSplit and SRbc using the Mah distance, fail to detect some of the sources—up to 7.45% of them in the case of SRbc with Mah. Overall, for these two methods full consistency is only achieved in three fourths of the datasets; and HRbc fulfills the property in 91.67% of the cases. On the flipside, the performance of HRbc using G(10,10) is remarkable, as it obtains perfect consistency even in these harder cases. The results also confirm the intuition that 2-dimensional datasets, being less sparse, are harder to deal with.

However, even if perfect consistency is not achieved, the fact that, in the worst of the cases, more than 94% of the sources are detectable suggests that the lack of full consistency does not necessarily hamper the actual performance of the Ewocs algorithm. The study of the clustering results over the same Synth collection in next section will shed light on this issue.

5.6.4.3 Clustering on the Synth Dataset Collection

Table 5.3 contains the AUC values for the compared methods across all datasets in the Synth collection, as well as their achievable precision, recall and F1 values, using the Best threshold selection criterion. As mentioned before, the degree of sparsity increases with the number of dimensions, and this simplifies the clustering task, and the results across datasets with different dimensionality



(a) F1 (Best criterion)

(b) AUC

Figure 5.11: Bergmann-Hommel tests for the compared approaches (Synth data)

may not be commensurable. For this reason, we have again opted to split the results according to dataset dimensionality.

For reasons of brevity, only the configurations which achieve the best results for each method are included. Later in this same section, experiments studying the sensitivity of each method to the tuning of their parameters will be presented.

Finally, regarding statistical significance, Figure 5.11 contains a graphical representation of the outcome of Bergmann-Hommel tests on the F1 and AUC measures across all datasets in Synth.

One of the conclusions that can be drawn after inspection of the figures in the table is that Ewocs-based approaches are able to obtain results in the state of the art for minority clustering, and that, particularly, Ew-SRbc is able to outperform the existing approaches for the task, achieving a performance close to the upper bound, given by Convex. We believe this is an excellent result, and one which confirms the validity of the Ewocs algorithm.

A number of additional conclusions can follow from a more detailed analysis. First of all, it is clear the BBOCC is the weakest approach among the compared ones. Even if its results are above the Random and AllFG baselines, the limitation to a single hyperelliptical cluster produces clusterings with a lower precision than those from other approaches. The differences are statistically significant in terms of both F1 and AUC.

Regarding Ew-RSplit, it is interesting to note that the extension from 2 to a larger number of hyperplanes improves the performance of the RSplit algorithm within the ensemble. However, the algorithm favours too much recall over precision, and even if this allows it to achieve a good AUC measure, its values of F1 are lower than other methods which exhibit a similar performance, such as Ew-HRbc and BBCPress. These too approaches trade some of the recall of Ew-RSplit for precision, thus obtaining lower AUC but higher F1. The differences between the three systems, nevertheless, are deemed not significant by the Bergmann-Hommel test, except for the one in terms of AUC between Ew-RSplit and BBCPress, and can hence be considered similar in terms of minority clustering power.

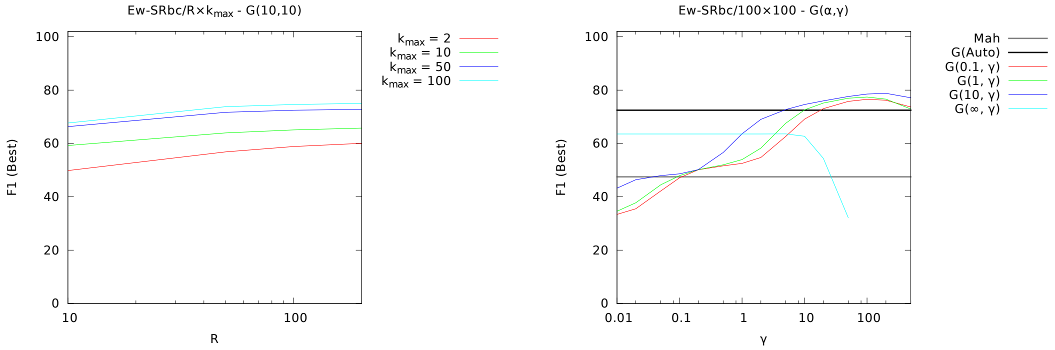
It can also be observed how the results for Ew-HRbc are similar using either of the Mah or G(∞,2) distances.

Finally, concerning kMD and Ew-SRbc, the results from the experiments allow us to say that their performance is significantly better than that of the other methods in terms of F1, and that of Ew-SRbc is also better in terms of AUC. This is true for Ew-SRbc not only when using the G(10,10) distance, which achieved the best results on Synth, but also when using the unsupervised one G(Auto). More precisely, the results for Ew-SRbc using G(Auto) are only slightly below those of kMD in terms of F1, and slightly below those obtained using G(10,10) in terms of AUC. In both cases the differences are not statistically significant. Taking into account that the determination of G(Auto) is completely unsupervised, we believe we can qualify these

			2 Dimensions				3 Dimensions			
				Best				Best		
			AUC	Prc	Rec	F1	AUC	Prc	Rec	F1
Random	-		**0.500**	14.5	14.5	**14.5**	**0.500**	14.5	14.5	**14.5**
AllFG	-		**0.500**	14.5	100.0	**24.9**	**0.500**	14.5	100.0	**24.9**
BBOCC	-	Mah	**0.752**	40.9	69.4	**44.5**	**0.841**	61.6	62.3	**56.9**
BBCPress	7	Mah	**0.849**	55.6	68.1	**60.7**	**0.934**	79.0	76.8	**77.4**
kMD	100×800–50		**0.808**	82.0	63.7	**68.5**	**0.945**	93.6	90.0	**91.6**
Ew-RSplit	500×2	-	**0.843**	41.1	78.0	**52.1**	**0.911**	55.9	77.2	**63.6**
	500×50	-	**0.862**	45.0	75.9	**54.5**	**0.950**	66.0	83.9	**73.1**
Ew-HRbc	100×100	Mah	**0.896**	59.1	71.5	**63.6**	**0.971**	76.1	85.0	**79.9**
		G(∞,2)	**0.896**	58.9	71.7	**63.5**	**0.971**	76.3	84.9	**79.9**
Ew-SRbc	100x100	Mah	**0.799**	37.1	73.3	**47.5**	**0.901**	53.6	78.2	**62.5**
		G(10,10)	**0.958**	66.4	85.9	**74.6**	**0.991**	85.3	94.9	**89.7**
		G(Auto)	**0.937**	64.5	83.7	**72.5**	**0.986**	83.7	93.2	**88.1**
Convex	-		**0.957**	67.6	100.0	**79.3**	**0.996**	95.4	100.0	**97.6**

			5 Dimensions				8 Dimensions			
				Best				Best		
			AUC	Prc	Rec	F1	AUC	Prc	Rec	F1
Random	-		**0.500**	14.5	14.5	**14.5**	**0.500**	14.5	14.5	**14.5**
AllFG	-		**0.500**	14.5	100.0	**24.9**	**0.500**	14.5	100.0	**24.9**
BBOCC	-	Mah	**0.942**	87.4	75.7	**79.9**	**0.993**	94.5	93.8	**94.0**
BBCPress	7	Mah	**0.983**	92.2	89.2	**90.2**	**0.996**	96.3	97.0	**96.6**
kMD	100×800–50		**0.983**	98.9	96.7	**97.8**	**0.991**	99.8	98.1	**99.0**
Ew-RSplit	500×2	-	**0.961**	77.2	84.7	**80.1**	**0.991**	91.8	91.3	**91.3**
	500×50	-	**0.986**	87.4	91.2	**89.0**	**0.998**	96.0	97.6	**96.8**
Ew-HRbc	100×100	Mah	**0.985**	86.7	90.4	**88.3**	**0.993**	91.6	94.2	**92.7**
		G(∞,2)	**0.984**	86.8	90.2	**88.2**	**0.991**	91.2	93.3	**91.9**
Ew-SRbc	100x100	Mah	**0.966**	79.9	85.9	**82.1**	**0.992**	91.2	94.2	**92.3**
		G(10,10)	**0.999**	98.3	99.4	**98.8**	**0.996**	99.9	99.4	**99.6**
		G(Auto)	**0.972**	90.4	96.5	**91.4**	**0.987**	96.1	99.7	**96.8**
Convex	-		**1.000**	100.0	100.0	**100.0**	**1.000**	100.0	100.0	**100.0**

Table 5.3: Results for Synth data



(a) Effect of R and k_{max} on F1

(b) Effect of α and γ on F1

Figure 5.12: Effect of parameters on Ew-SRbc (2-dimensional Synth data)

results as really encouraging.

However, the results using the Mah distance within Ew-SRbc fall much below those obtained with the G(α,γ) family. One reason for this behaviour may lie in the fixed degree of fuzziness allowed by Mah: the standardized scale that this distance provides may not always give the most suitable fuzziness for use in soft Bregman clustering in general, and Ew-SRbc in particular. The greater versatility offered by the G(α,γ) distances is thus a valuable property.

It is important to note that the high F1 score of kMD comes from its elevate precision, which is particularly high, for instance, in 3-dimensional datasets; whereas Ew-SRbc tends to favour recall over precision. Also, we can see how the value of AUC for kMD is lower than for all other methods except BBOCC. Both phenomena were also observed in the Toy dataset, and the explanation for the latter is, as already mentioned in Section 5.6.4.1, the lack of an adjustable threshold in the output of kMD.

At the light of these results, we believe we can assert that Ewocs-based approaches perform competitively with respect to the state of the art in the minority clustering task, in terms of AUC and F1 of the obtained clusterings. Ensemble clustering methods have hence been proven to be useful for this task.

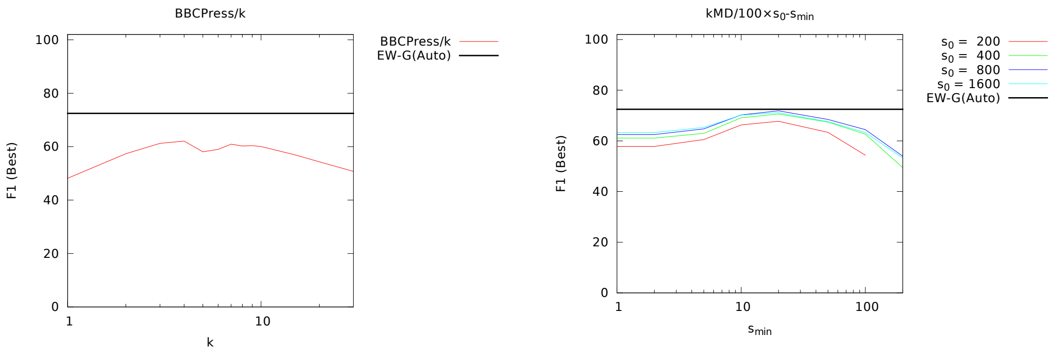
Moreover, the fact that the Ew-SRbc method is able to outperform all other compared approaches when using the manually tuned Gaussian-kernel distance, and most of them when using the automatically tuned one, leads us to believe that, on the one hand, kernel-based distances are a serious alternative to other similarity measures used in clustering tasks; and that, on the other, the proposed Rbc algorithm can be successfully employed to construct individual clusterings suitable for combination within a clustering ensemble.

However, these conclusions require an evaluation of the sensibility to parameter tuning of the compared approaches. A detailed study of this issue follows below.

Parameter Sensitivity A number of experiments have been performed to assess the relevance of parameter tuning on the different approaches, in terms of the impact these parameters have in their performance on the minority clustering task.

Figure 5.12 provides two plots of the Best F1 score as a function of the parameters in Ew-SRbc: the ensemble size R, the maximum number of clusters in each individual clustering k_{max}, and the Gaussian-kernel distance scaling factors α and γ. The plots correspond to the 2-dimensional subset of the Synth collection, being the datasets where the difference in performance between approaches is the largest.

First Subfigure 5.12a plots the curves of F1 for a fixed distance G(10,10). It can be seen how a change in any of the two parameters does influence the F1 score. However, the difference in performance is small, and, more importantly, the value stabilizes with increasing values of both R and k_{max}. The curves for $k_{max} = 50$ and $k_{max} = 100$ are only shifted a small offset from each other, and all curves present an almost flat slope after $R = 50$. Hence, we believe the tuning of these two parameters is not an issue of clustering performance but rather of computational burden, and that



(a) Effect of k on F1 of BBCPress

(b) Effect of s_0 and s_{min} on F1 of kMD

Figure 5.13: Effect of parameters on BBCPress and kMD (2-dimensional Synth data)

a value such as that of $R = k_{max} = 100$, which we have used in our experiments, will produce good quality clusterings across a wide range of situations.

However, the plot in Subfigure 5.12b, which shows the curves of F1 for fixed values of $R = k_{max} = 100$, presents a different picture. The scaling parameters of the Gaussian-kernel distance also have an impact on the F1 of the clusterings produced by Ew-SRbc, but in this case the values do not stabilize: the curves for increasing values of α obtain higher F1 values, but nevertheless the limit when $\alpha \to \infty$ presents a completely different behaviour. Moreover, the curves for finite α present a maximum around $\gamma = 10$, and lower values of F1 are obtained at either side of these maxima. Hence, the score using G(α,γ) distances can exceed significantly that obtained using Mah, but it can also eventually drop much below.

It is hence clear that, as intuited in Section 5.4.2.3, the selection of the suitable values for α and γ is a crucial issue when using Ew-SRbc. Nevertheless, the plot in Subfigure 5.12b also shows how the value of F1 obtained using the automatically tuned G(Auto) distance provides an approximation to the optimum. Even if there does exist a gap between the highest achievable results and those obtained using the unsupervised distance, previous Table 5.3 and Figure 5.11 have already shown how this does not prevent them to be higher than those of most other approaches. We hence believe that G(Auto) can be used to perform the minority clustering task satisfactorily, even if we must also admit that fine tuning can improve the overall results.

A different behaviour is also observed for the subfamily G(∞,γ), which corresponds to the Ew-HRbc method. For this particular case, the curve stabilizes for low values of γ, and hence one may obtain a simple rule of thumb to tune the distance for this method. Nevertheless, the value of F1 is lower than that achievable using Ew-SRbc, and also lower than that obtained using G(Auto).

Regarding non-Ewocs-based approaches, Figure 5.13 contains plots of the Best F1 score as a function of the number of clusters k of BBCPress, and as a function of the starting s_0 and minimum required s_{min} cluster sizes of kMD, for a fixed number of detection iterations $R = 100$. For reference, the plots also include the value obtained by Ew-SRbc/100×100 using G(Auto)[7].

Subfigure 5.13a shows the impact of parameter k, the number of clusters, on the F1 score of BBCPress. In addition to the fact that the maximum achievable value is still below that obtained by Ew-SRbc, this maximum happens only for a precise value of k, at whose sides the scores fall behind. Given that Gupta and Ghosh (2006) do not provide any hint about the determination of a suitable value for k (and use instead for their evaluation the actual number classes in the dataset), we believe the tuning of this parameter can become a problematic issue when using the BBCPress method, and remains one more of its drawbacks.

Similarly, Subfigure 5.13b contains the F1 curves obtained by kMD for different values of the starting s_0 and minimum required s_{min} cluster sizes. In this case, the method seems robust to the choice of s_0, but presents a maximum for one single value of s_{max}, slightly below the results obtained by Ew-SRbc. Again, the results drop at either side of the maximum, and hence careful parameter tuning may be required. However, Ando (2007) does not explain how this may be achieved, and the issue, once more, remains one which may become problematic.

[7] For space reasons, the name is shown shortened as EW-G(Auto).

			Best			Size			Dist		
			Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
Random	-		14.5	14.5	**14.5**	-	-	**-**	-	-	**-**
AllFG	-		14.5	100.0	**24.9**	-	-	**-**	-	-	**-**
BBOCC	-	Mah	40.9	69.4	**44.5**	35.4	35.4	**35.4**	-	-	**-**
BBCPress	7	Mah	55.6	68.1	**60.7**	58.3	58.3	**58.3**	-	-	**-**
kMD	100×800–50		82.0	63.7	**68.5**	-	-	**-**	-	-	**-**
Ew-RSplit	500×2	-	41.1	78.0	**52.1**	41.2	41.2	**41.2**	29.1	85.9	**42.2**
	500×50	-	45.0	75.9	**54.5**	48.0	48.0	**48.0**	30.2	87.9	**43.8**
Ew-HRbc	100×100	Mah	59.1	71.5	**63.6**	61.2	61.2	**61.2**	37.8	86.3	**51.0**
		G(∞,2)	58.9	71.7	**63.5**	61.2	61.3	**61.2**	37.9	86.4	**51.1**
Ew-SRbc	100x100	Mah	37.1	73.3	**47.5**	38.1	38.1	**38.1**	25.0	82.6	**37.2**
		G(10,10)	66.4	85.9	**74.6**	71.2	71.2	**71.2**	45.2	97.4	**60.6**
		G(Auto)	64.5	83.7	**72.5**	68.7	68.7	**68.7**	56.6	81.8	**63.4**
Convex	-		67.6	100.0	**79.3**	-	-	**-**	-	-	**-**

			nGauss+Best			nGauss+Var			2Gauss		
			Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
Ew-RSplit	500×2	-	39.5	77.7	**50.3**	39.2	58.7	**35.9**	34.2	81.2	**45.8**
	500×50	-	42.0	78.5	**51.8**	25.6	89.3	**32.2**	35.7	85.2	**46.8**
Ew-HRbc	100×100	Mah	56.0	72.2	**60.5**	41.2	80.0	**45.6**	38.5	89.9	**48.7**
		G(∞,2)	56.6	72.0	**60.4**	39.6	81.6	**45.5**	38.4	89.7	**48.5**
Ew-SRbc	100x100	Mah	35.7	74.1	**46.3**	24.3	88.8	**31.7**	32.8	76.1	**43.1**
		G(10,10)	62.0	88.0	**70.9**	50.3	94.2	**64.2**	49.3	96.4	**64.6**
		G(Auto)	58.5	87.2	**68.7**	63.7	60.1	**51.4**	51.6	90.2	**63.4**

Table 5.4: Results for 2-dimensional Synth data

From the outcome of our experiments, we believe that, as usual in clustering tasks, parameter tuning is crucial to the performance of the compared minority clustering methods. In particular, the number of clusters k of BBCPress, the minimum required cluster size s_{min} of kMD, and the α and γ Gaussian-kernel scaling parameters used within Ew-SRbc may have a dramatic impact of the F1 scores of the produced clusterings.

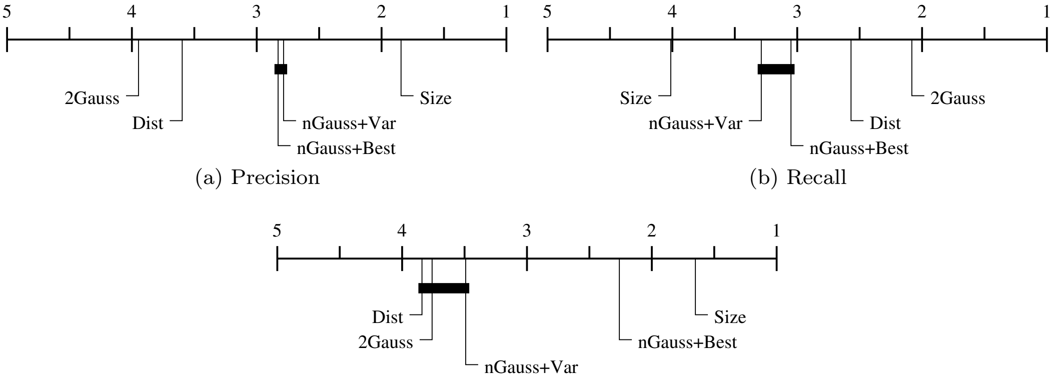
However, whereas their respective authors proposed no method to automate (or even guide) the tuning of the parameters of BBCPress or kMD, the automatic tuning procedure of the proposed G(Auto) distance has empirically been proved to be a useful tool for Ew-SRbc, and one which provides a solution to the tuning of the scaling parameters of a Gaussian-kernel distance in the task of minority clustering. Moreover, the results obtained using Ew-SRbc and G(Auto) are better than those of the compared approaches in terms of AUC and F1.

We believe the existence of such a tool is a significant difference with respect to other approaches, and that this makes Ew-SRbc suitable for completely unsupervised minority clustering tasks.

Threshold determination The last of the aspects in the Ewocs-based approaches which remains open to study is that of the criteria for threshold determination. Table 5.4 contains the values of precision, recall and F1 obtained when applying the different criteria to the output of each minority clustering method. Again, for brevity the table contains only the results across the 2-dimensional datasets of Synth.

Concerning the statistical significance of the differences, Figure 5.14 contains the graphical representation of the outcome of Bergmann-Hommel tests on precision, recall and F1 across all (not only 2-dimensional) datasets in Synth.

There is a number of trends that can be observed in the results. First of all, there is still a gap



(a) Precision

(b) Recall

(c) F1

Figure 5.14: Bergmann-Hommel tests for the compared criteria (Ew-SRbc/100×100 using G(10,10) on Synth data)

between the maximum achievable F1 score (criterion Best) and that obtained using the different criteria. Second, there is another gap between the F1 scores of the criteria that contain some element of supervision (Size and nGauss+Best) and those of the completely unsupervised ones (Dist, nGauss+Var and 2Gauss). These differences are present in a consistent way across all the Ewocs-based approaches.

Criterion Size is hence the one to obtain results closest to Best in terms of F1. It is also the one to obtain the best figures for precision, but at the cost of being the one which gives the least recall. All differences are statistically significant.

However, the upper bound achievable using Gaussian modelling of the scores, that of nGauss+Best, lies quite close to the output of Size. For the Ew-SRbc/100×100 method using G(10,10) on 2-dimensional data, the difference is only a 0.3% in terms of F1. nGauss+Best also shifts the bias towards recall instead of precision, which is much closer to the region where the optimal threshold (that of Best) lies.

Finally, regarding the three unsupervised criteria, nGauss+Var seems the one which comes closest in terms of performance to nGauss+Best. Even if this does not hold for the particular subset of 2-dimensional data, overall nGauss+Var gives higher precision and lower recall than nGauss+Best. These differences are not statistically significant, but overall the one in F1 score is.

The remaining criteria Dist and 2Gauss show a strong bias for recall, particularly the latter, and fall much below nGauss+Best in precision. They perform worse in terms of F1 than the other proposed approaches. However, from the statistical point of view, the difference is not significant between them and nGauss+Var.

Taking these and all obtained results into account, we believe we can affirm that, even if elements of supervision improve the results in the task of minority clustering, the proposed Ewocs algorithm allows us to obtain competitive results using an unsupervised[8] approach: the results obtained by Ew-SRbc/100×100 using G(Auto) and one of Dist, nGauss+Var or 2Gauss are above those obtained by other supervised approaches, such as BBOCC or BBCPress.

Regarding the elements of supervision introduced by each one of the criteria, it is remarkable that the use of nGauss+Best, which would require an a posteriori selection of the number of background Gaussian components from a small number of them, suffices for Ew-SRbc/100×100 using G(Auto) to outperform all other approaches, including kMD, which requires careful tuning of three parameters R, s_0, s_{min}.

Even if it is true that manual determination of the most suitable G(α,γ) distance, or more informed (i.e., supervised) threshold detection criteria, such as Size or Best, allow further increases in the F1 scores obtained by Ew-SRbc, we believe that the fact that, using no or little supervision, Ew-SRbc outperforms supervised minority clustering approaches in the state of the art is an

[8] As observed in the previous section, the Ew-SRbc method is robust to the tuning of R and k_{max}, so we can consider it unsupervised.

excellent result, and one which proves the validity of the whole minority clustering framework introduced by the EWOCS algorithm.

5.7 Conclusions

In this chapter, we have considered the problem of minority clustering, contrasting it with regular all-in clustering. We have identified a key limitation of existing minority clustering algorithms—namely, we have seen how the approaches proposed so far for minority clustering are supervised, in the sense that they require the number and distribution of the foreground clusters, as well as the background distribution, as input.

The fact that, in supervised learning and all-in clustering tasks, combination methods have been successfully applied to obtain distribution-free learners, even from the output of weak individual algorithms, has led us to make a three-fold proposal.

First, we have presented a novel ensemble minority clustering algorithm, EWOCS, suitable for weak clustering combination. The properties of EWOCS have been theoretically proved under a set of weak constraints. Second, we have presented two weak clustering algorithms: one, RBC, based on Bregman divergences; and another, RSPLIT, an extension of a previously presented random splitting one. Third, we have proposed an unsupervised procedure to determine the scaling parameters for a Gaussian kernel, used within a minority clustering algorithm.

We have implemented a number of approaches built from the proposed components, and evaluated them on a collection of datasets, for a comparison to other minority clustering methods in the state of the art. The results of the evaluation show how approaches based on EWOCS, and especially the one built using SRBC as weak clustering algorithm and G(AUTO) as object divergence function, are competitive with respect to—and even outperform—other minority clustering approaches in the state of the art, in terms of F1 and AUC measures of the obtained clusterings.

The completely unsupervised minority clustering approach, built from EWOCS, SRBC, G(AUTO) and an unsupervised threshold detection criterion (one of DIST, NGAUSS+VAR or 2GAUSS) already outperforms other supervised minority clustering approaches. With only the minor supervision introduced by replacing the threshold detection by NGAUSS+BEST, the resulting approach outperforms all other considered systems, including the much more supervised kMD.

At the light of the results, we believe that the EWOCS algorithm is an effective method for ensemble minority clustering, and that it allows the building of competitive and unsupervised approaches to the task. It is appealing because of its simplicity, flexibility and theoretical well-foundedness, and can hence be taken into account for clustering on a diversity of domains, where unsupervised minority clustering tasks may be the rule and not the exception.

This new unsupervised minority clustering algorithm represents a cornerstone in the work of this thesis. Its development has allowed us, as it will be shown in next chapter, to develop a completely new approach to pattern learning, based in a joint combination with minority clustering.

6 Joint Learning

> The gross and net result of it is that people who spent most of their natural lives riding iron bicycles over the rocky roadsteads of this parish get their personalities mixed up with the personalities of their bicycle as a result of the interchanging of the atoms of each of them and you would be surprised at the number of people in these parts who are nearly half people and half bicycles... when a man lets things go so far that he is more than half a bicycle, you will not see him so much because he spends a lot of his time leaning with one elbow on walls or standing propped by one foot at kerbstones.
>
> Flann O'Brien
> *The Third Policeman*

This chapter presents our proposal for joint combination of clustering and IE pattern learning. The presented approach restates relation detection as a minority clustering problem, and uses the EWOCS *algorithm presented in the previous chapter to solve it.*
Section 6.1 explains this transformation from relation detection to minority detection, and emphasizes why our approach should be able to overcome the limitations of other solutions. Section 6.2 describes the different components that are incorporated into the basic EWOCS *algorithm to solve this particular problem. An empirical evaluation on the ACE relation mention detection task is presented in Section 6.3. Last, Section 6.4 draws conclusions of our work.*

Parts of this work are also described in (Gonzàlez and Turmo, 2009).

MINORITY CLUSTERING ALGORITHMS, such as the one presented in the previous chapter, open the door to unsupervised exploration of noisy datasets—and to the discovery of dense regions within them. In particular, in this chapter we will use minority clustering to our benefit, so as to implement a joint clustering and pattern learning approach. The goal is to avoid the limitations of combinations including document clustering with respect to capturing generic relations (see Section 4.5.3).

6.1 Joint Clustering and Pattern Learning

One alternative to circumvent the problems arising from the usage of document clustering for pattern learning, as proposed in Chapter 4, is to skip the document clustering step altogether, and perform clustering directly on entity pair contexts. This is the direction taken, for instance, by Hasegawa et al. (2004), Chen et al. (2005) or Eichler et al. (2008) (see Section 2.2.2).

Similarly to these works, the starting step for the pattern learning approach we propose in this chapter is to gather, across a document collection, all pairs of entities which co-occur in the same sentence, and to generate a representation for each linguistic context in which these entity pairs occur. After this process, the obtained data will contain two types of objects:

- The majority of entity pairs will not be linked by any particular relation. The contexts which join them will be disparate, and scattered across the whole space of linguistic features.
- On the contrary, the entity pairs which are linked by some relation will often be grouped together in the linguistic feature space: the construction expressing the relation will be shared with other similarly related pairs, and hence the pairs which express the same relation using the same construction will form dense regions in the feature space—i.e., clusters.

In this setting, applying a minority clustering algorithm will hopefully detect the groups of related entity pairs as foreground clusters, and discard the non-related ones as background noise.

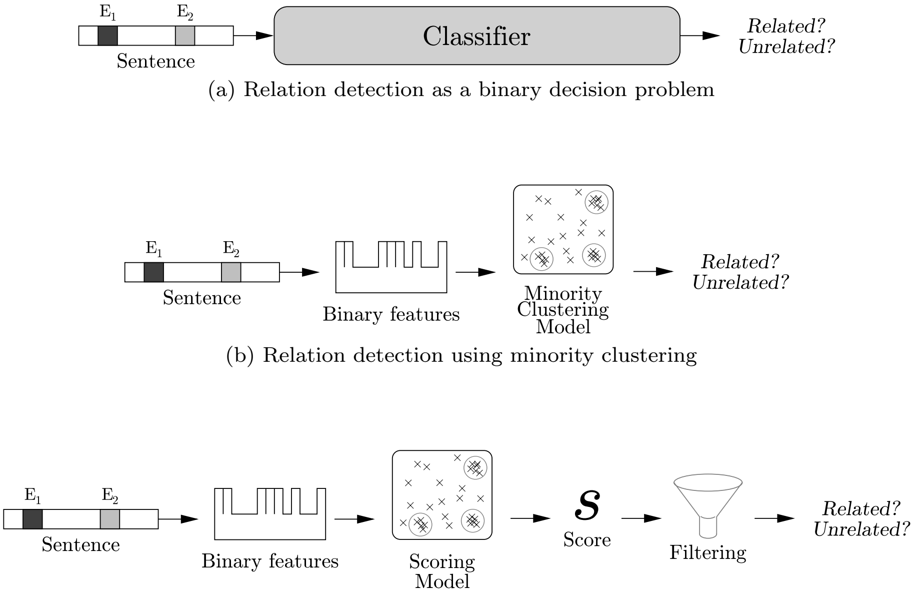
We believe this scheme is able to avoid a number of drawbacks present in other approaches:

- **Limitation to domain-specific relations:** As mentioned, one of our main motivations for discarding the approaches which combined document clustering and pattern learning was their bias towards domain-specific relations, which prevented them from detecting those of a more generic nature. In the proposed joint scheme, the source document of contexts is disregarded—allowing the formation of clusters of entities linked by both general and domain-specific relations.
- **Computational cost:** The unsupervised document clustering methods proposed in Chapter 3, and used within the sequential and collaborative methods, presents an additional drawback: that of its computational complexity. Due to the use of hierarchical clustering algorithms (Geo, IT, Hi), or to the construction of the document co-association matrix (Minor, Major), the space and time requirements of the proposed approaches grow quadratically with the number of documents. This limits the applicability of the approaches to huge document collections—such as those obtained from web data.

  On the contrary, the runtime of the Ewocs algorithm increases only linearly with the size of its input, and is thus able to cope with—and benefit from—such large datasets.
- **No model generation:** Some clustering-based approaches to unsupervised relation extraction (e.g., Hasegawa et al., 2004; Shinyama and Sekine, 2006) do not perform a generalization step, nor provide any detection model which may be applied to find relations in new data. The acquired knowledge is thus limited to the relations existing between entities in the used collection.

  As mentioned in Section 5.3.7, the Ewocs algorithm allows obtaining minority clustering models so as to classify new instances as belonging to the background or to one of the foreground clusters. This model can be used as an effective—albeit complex and not necessarily interpretable—pattern set for relation detection.
- **Limited flexibility:** Other systems (e.g., Hassan et al., 2006; Rozenfeld and Feldman, 2006) are tailored for a specific and fixed pattern format, usually word form, POS and lemma sequences, and hence it is impossible to incorporate new sources of linguistic information—such as chunks, semantics…

  In the case of Ewocs, the fact that a number of soft clustering algorithms may be plugged as inner clustering families, and the overall procedure will not depend on the object representation used by them, opens the door to the experimentation with more complex feature sets.



(a) Relation detection as a binary decision problem

(b) Relation detection using minority clustering

(c) Minority clustering decomposition as scoring and filtering

Figure 6.1: Joint approach for relation detection

Systems built using this approach will hence be able to acquire Ewocs models, usable as detection patterns for both generic and domain-specific relations, in an unsupervised fashion from potentially large document collections. Hopefully, such an achievement would place us one step closer to the goal set in Section 1.2.1: that of *unsupervised information extraction*.

6.2 Our Approach

Our approach is based on reduction of relation detection to a binary decision problem, previously used for our sequential and collaborative approaches (see Section 4.2). The reduction is depicted in Figure 6.1a[1]. Also similarly to those approaches, the first step towards determination of the relatedness of an entity pair is the generation of a binary feature vector, capturing linguistic traits in their context. However, instead of using a pattern-based decision list, the relatedness of the pair is determined using a minority clustering model (Figure 6.1b). The model determines whether new entity pair contexts belong to one of the foreground clusters or to the background, and the entities are correspondingly classified as related or unrelated. Using the Ewocs minority clustering algorithm, the classification is split in two different steps: a first scoring one, and a second filtering one (Figure 6.1c).

The learning of the models starts by gathering all contexts in which entity pairs co-occur in an unannotated corpus, and generating a feature vector for each one of them. The Ewocs minority clustering algorithm can then be applied on the obtained context matrix. Following other work (e.g., Chen et al., 2005; Hassan et al., 2006), we can optionally restrict the entities in the pair to belong to given types T_1 and T_2.

The feature generation step is an extension of that presented for the binary feature conjunction formalism of Section 4.3.1.2. However, as the Ewocs algorithm is not affected by the problem of combinatorial pattern explosion, we have devised a much richer set of feature patterns, and explored a number of combinations of them. Moreover, in order to work with binary feature vectors, we have considered a different set of weak clusterers than those proposed in the previous chapter (see Section 5.4). Last, we have identified some potential sources of bias within the scoring scheme used by Ewocs, and we propose a number of alternative scoring functions which may help in avoiding them.

[1] Reproduced from Figure 4.2.

	`w:t`	`w:tl`	`w:tw`	`c:t`	`c:tl`	`c:tw`	`w:t+c:t`	`w:t+c:tl`
Structure-based								
Distance between the pair is `%d` words	●	●	●	●	●	●	●	●
Distance between the pair is `%d` chunks	·	·	·	●	●	●	●	●
Left/rightmost entity is of type `%t`	●	●	●	●	●	●	●	●
Word-based								
Word `%d` positions before/after the left/rightmost entity...								
...has POS tag `%t`	●	●	●	·	·	·	●	●
...has lemma `%l`	·	●	·	·	·	·	·	·
...can have synset `%w`	·	·	●	·	·	·	·	·
Chunk-based								
Chunk `%d` positions before/after that containing the left/rightmost entity...								
...has type `%t`	·	·	·	●	●	●	●	●
...has a head with POS tag `%t`	·	·	·	●	●	●	●	●
...has a head with lemma `%l`	·	·	·	·	●	·	·	●
...has a head which can have synset `%w`	·	·	·	·	·	●	·	·

Table 6.1: Feature patterns used by feature sets

Sections 6.2.1 and 6.2.2 give details of the feature generation process and of the weak clustering algorithms used, respectively, and Section 6.2.3 presents the alternative cluster scoring functions we have considered.

6.2.1 Feature Generation

As mentioned in the introduction of this section, the binary feature formalism we use for our joint pattern learning approach is the same that was already presented in Section 4.3.1.2. Nevertheless, the features are no longer used to directly construct conjunction-based patterns. Instead, they are collected into a binary matrix, which is then fed as input to a minority clustering algorithm.

The use of a numerical algorithm, instead of a combinatorial one as frequent-itemset mining, allows for a larger number of features to be used without efficiency concerns, and we have hence explored richer feature sets for use within our approach. Table 6.1 contains an overview of the feature patterns used, and which ones are included by each feature set.

Similarly to Section 4.3.1.2, we have used a combination of syntactical and lexical features, at both the word and chunk level. In addition to the larger number of combinations, the main novelty here is the use of semantic information, in the form of WordNet synsets.

WordNet is a large lexical database of English, and a *de facto* standard for semantic representation within the NLP community. The basic element of semantics in WordNet is the synonym set or *synset*, which, following differential semantic theory, identify concepts by means of a set of synonym words which are all lexicalizations of it—so as to allow a user to distinguish this sense from other possible senses of the words (Fellbaum, 1998). Among the different semantic and lexical relations present in WordNet, the one that has been devoted the most attention is that of *hypernymy* and its inverse *hyponymy*, informally corresponding to the *is-a* or concept generalization relation. The noun and verb synsets are organized in a mostly tree-like hierarchy, where specific concepts are linked by hypernymy links to more general ones.

Even if WordNet has been used for research across virtually all subfields of NLP (Fellbaum, 1998), the fine-grainedness of its sense distinctions has been often cited "*as one of its main problems for practical applications*" (Agirre and López de Lacalle, 2003). For this reason, a number of works have tried either to group the synsets in clustering-like fashion (e.g., Peters et al., 1998; Agirre and López de Lacalle, 2003), or to prune the sense hierarchy, effectively disregarding distinctions below a determined tree cut line (e.g., Li and Abe, 1998; Clark and Weir, 2002).

For our purposes we have adapted the method of Li and Abe (1998), which regards the possible pruned trees as probabilistic models to generate the distribution of senses observed across a dataset. The selection of the best model—and tree cut—is made according to the minimum description length principle (Rissanen, 1978). Even if the method was originally devised to generalize verbal case frames using any sense thesaurus, it has been used subsequently to prune the WordNet hypernymy hierarchy (McCarthy, 2000; Tomuro, 2001).

In our system, the application of this method requires a preliminary scan of the document collection, in which the synsets of all nouns and verbs occurring within the context window are collected. No word sense disambiguation is performed, so all synsets for every word are considered with uniform weighting. At the end of the scan, the optimal generalization level for the WordNet tree is found for each one of the pattern positions, following Li and Abe[2]. In the second pass over the document collection, the binary features are instantiated, and, for each word, the senses and hypernyms that remain after position-dependent pruning are considered.

Similarly to previous approaches, thresholds over inter-entity distance and feature frequency are employed for efficiency. As mentioned in Section 4.3.1, relations are usually expressed using short-distance constructions, so pairs of entities further than 8 tokens away have been removed in the learning step. Features occurring in less than 10 contexts have also been discarded.

6.2.2 Weak Clustering

The previous chapter presented a number of weak clustering algorithms to be used within the Ewocs algorithm. However, the fact that the objects generated by the collection scanning process has the form of binary feature vectors led us to consider additional experimentation with other methods, more specific to binary data.

In particular, we have considered the incorporation of one probabilistic and one margin-based clustering algorithm. Next sections, 6.2.2.1 and 6.2.2.2, describe these two proposed methods in detail.

6.2.2.1 Probabilistic Clustering

The field of probability theory has often provided well-founded grounds for the development of learning algorithms. In particular, significant attention has been devoted to the family of *graphical models* (Jordan, 1998), which includes a number of popular methods such as Bayesian networks (Heckerman, 1996), hidden Markov models (Rabiner, 1989), maximum entropy classifiers (Berger et al., 1996) or conditional random fields (Lafferty et al., 2001). However, the naive Bayes framework—despite being the simplest possible one—provides methods which frequently match the performance of more complex algorithms, both probabilistic and non-probabilistic. The good behaviour of naive Bayes on classification tasks—where more often that not its assumptions do not hold—together with its simplicity and low computational cost, has fostered its use in a variety of problems, including text classification (McCallum and Nigam, 1998). Zhang (2004a) provides a theoretical analysis which tries to shed light on the reasons behind this unexpected success.

For our purposes, we have considered a mixture model of k components, where each component is a sequence of Bernoulli distributions, one per feature w, combined using the naive Bayes assumption of conditional independence given the component. Formally, the model has the form:

$$\begin{aligned}
p(x_i\,;\Theta) &= \sum_{c=1}^{k} p(y_i = c\,;\Theta)\cdot p(x_i \mid y_i = c\,;\Theta)\\
p(y_i = c\,;\Theta) &= \alpha_c\\
p(x_i \mid y_i = c\,;\Theta) &= \prod_{w=1}^{z} p(x_{iw} \mid y_i = c\,;\Theta)\\
&= \prod_{w=1}^{z} (\vartheta_{cw})^{x_{iw}} \cdot (1-\vartheta_{cw})^{1-x_{iw}}
\end{aligned}$$

The $\{\alpha_c\}$ and $\{\vartheta_{cw}\}$ are the parameters of the model, which, additionally, should accomplish the restrictions:

$$\sum_{c=1}^{k} \alpha_c = 1 \qquad \forall c \in \{1\ldots k\}, w \in \{1\ldots z\} :\ 0 \le \vartheta_{cw} \le 1$$

[2] Where positions are expressed as, for instance, *two words after the rightmost entity* (see Section 4.3.1.2).

The optimal parameters $\hat{\Theta} = (\hat{\alpha}_c, \hat{\vartheta}_{cj})$ for the model are obtained through Maximum a Posteriori estimation, using the Expectation-Maximization algorithm. As usual, Dirichlet distributions are used as priors for both the α_c and ϑ_{cj}, and conditional probabilities are identified with grades of membership to obtain a soft clustering.

This model family is similar to that of Meilă and Heckerman (2001) for document clustering (see Section 3.4.1.1), but uses Bernoulli instead of multinomial distributions, the former being more suitable for binary features. An equivalent model, but with different features, was also used for unsupervised relation extraction by Banko et al. (2007)

The only parameter that remains open in the algorithm is the number of clusters k. Similarly to the previously presented weak clusterers RSPLIT and RBC algorithms, the value k is chosen at random from the range $\{2 \dots k_{max}\}$, for a given k_{max}. Moreover, no attempt is made to provide a good starting point for the EM algorithm, and the initial model parameters are chosen at random.

We will refer to this algorithm as PROB.

6.2.2.2 Random Support Vector Clustering

Even if naive Bayes methods are able to provide competitive classifiers across a wide spectrum of tasks, it is known that they are also susceptible to perform poorly in other occasions (Domingos and Pazzani, 1997). We thus believe necessary to compare the performance of PROB with an alternative which does not assume feature independence.

The RBC algorithm presented in Section 5.4.2 is one such alternative. However, among the algorithms in the kernel method family to which it belongs, support vector machines (SVMs) have been, since its introduction by Vapnik (1995), one of the most successful ones. SVMs have been applied on a wide spectrum of ML problems (Wang, 2005). In particular, they have been used on NLP tasks such as supervised relation extraction (Zelenko et al., 2003; Bunescu and Mooney, 2005; Zhao and Grishman, 2005).

Even if support-vector-based clustering methods do exist, such as the ones proposed by Ben-Hur et al. (2002) and Zhao et al. (2008), their elevate computational cost makes them unsuitable for our purposes: i.e., to be used as weak—and thus necessarily *cheap*—clusterers within the EWOCS algorithm.

We have thus decided to modify the RBC algorithm to incorporate SVM machinery. The idea remains to select a set of seed objects and create clusters centered around those seeds. However, instead of defining the clusters according to the Bregman divergence between the objects and the seeds, we shall define them through a SVM. The classifier is trained from the seed set, with each seed in a class by itself, and then applied to the rest of the dataset, in order to assign every object to a class. A clustering is obtained by identifying these classes with clusters. We shall name this method Random Support Vector Clustering (RSVC).

The multi-class SVM framework used is that of Crammer and Singer (2001), where the SVM for a dataset $\mathcal{X} = \{x_1 \dots x_n\}$, whose objects respectively belong to classes $\mathcal{Y} = \{y_1 \dots y_n\}$, is found by optimizing the functional

$$\hat{T} = \arg\max_{T} -\frac{1}{2} \sum_{i,j=1}^{n} \sum_{c=1}^{k} K(x_i, x_j) \tau_{ic} \tau_{jc} + \beta \sum_{i=1}^{n} \tau_{iy_i}$$

subject to the restrictions

$$\forall i \in \{1 \dots n\} : \sum_{c=1}^{k} \tau_{ic} = 1 \qquad \forall i \in \{1 \dots n\}, c \in \{1 \dots k\} : \tau_{ic} < \delta(y_i, c)$$

where $\delta(x,y)$ is the usual Kronecker delta function[3].

For the particular case in which our dataset consists only of the seed set $\hat{\mathcal{X}} = \{\hat{x}_1 \dots \hat{x}_k\} \subset \mathcal{X}$, and for every seed $\hat{x}_i$ its class is $y_i = i$, the problem becomes

$$\hat{T} = \arg\max_{T} -\frac{1}{2} \sum_{i,j=1}^{k} \sum_{c=1}^{k} K(x_i, x_j) \tau_{ic} \tau_{jc} + \beta \sum_{i=1}^{k} \tau_{ii}$$

[3] This Kronecker delta function is defined as: $\delta(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{otherwise} \end{cases}$

subject to

$$\forall i \in \{1\dots k\} : \sum_{c=1}^{k} \tau_{ic} = 1 \qquad \forall i, c \in \{1\dots k\} : \tau_{ic} < \delta(i,c)$$

The obtained multi-class SVM can be used directly to obtain a hard clustering of the data. However, in order to obtain a soft clustering from the output of the classifier SVM, we have made a loose adaptation of the method of Platt (2000). The author therein proposes converting the output of a regular two-class SVM to a probabilistic output $p(c_+ \mid x_i)$—the conditional probability of object x_i to belong to the positive class c_+. The margin $M(x_i, c_+)$ between x_i and the decision hyperplane is mapped into the $[0,1]$ range by using a sigmoid function

$$p(c_+ \mid x_i) = \frac{1}{1 + e^{A \cdot M(x_i, c_+) + B}}$$

The function parameters A and B should be fitted using the training data.

Given that our algorithm is unsupervised and we do not have training data at our disposal, we have chosen $B = 0$, which makes the probability symmetric at both sides of the hyperplane. Moreover, we want the probability to increase with the value of the margin, so $A < 0$. Introducing $A' = -A > 0$, we can then reexpress the probability as:

$$p(c_+ \mid x_i) = \frac{1}{1 + e^{-A' \cdot M(x_i, c_+)}} = \frac{e^{\frac{A'}{2} \cdot M(x_i, c_+)}}{e^{\frac{A'}{2} \cdot M(x_i, c_+)} + e^{-\frac{A'}{2} \cdot M(x_i, c_+)}}$$

Taking into account that the margin for the negative class is $M(x_i, c_-) = -M(x_i, c_+)$, we finally obtain the expression.

$$p(c_+ \mid x_i) = \frac{e^{\frac{A'}{2} \cdot M(x_i, c_+)}}{e^{\frac{A'}{2} \cdot M(x_i, c_+)} + e^{\frac{A'}{2} \cdot M(x_i, c_-)}}$$

which allows a natural generalization to more than two classes as:

$$p(c \mid x_i) = \frac{e^{\frac{A'}{2} \cdot M(x_i, c)}}{\sum_{q=1}^{k} e^{\frac{A'}{2} \cdot M(x_i, q)}}$$

The coefficient $A'/2$ can be integrated into the kernel function—and hence into the margin—as a scaling parameter, and we can come to our final expression for $p(c \mid x_i)$ as

$$p(c \mid x_i) = \frac{e^{M'(x_i, c)}}{\sum_{q=1}^{k} e^{M'(x_i, q)}}$$

By identifying the classes c of the SVM with clusters π_c, and conditional probabilities with grades of membership, we finally obtain the RSvc algorithm, which is presented in Algorithm 6.1. Even if there is a number of raw approximations in the algorithm, we believe that it may still remain useful in a weak clustering context, and we have hence incorporated it to our experiments. In them, we have used a Gaussian kernel, analogous to that proposed in Section 5.4.2.

Being both based around kernel functions, there is a strong connection between RSvc and Rbc. In particular, we can prove that

Theorem 6.1
When $k_{max} = 2$, Rbc using the Gaussian kernel $K_\phi(x,y) = \alpha \cdot e^{-\gamma \|x-y\|^2}$ and RSvc using the Gaussian kernel $K'_\phi(x,y) = 2\alpha \cdot e^{-\gamma \|x-y\|^2} = 2K_\phi(x,y)$ are equivalent algorithms.

Proof See Appendix B. ■

6.2.3 Cluster Scoring

In the previous chapter, we derived the Ewocs algorithm from the study of the probability distribution, for objects x_i in a dataset $\mathcal{X}$, of the average scores assigned by clustering functions f_r

Algorithm 6.1 Random Support Vector Clustering (RSvc)

```
Input: A dataset $\mathcal{X}$
Input: A kernel function $K$
Input: A maximum number of clusters $k_{max}$
Output: A (hard or soft) all-in clustering $\Pi$ of $\mathcal{X}$
Draw a number of clusters $k$ at random from the range $\{2 \dots k_{max}\}$
        $k \in \{2 \dots k_{max}\}$
Select a subset $\hat{\mathcal{X}}$ of $k$ seeds from $\mathcal{X}$
        $\hat{\mathcal{X}} = \{\hat{x}_1 \dots \hat{x}_k\} \subset \mathcal{X}$
Optimize the SVM functional
        $\hat{T} = \arg\min_{T} -\frac{1}{2} \sum_{i,j=1}^{k} \sum_{c=1}^{k} K(\hat{x}_i, \hat{x}_j) \tau_{ic} \tau_{jc} + \beta \sum_{i=1}^{k} \tau_{ii}$
    subject to the constraints
        $\forall i \in \{1 \dots k\} : \sum_{c=1}^{k} \tau_{iq} = 1 \qquad \forall i, c \in \{1 \dots k\} : \tau_{ic} < \delta(i,c)$
Find the margin of all objects $x_i$ with respect to each one of the classes $c$
        $M(x_i, c) = \sum_{j=1}^{k} \hat{\tau}_{jc} K(x_i, \hat{x}_j)$
If Hard clustering desired then
    Assign each object $x_i$ to the cluster $\pi_c$ which gives it the largest margin,
        $\pi_c = \{x_i \in \mathcal{X} \mid \arg\max_{c} \; M(x_i, c)$
Else
    Find membership grade for each object $x_i$ and cluster $\pi_c$,
        $\text{grade}(x_i, \pi_c) = \frac{e^{M(x_i,c)}}{\sum_{q=1}^{k} e^{M(x_i,q)}}$
Return The clustering $\Pi = \{\pi_1 \dots \pi_k\}$
```

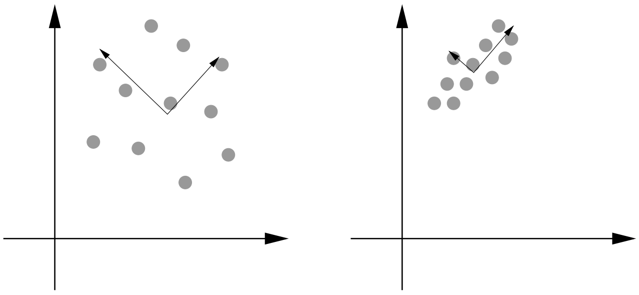
drawn at random from a family F. The fundamental step in the process was the accumulation of object scores (line 6 in Algorithm 5.1):

$$s_i^+ \leftarrow s_i^+ + s_{ri} = s_i^+ + \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_i, \pi_{rc}) \cdot \text{size}(\pi_{rc})$$

It is interesting to note here that the size of cluster π_{rc} can be regarded as a particular case of a *score* of the cluster—related to its *confidence*—and one can hence wonder if there exist other *scores* which will provide different properties to the output of Ewocs. For our experiments on joint pattern learning, we have considered a number of such scoring functions, with the aim of trying to correct some biases that we have observed, arising from the use of the raw cluster size.

More specifically we have observed that the pure Ewocs algorithm is subject to two main biases:

Clustering cardinality bias The first one of such biases comes from the fact that, as we are selecting the number of clusters k_r in each weak clustering at random from the range $\{2 \dots k_{max}\}$, clusterings with a lower cardinality have a larger influence in the final object score. The reason is that, on average, clusters in a clustering have size $|\mathcal{X}|/k_r$, and hence, the lower the value of k_r, the larger the contribution of the size(π_{rc}) to the object scores s_i^+.



(a) Loose cluster (b) Tight cluster

Figure 6.2: Relationship between cluster density and variance eigenvector decomposition

In order to compensate this bias, we propose here the use of a *normalized cluster size*, which takes into account the cardinality of the clustering. However, in some cases (especially with the PROB algorithm) the number of effective clusters in the output clustering is lower than the randomly selected k_r. For this reason, we have considered the number of non-empty clusters k_r^{NE}, which are those whose size exceeds a certain threshold th_{NE}

$$k_r^{NE} = |\{\pi_{rc} \mid size(\pi_{rc}) \geq th_{NE}\}|$$

The normalized cluster size is then the product of cluster size and the number of non-empty clusters:

$$\text{nsize}(\pi_{rc}) = \text{size}(\pi_{rc}) \cdot k_r^{NE}$$

We have used a value of $th_{NE} = 1$ for the non-empty threshold in our experiments.

Cluster density bias The second bias we have identified is that related to the differences between loose and tight clusters. As shown in Figure 6.2 for two-dimensional data, even if two clusters have the same size, one would expect clusters which are *loose* (a) to have a lesser influence on object score that those which are *tight* (b).

In order to quantify this difference between loose and tight clusters, we have used properties of the variance of the objects in the cluster: loose clusters present larger variances than tight ones. In particular, if we find the eigenvectors of the covariance matrix, which correspond to the main directions of variability within the cluster, their corresponding eigenvalues will provide a magnitude of this variability. We have defined the *radius* of a cluster to be the sum of eigenvalues of its covariance matrix.

If we consider the expectation vector $\vec{e}^{rc}$ and covariance matrix V^{rc} of cluster π_{rc}, whose entries are:

$$\begin{aligned} e_w^{rc} &= \frac{\sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc}) \cdot x_{iw}}{\sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc})} \\ V_{ww'}^{rc} &= \frac{\sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc}) \cdot (x_{iw} - e_w^{rc}) \cdot (x_{iw'} - e_{w'}^{rc})}{\sum_{x_i \in \mathcal{X}} \text{grade}(x_i, \pi_{rc})} \end{aligned}$$

the radius of the cluster is the sum of eigenvalues of this V^{rc} matrix. However, by linear algebra we know that the sum of eigenvalues of a matrix is equal to its trace and, hence, the radius can be found as the sum of the feature variances:

$$\text{radius}(\pi_{rc}) = \sum_{w=1}^{z} V_{ww}^{rc}$$

The formula requires calculation of the diagonal of the covariance matrix only, and no eigenvector decomposition.

By combining the two heuristics, we have devised a set of five different cluster scoring functions for our experiments:

Siz The original size function:

$$\text{score}_{Siz}(\pi_{rc}) = \text{size}(\pi_{rc})$$

NSiz The normalized size function:

$$\text{score}_{NSiz}(\pi_{rc}) = \text{nsize}(\pi_{rc})$$

Rad The inverse of the cluster radius[4]:

$$\text{score}_{Rad}(\pi_{rc}) = 1/\text{radius}(\pi_{rc})$$

Dns The *density* of the cluster, as a combination of size and radius:

$$\text{score}_{Dns}(\pi_{rc}) = \text{size}(\pi_{rc})/\text{radius}(\pi_{rc})$$

NDns The *normalized density* of the cluster, as a combination of normalized size and radius:

$$\text{score}_{NDns}(\pi_{rc}) = \text{nsize}(\pi_{rc})/\text{radius}(\pi_{rc})$$

6.3 Evaluation

To evaluate the validity of our approach, we have applied a number of learning methods, built from combinations of the proposed components, on the same relation detection task in which we had evaluated our sequential and collaborative approaches (Section 4.5).

Section 6.3.1 reviews the data used for the task, and Section 6.3.2 discusses the evaluation protocol. Section 6.3.3 then contains a thorough presentation and discussion of the obtained results. Final Section 6.3.4 examines the model obtained for one particular case study—so as to give an intuition of the nature of the obtained knowledge.

6.3.1 Data

Similarly to our previous evaluation, we have used the document collection coming English training data provided by the organizers of the ACE-2003, ACE-2004 and ACE-2005 evaluations. The documents come from sources of different nature: Table 6.2 contains an overview of the size, in terms of documents and words, of these different parts of the corpus[5]. The table also contains the short two-letter codes which are used in the official distribution to identify the source-specific subcollection, and which we will also employ in our discussion[6]. As mentioned previously, 100,932 entity mentions and 19,160 relation mentions between them are annotated within them. We refer back to Section 4.5.1 for a more detailed description.

Nevertheless, we have already mentioned how the proposed joint approach is able to benefit from large document collections. For this reason, in this evaluation we have also used an unannotated document set whose size is almost two orders of magnitude larger than ACE. More specifically, we have used the year 2000 subset of the Associated Press (APW) section of the AQUAINT Corpus. The considered data set contains 53,818 documents and 28,618,205 words from newswire data.

Being unannotated, an entity mention recognition process was required. To this end, the BIOS suite[7] was used. In particular, a recognition model for the first level of the ACE entity type hierarchy was trained (see Appendix C), using the ACE data itself. Applying the obtained model on the APW corpus, a total of 4,544,830 entities were automatically recognized.

[4] Clusters with a larger radius should be less rewarded than those with a smaller one. Inverting the radius is one way to achieve this effect.

[5] Reproduced from Table 4.3.

[6] The code for telephone speech has been changed from `<cts>` to `<ts>` for consistency.

[7] Freely available from `http://www.surdeanu.name/mihai/bios/`.

		2003		2004	
		Docs.	Words	Docs.	Words
<bn>	Broadcast news	147	38,298	220	69,547
<nw>	Newswire	105	67,100	223	101,109
<ts>	Telephone speech	-	-	8	14,937
<all>	TOTAL	252	105,398	451	185,593

		2005		TOTAL	
		Docs.	Words	Docs.	Words
<bc>	Broadcast conversation	60	46,587	60	46,587
<bn>	Broadcast news	226	62,820	593	170,665
<nw>	Newswire	106	54,766	434	222,975
<ts>	Telephone speech	39	48,901	47	63,838
<un>	Usenet groups	49	42,084	49	42,084
<wl>	Weblogs	119	42,316	119	42,316
<all>	TOTAL	599	297,474	1,302	588,465

Table 6.2: ACE subcollection sizes

Fac-Gpe	Gpe-Org	Org-Per	Fac-Loc	Gpe-Per	Org-Veh
Fac-Per	Gpe-Veh	Per-Veh	Gpe-Loc	Loc-Per	

Table 6.3: Evaluated entity type pairs

6.3.2 Protocol

In order to assess the performance of our Ewocs-based approach, we have implemented the method of Hassan et al. (2006), an unsupervised pattern acquisition approach for relation detection. As presented in Section 2.2.1, the method finds sequence-based patterns from the distribution of POS n-grams on an input corpus. The confidence of these patterns is then estimated from that of the extracted entity pairs—and vice-versa—using a mutual reinforcement algorithm. The fact that the algorithm obtains patterns which can later be applied on new documents makes it particularly suitable for comparison with our approaches. Being n-gram based, we will henceforth refer to this method as Grams.

Given that the distance between the candidate entities is a critical indicator of relatedness, we have also included, as a baseline, a system which determines two entities to be related if and only if their distance in tokens is lower than a certain threshold. For each entity type pair, the selected threshold is that giving the best F1 over the test data—so the results can be regarded as the upper bound achievable by such an approach. We will use the name Base to refer to this baseline.

Together with these two baselines, we have considered Ewocs-based approaches using the Prob, Rbc and RSvc algorithms as weak clusterers. We have used a default value $k_{max} = 100$ thorough all the evaluation, but other values in the range $\{2 \dots 100\}$ were tried. Within these algorithms, we have explored the use of all feature sets presented in Section 6.2.1 to extract linguistic traits from the entity contexts. A Gaussian kernel $K_\phi(x, y) = 0.1 \cdot e^{-0.1\|x-y\|^2}$ and the Best and Dist threshold selection criteria (see Sections 5.5.1 and 5.5.3) have been used.

We have also mimicked the evaluation protocol proposed by Hassan et al. (2006). In particular, we have considered 11 entity type pairs among the most frequently annotated as related in the ACE corpus—including the two type pairs, Gpe-Per and Org-Per, used by Hassan et al. in their evaluation—and performed the pattern learning and evaluation process for each one of them separately. The selected types are those listed in Table 6.3.

For our experiments, the AQUAINT collection has been used to perform the learning process, and the obtained patterns have then been evaluated by their extraction performance on the ACE documents. The previously defined metrics of precision, recall and F1 have been used to this end (see Section 4.5.2).

Additionally, in order to compare the performance of systems across different entity type pairs—where commensurability cannot be assumed—we have considered *relative performance* metrics with respect to a reference system (similar to those proposed in Section 3.5.2). For each pair, the precision, recall and F1 values of each system are divided by those of the reference one, and the means and standard deviations on these ratios are reported. However, even if relative performance ratios are informative, they are not totally sound from a statistical point of view. In order to correctly assess the significance of the results, Bergmann and Hommel hypothesis tests have been used, as usual.

For EWOCS-based approaches, five runs were performed for each combination of entity type pair, algorithm and feature set. The presented figures are the average of the results obtained across all runs.

6.3.3 Results

The performed evaluation compares a number of different aspects of the overall process of pattern learning. In order to organize the exposition of results, the following sections deal each with a single one of the involved components, starting with the cluster scoring function (Section 6.3.3.1) and following with the feature set, the weak clustering algorithm and the threshold detection criterion (Sections 6.3.3.2 to 6.3.3.4, respectively). Section 6.3.3.5 makes a brief discussion of the influence of the parameter k_{max} in the obtained results, before the last two sections 6.3.3.6 and 6.3.3.7 proceed to deal with the variations in performance across the different subcollections in the ACE corpus and the computational requirements—in terms of runtime—of the built systems.

6.3.3.1 Cluster Scoring

Figure 6.3 contains histograms for the performance—relative precision, recall and F1 with respect to SIZ—of the EWOCS-based approaches using the different clustering scoring functions. All entity type pairs and feature sets are included in the comparison. In order to evaluate the performance of the scoring and threshold detection stages isolatedly, the results in this section—and the following ones, up to 6.3.3.4—are those obtained using the threshold giving the BEST F1 value.

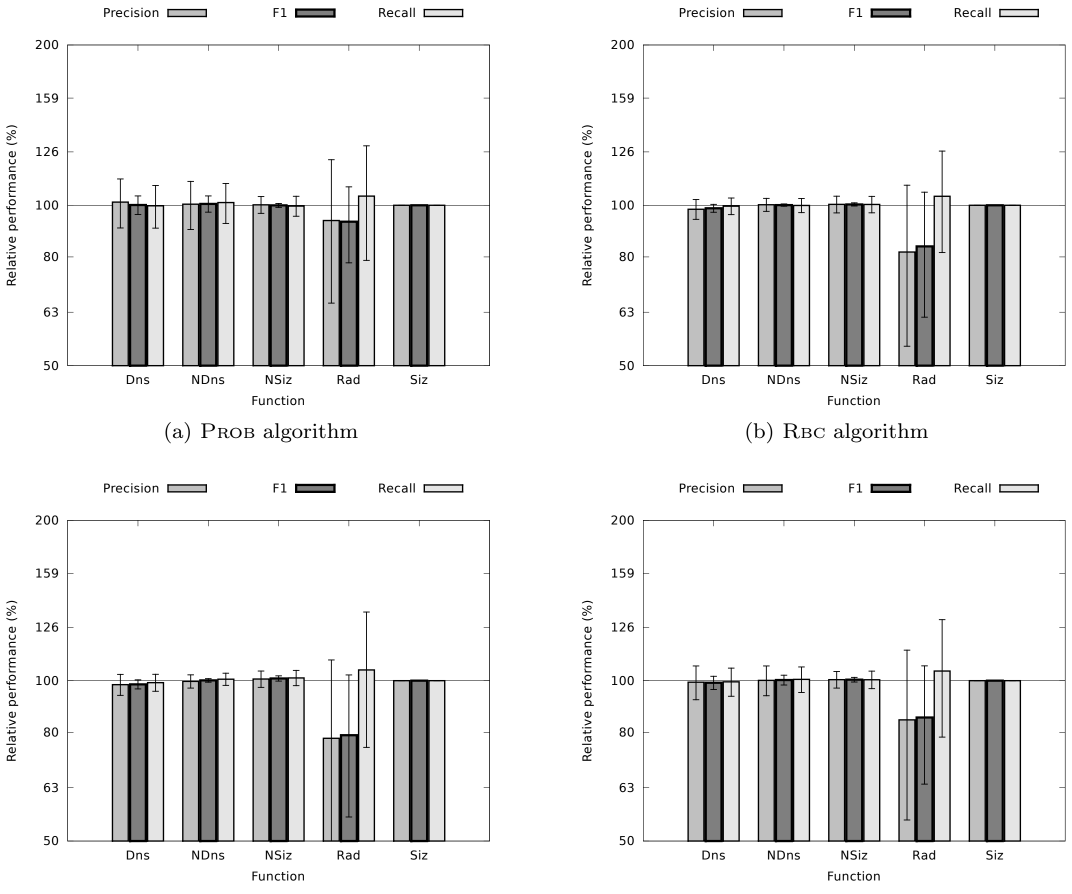
The histograms depict both the averages and the standard deviation of the values, using a logarithmic scale. Plots (a) to (c) contain the data for each one of the weak clustering algorithms, whereas plot (d) contains the aggregated information for all of them. Figure 6.4 contains the outcome of the corresponding Bergmann-Hommel hypothesis tests on the F1 metric.

Overall, the results confirm the utility of the clustering-cardinality normalization but, on the contrary, raise doubts about the suitability of the incorporation of cluster radius into the score. The hypothesis tests show how, globally and for all but the PROB algorithm, function NSIZ obtains the best results, whereas DNS and NDNS are both worse ranked than their respective radius-less counterparts SIZ and NSIZ—with function RAD, which does not include cluster-size information, achieving the lowest F1 values in all cases.

Delving more into details, the plots also bring into relief the high variance of the results which use radius information. In particular, those for RAD present a standard deviation of more than 20%. Together with the significant decrease in average F1 with respect to SIZ, and the fact that the hypothesis tests always rank RAD worse, with a significant difference within all but the RBC algorithm, these values are an indicator that cluster size information is essential to the EWOCS algorithm—in addition to being the grounds for its theoretical properties.

The variance of the results of DNS and NDNS is lower than that of RAD, but, as mentioned, the values remain lower than by using (unnormalized or normalized) cluster size alone. Only under the PROB algorithm DNS is better than SIZ, and NDNS is better than NSIZ—placing the former as the best choice for this algorithm—but without statistical significance.

The overall comparison is thus clear, and points to NSIZ as the best cluster scoring function—with *all* differences between methods being significant. For this reason, unless otherwise stated, further discussion in this section will consider the results obtained with this function.



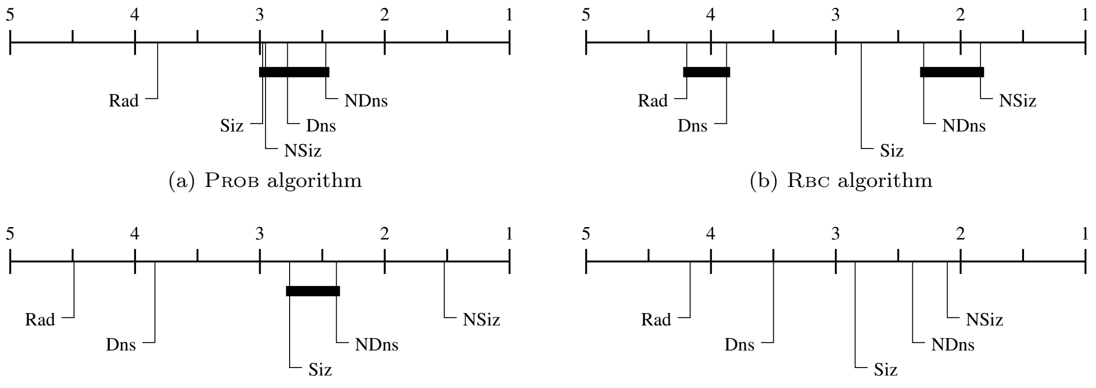
(a) Prob algorithm

(b) Rbc algorithm

(c) RSvc algorithm

(d) Overall

Figure 6.3: Performance histograms, relative to Siz, for the compared clustering scoring functions (across all pairs and features, Best threshold)



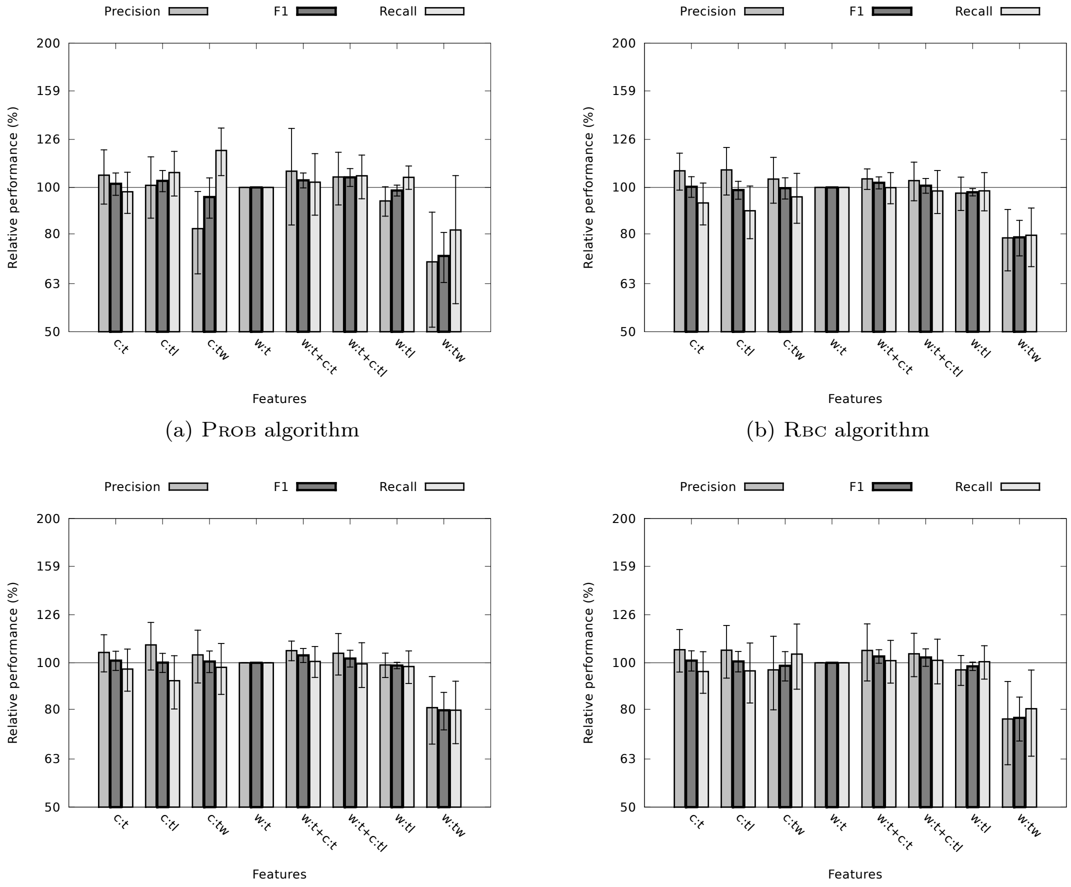
(a) Prob algorithm

(b) Rbc algorithm

(c) RSvc algorithm

(d) Overall

Figure 6.4: Bergmann-Hommel tests on F1 for the compared clustering scoring functions (across all pairs and features, Best threshold)



(a) Prob algorithm

(b) Rbc algorithm

(c) RSvc algorithm

(d) Overall

Figure 6.5: Performance histograms, relative to `w:t`, for the compared feature sets (across all pairs, NSiz function, Best threshold)

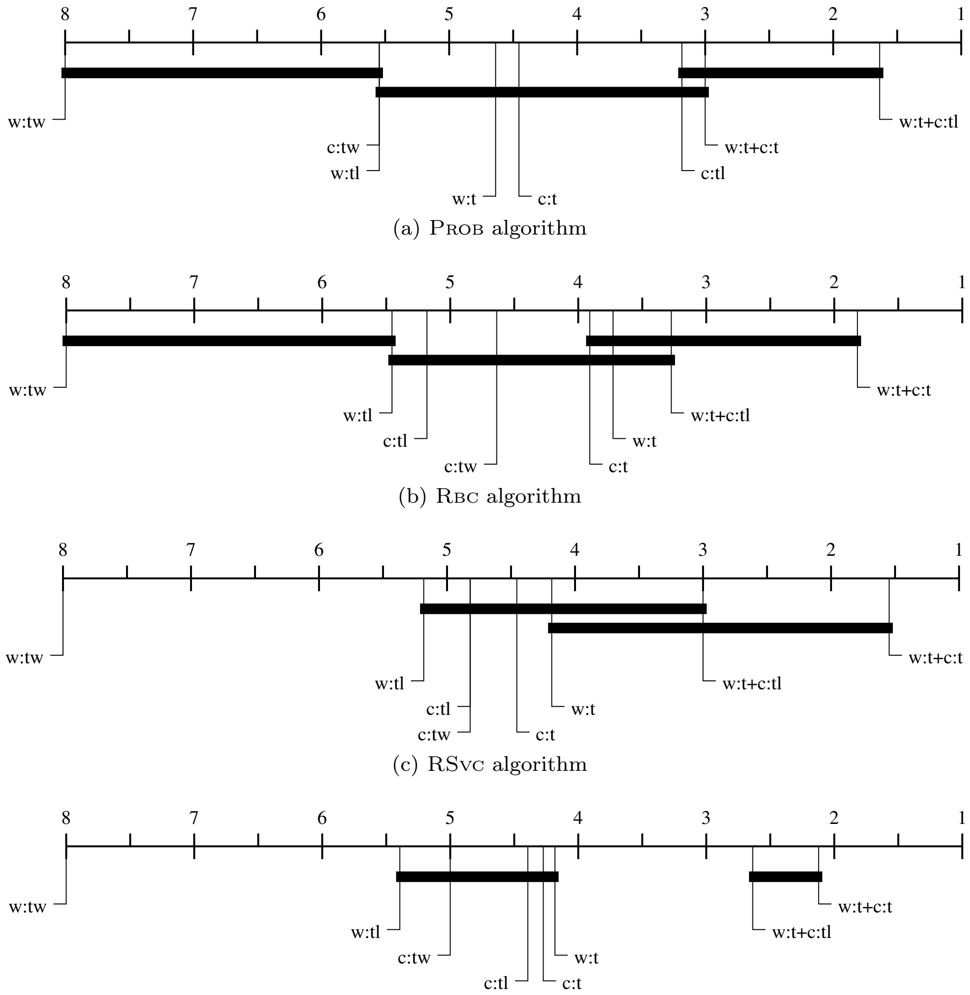
6.3.3.2 Feature Set

Figure 6.5 contains a new set of logarithmic histograms, which depict the performance of Ewocs-based approaches using the different feature sets, relative to that obtained with the simplest one, `w:t`. Figure 6.6 graphically presents the results of the corresponding F1 Bergmann-Hommel hypothesis tests.

Overall, the evaluation points to the two combined feature sets, `w:t+c:t` and `w:t+c:tl`, as the best choices in terms of F1 score, and suggest the failure of the procedure proposed to incorporate semantic information into `c:tw` and, especially, `w:tw`. Besides, the results bring up again the lack of correspondence between feature set extension and performance increase in unsupervised learning problems.

To be more specific, among the word-only feature sets, using any algorithm the option which achieves the best F1 results is the simplest `w:t` set, which includes only the POS-tag information. The inclusion of lemmas `w:tl` causes a slight decrease in all cases; whereas the case of `w:tw` is clear: it is systematically ranked as the worst feature set by the hypothesis tests in all cases—with an statistically significant difference under the RSvc algorithm and in the overall comparison.

This phenomenon—namely, the decrease of system performance caused by the incorporation of more features—has been identified as characteristic of unsupervised learning settings in previous chapters. In this case, the addition of the lemmas, and particularly the senses, of *all* words in the entity pair contexts produces an explosion of the number of features. This can be observed in Table 6.4, which contains the average number of features—across all entity pairs, and after the generation and frequency pruning steps—obtained for each one of the used feature sets. The figures there confirm an explosion: the values for `w:tl` and `w:tw` are the highest among all sets, with



(a) Prob algorithm

(b) Rbc algorithm

(c) RSvc algorithm

(d) Overall

Figure 6.6: Bergmann-Hommel tests on F1 for the compared feature sets (across all pairs, NSiz function, Best threshold)

`w:t`	784.9	`c:t`	1029.9	`w:t+c:t`	1808.7
`w:tl`	19362.1	`c:tl`	18047.5	`w:t+c:tl`	18989.2
`w:tw`	25123.0	`c:tw`	11551.0		

Table 6.4: Average number of features (after generation and frequency pruning) for each compared feature set

over 19, 000 and 25, 000 features, respectively. In particular, the large number of features of `w:tw`, together with its poor performance, arises questions about the suitability of the WordNet-pruning method of Li and Abe (1998) for this purpose.

Among chunk-based feature sets, there is not much difference between `c:t` and `c:tl`, even though the inclusion of chunk head lemmas produces a large increase in the number of features. In general, both feature sets generate patterns with higher precision and lower recall than those of `w:t`, being overall ranked worse in terms of F1 by the Bergmann-Hommel tests, when using Rbc and RSvc. The tendency is reversed under Prob, but neither in this nor in the other two algorithms significant differences are found. Regarding `c:tw`, in this set the WordNet-pruning strategy is successful in controlling the number of features, as only 11, 500 are used during the minority clustering process—much less than the 18, 000 used with `c:tl`. The hypothesis tests deem its results comparable to those of `c:t` and `c:tl`: no significant differences are found with respect to the other two chunk-based sets, nor with the reference `w:t`.

	Base			Grams			Prob			Rbc			RSvc		
	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
Fac-Gpe	61.1	70.6	**65.5**	67.6	56.8	**61.7**	71.9	68.5	**70.2**	65.5	77.0	**70.7**	64.6	78.8	**71.0**
Fac-Loc	58.0	68.9	**63.0**	64.7	29.7	**40.7**	53.3	80.4	**64.1**	61.7	66.2	**63.8**	61.6	66.2	**63.8**
Fac-Per	36.8	58.0	**45.0**	51.0	22.9	**31.6**	44.7	53.2	**48.4**	38.2	59.1	**46.4**	37.8	59.5	**46.2**
Gpe-Loc	57.2	74.5	**64.7**	67.8	58.6	**62.8**	69.4	65.6	**67.5**	61.6	71.9	**66.3**	61.4	72.0	**66.3**
Gpe-Org	60.1	69.1	**64.3**	68.3	73.8	**70.9**	65.5	78.5	**71.3**	58.6	77.8	**66.8**	61.8	73.6	**66.9**
Gpe-Per	61.0	55.1	**57.9**	55.1	62.6	**58.6**	69.0	55.5	**61.5**	64.7	58.0	**61.2**	63.8	59.0	**61.3**
Gpe-Veh	57.1	63.1	**59.9**	71.1	50.6	**59.1**	63.5	70.5	**66.8**	63.9	69.8	**66.7**	65.1	69.3	**67.1**
Loc-Per	34.1	54.7	**42.0**	47.9	25.7	**33.4**	37.2	58.6	**45.5**	37.9	49.1	**42.8**	38.2	48.5	**42.7**
Org-Per	57.6	58.1	**57.9**	52.1	71.4	**60.2**	67.1	60.3	**63.5**	56.5	70.0	**62.5**	56.5	71.5	**63.1**
Org-Veh	67.7	66.3	**67.0**	91.1	50.5	**65.0**	70.1	70.3	**70.2**	78.2	65.3	**71.2**	78.7	64.6	**70.9**
Per-Veh	32.7	57.8	**41.8**	59.1	24.5	**34.7**	60.3	37.5	**46.1**	36.1	57.3	**44.3**	36.6	56.5	**44.4**

Table 6.5: Relation detection results across the different entity type pairs (NSiz function, `w:t+c:t` features, Best threshold)

Therefore, the two combined feature sets are those which, as mentioned, occupy the high end of all Bergmann-Hommel tests. Sets `w:t+c:t` and `w:t+c:tl` obtain the best results. Moreover, they seem rather robust: their results present a lower variance than those of any other set but `w:tl`. The results for `w:t+c:t` are slightly better than those with chunk head lemma information, and it is ranked the best in the overall comparison. Being also the simplest of the two, its use will be assumed in the remaining result analyses unless otherwise stated.

The results in this section suggest that both word- and chunk-level information are useful to the task of relation detection, and that the Ewocs algorithm is able to benefit from them. However, it remains necessary to find ways to avoid feature explosion, which degrades the performance of the approaches. The problem is not unique to Ewocs, but known to be common to most other unsupervised learning algorithms (for further discussion, see for instance Kim et al., 2000; Dy and Brodley, 2004).

It is also interesting to note how the Prob algorithm seems to be able to benefit from more complex feature sets—in spite of being based on a naive Bayes model, whose performance might have been damaged by the increasing lack of independence coming with feature set growth.

6.3.3.3 Weak Clustering

The histograms and hypothesis tests shown in Figures 6.7 and 6.8, respectively, contain the results obtained by Ewocs using the three different algorithms as inner weak clusterers. In the histograms, the performance is plotted relative to that of the baseline distance-based method Base.

As seen in the histogram and confirmed by the tests, all Ewocs-based methods obtain higher precision and recall than Base, and hence systematically achieve better F1 values. The strict POS-sequence formalism used by Grams leads it to the highest precision scores, but damages its recall and makes it score a low F1 value, even lower than that of the baseline.

The differences in terms of precision, recall and F1 between Prob and Base are deemed significant despite the reduced number of entity pairs on which the evaluation is performed. Regarding Grams, it is significantly outperformed in terms of recall and F1 by all Ewocs systems, and the advantage it obtains with respect to precision is small and not significant with respect to Prob.

In the comparison among Ewocs methods, it is thus Prob which sets himself as the best option, obtaining both better precision and recall than their kernel-based counterparts. Lack of feature independence does hence not seem to damage the performance of the naive-Bayes-based method for the task. The relative merits of Rbc and RSvc are quite similar: even if RSvc obtains better precision results, they are deemed equivalent by the Bergmann-Hommel tests in terms of recall, and the overall difference of F1 is small and non-significant.

We believe these are excellent results, which state the validity of our joint clustering and pattern learning combination strategy, and in particular of the Ewocs-Prob approach.

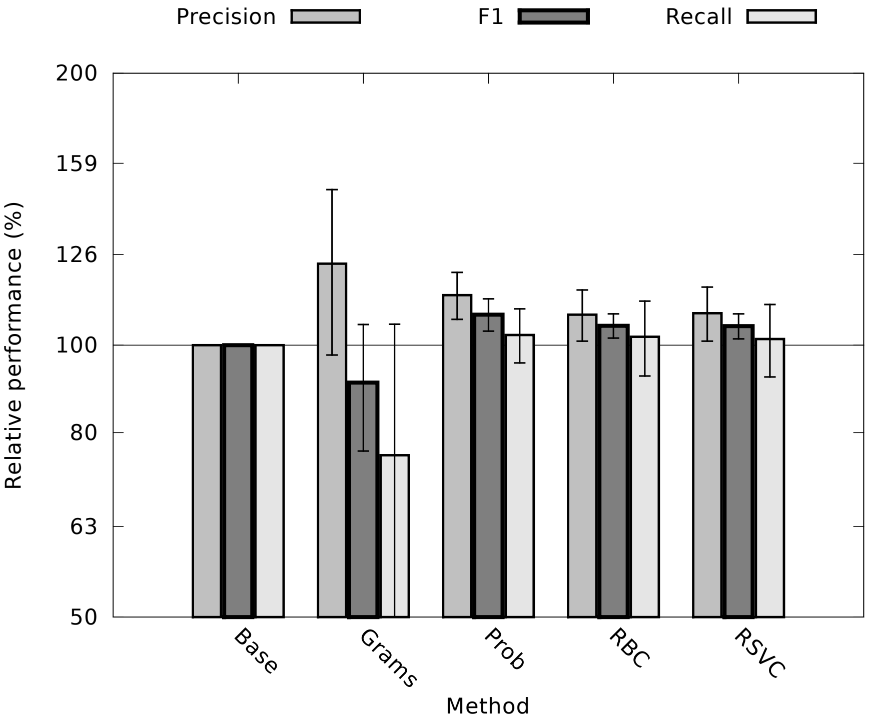
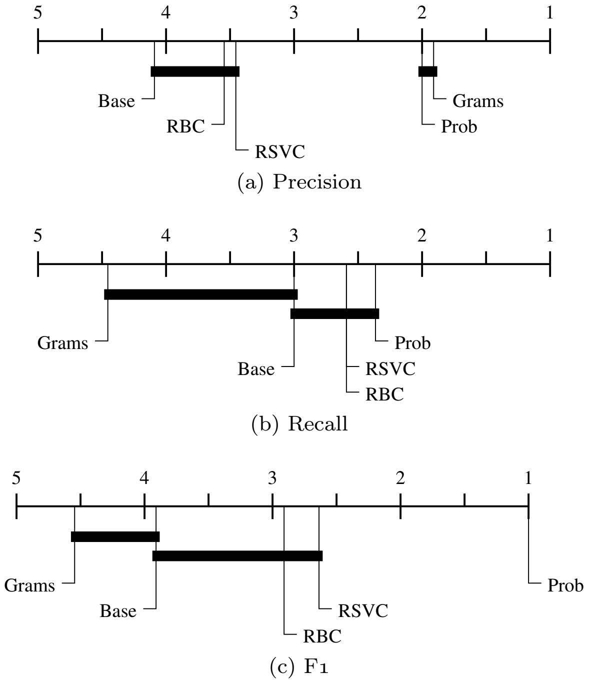


Figure 6.7: Performance histograms, relative to Base, for the compared approaches (across all pairs, NSiz function, `w:t+c:t` feature set, Best threshold)



(a) Precision

(b) Recall

(c) F1

Figure 6.8: Bergmann-Hommel tests on all metrics for the compared approaches (across all pairs, NSiz function, `w:t+c:t` feature set, Best threshold)

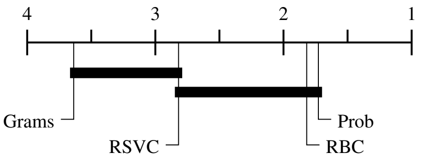


Figure 6.9: Bergmann-Hommel tests on F1 for the compared approaches (across all pairs, NSiz function, `w:t` features, Best threshold)

For reference, Table 6.5 contains the detailed scores of each one of the proposed approaches for the different entity type pairs. The values therein confirm the conclusions drawn from the histograms and hypothesis tests: the Ewocs-based approaches outperform Grams on all pairs, and Base on most of them—with Prob in turn outperforming Rbc and RSvc. The gap in terms of F1 between Prob and Grams is often considerable: 8.5% for Fac-Gpe, 13.4% for Fac-Loc, 16.8% for Fac-Per...

Regarding the precision-recall trade-off, Grams markedly favours the former, whereas for the other approaches the results are more balanced or, occasionally, shifted towards the latter.

POS-only patterns One question that is left open by the previous results is whether the difference in performance between Grams and the Ewocs-based approaches comes from the use of the minority clustering framework *per se*; or whether, on the contrary, it is only the fact that it is possible to incorporate additional information into the patterns—such as chunk head POS-tags and lemmas—which boosts their results. To find an answer, we have compared the results of Grams with those of Prob, Rbc and RSvc using the `w:t`, which includes only word POS-tags, similarly to Grams patterns.

Table 6.6 contains the results obtained by the Ewocs-based approaches under these conditions, compared to those of Grams. Even if the results are slightly lower than those obtained with `w:t+c:t`, in most cases the comparison is still favorable for the Ewocs-based approaches, particularly for Prob. The performance of Rbc and RSvc occasionally falls below that of Grams (Gpe-Org, Org-Per), but Prob obtains better F1 values across all pairs.

The Bergmann-Hommel test on the F1 scores, shown in Figure 6.9, confirms this behaviour and assesses the significance of the difference between Prob and Grams. It is thus clear that the improvement of the results over the Grams baseline is not only caused by the enrichment of the patterns with additional features, but also by the joint detection-as-minority-clustering approach we have adopted—and which has allowed the use of the Ewocs algorithm.

6.3.3.4 Threshold Detection

So as to evaluate the last remaining component, that of threshold detection, Table 6.7 contains the results obtained, for each one of the entity pair types, by the Ewocs-based approaches, using the Dist and Best threshold detection methods. The results for baseline method Grams are replicated from Table 6.5 for ease of comparison.

As seen in the table, the goodness of the threshold determined by Dist varies across the different entity type pairs and methods. Some of them allow scores close to the Best achivable (e.g., Org-Per), whereas others cause a considerable loss in the F1 score of the output (e.g., Fac-Gpe). Overall, the thresholds detected by Dist from the output of the Prob algorithm are tighter to Best than those coming from Rbc and RSvc.

It is interesting to note that the F1 values obtained by Prob using the Dist criterion are higher than the upper bound value (Best) achievable by the Grams approach in all but the Gpe-Org pair, in which the latter obtains a small favourable gap. Using Rbc and RSvc, the obtained values also usually exceed those of Grams, but there are more pairs for which the behaviour is reversed—in most of them because the threshold is far from the Best one: only for the Gpe-Org pair the maximum achievable value is below that of Grams.

Figure 6.10 contains the Bergmann-Hommel hypothesis tests on F1 score under these conditions. The tests confirm the advantage of Prob, RSvc and Rbc with respect Grams; but also determine that, in this comparison, the performance of Rbc falls below that of the baseline system, a fact

	Grams			Prob			Rbc			RSvc		
	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
Fac-Gpe	67.6	56.8	**61.7**	63.5	71.5	**67.2**	58.8	80.2	**67.8**	62.2	74.5	**67.6**
Fac-Loc	64.7	29.7	**40.7**	58.9	69.2	**63.6**	60.9	65.3	**62.8**	54.8	68.8	**60.6**
Fac-Per	51.0	22.9	**31.6**	41.3	47.1	**44.0**	37.3	58.5	**45.5**	38.2	56.1	**45.5**
Gpe-Loc	67.8	58.6	**62.8**	62.2	71.9	**66.7**	57.8	75.0	**65.3**	55.9	76.1	**64.4**
Gpe-Org	68.3	73.8	**70.9**	77.4	67.7	**72.2**	59.4	72.2	**65.2**	60.3	70.7	**65.0**
Gpe-Per	55.1	62.6	**58.6**	68.7	52.7	**59.6**	57.9	62.0	**59.9**	57.8	60.3	**59.0**
Gpe-Veh	71.1	50.6	**59.1**	63.6	62.1	**62.8**	60.9	64.5	**62.6**	59.4	63.2	**61.2**
Loc-Per	47.9	25.7	**33.4**	39.1	52.4	**44.8**	36.1	56.1	**43.9**	35.5	56.8	**43.7**
Org-Per	52.1	71.4	**60.2**	55.4	67.5	**60.8**	55.6	61.9	**58.6**	54.3	64.5	**58.8**
Org-Veh	91.1	50.5	**65.0**	77.6	65.0	**70.7**	82.5	65.1	**72.8**	80.6	65.0	**71.9**
Per-Veh	59.1	24.5	**34.7**	34.6	55.2	**42.5**	33.5	60.6	**43.2**	33.3	57.8	**42.2**

Table 6.6: Relation detection results across the different entity type pairs (NSiz function, `w:t` features, Best threshold)

		Grams			Prob			Rbc			RSvc		
		Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1	Prc	Rec	F1
Fac-Gpe	Dist	–	–	–	70.5	55.2	**61.9**	81.3	38.6	**52.3**	81.5	43.8	**56.9**
	Best	67.6	56.8	**61.7**	71.9	68.5	**70.2**	65.5	77.0	**70.7**	64.6	78.8	**71.0**
Fac-Loc	Dist	–	–	–	65.1	60.1	**62.5**	76.0	38.1	**50.7**	73.6	40.4	**52.1**
	Best	64.7	29.7	**40.7**	53.3	80.4	**64.1**	61.7	66.2	**63.8**	61.6	66.2	**63.8**
Fac-Per	Dist	–	–	–	52.8	37.6	**43.8**	60.9	22.0	**32.3**	57.8	23.3	**33.2**
	Best	51.0	22.9	**31.6**	44.7	53.2	**48.4**	38.2	59.1	**46.4**	37.8	59.5	**46.2**
Gpe-Loc	Dist	–	–	–	60.8	67.6	**64.0**	71.9	47.7	**57.3**	71.3	48.0	**57.2**
	Best	67.8	58.6	**62.8**	69.4	65.6	**67.5**	61.6	71.9	**66.3**	61.4	72.0	**66.3**
Gpe-Org	Dist	–	–	–	70.1	64.9	**67.2**	67.4	63.9	**65.6**	66.2	65.8	**66.0**
	Best	68.3	73.8	**70.9**	65.5	78.5	**71.3**	58.6	77.8	**66.8**	61.8	73.6	**66.9**
Gpe-Per	Dist	–	–	–	62.5	57.7	**60.0**	71.2	51.3	**59.6**	70.8	51.6	**59.7**
	Best	55.1	62.6	**58.6**	69.0	55.5	**61.5**	64.7	58.0	**61.2**	63.8	59.0	**61.3**
Gpe-Veh	Dist	–	–	–	54.5	73.8	**62.7**	64.1	67.8	**65.9**	63.3	69.4	**66.2**
	Best	71.1	50.6	**59.1**	63.5	70.5	**66.8**	63.9	69.8	**66.7**	65.1	69.3	**67.1**
Loc-Per	Dist	–	–	–	43.2	39.2	**41.0**	48.2	29.8	**36.8**	48.0	31.1	**37.6**
	Best	47.9	25.7	**33.4**	37.2	58.6	**45.5**	37.9	49.1	**42.8**	38.2	48.5	**42.7**
Org-Per	Dist	–	–	–	66.9	59.3	**62.8**	68.1	52.6	**59.3**	67.3	54.5	**60.2**
	Best	52.1	71.4	**60.2**	67.1	60.3	**63.5**	56.5	70.0	**62.5**	56.5	71.5	**63.1**
Org-Veh	Dist	–	–	–	73.1	65.0	**68.8**	70.2	66.3	**68.2**	69.4	66.3	**67.8**
	Best	91.1	50.5	**65.0**	70.1	70.3	**70.2**	78.2	65.3	**71.2**	78.7	64.6	**70.9**
Per-Veh	Dist	–	–	–	42.4	44.3	**41.9**	53.1	35.5	**42.6**	51.1	36.8	**42.8**
	Best	59.1	24.5	**34.7**	60.3	37.5	**46.1**	36.1	57.3	**44.3**	36.6	56.5	**44.4**

Table 6.7: Relation detection across the different entity type pairs (NSiz function, `w:t+c:t` features)

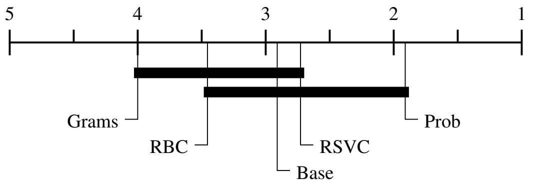
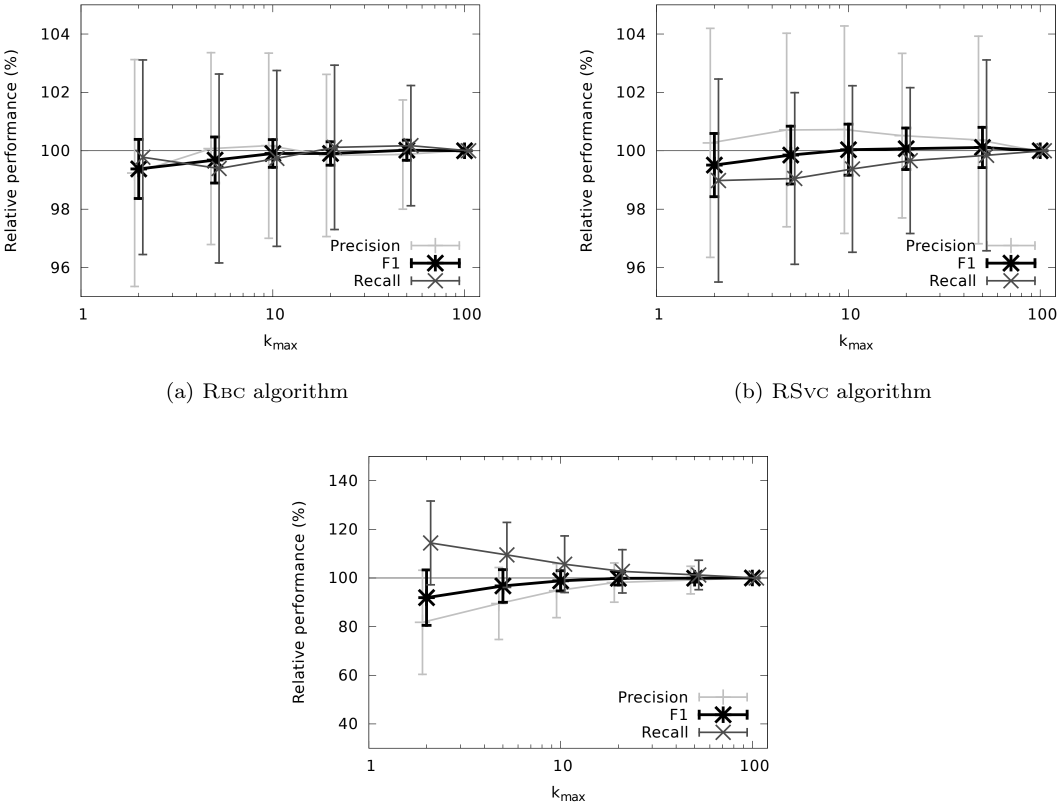


Figure 6.10: Bergmann-Hommel tests on F1 for the compared approaches (across all pairs, NSIZ function, `w:t+c:t` features, CUT threshold)



(a) RBC algorithm

(b) RSVC algorithm

(c) PROB algorithm

Figure 6.11: Influence of the k_{max} parameter on relation detection performance (across all pairs and features, NSIZ function, BEST threshold)

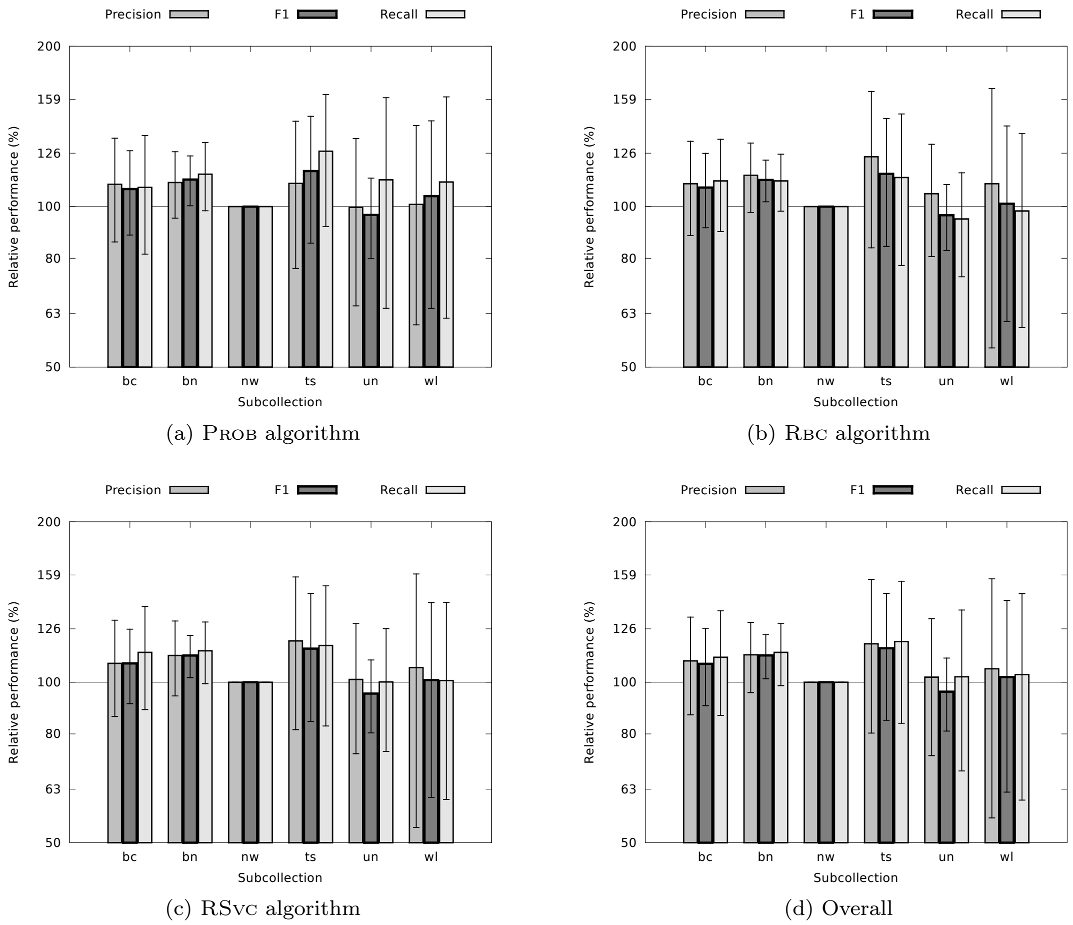
which may raise slight concerns. Nevertheless, the difference is not significant and it is important to remember that the results for BASE are an upper bound.

Given that by incorporating the DIST threshold selection the last remains of supervision are removed, the results of this section confirm that it is possible to build a completely unsupervised system for relation detection pattern acquisition using the proposed joint learning approach—and one whose extractions outperform other compared approaches in terms of the considered metrics.

6.3.3.5 Influence of k_{max}

One element that has been so far overlooked in the present evaluation is the influence on the relation detection process of the k_{max} parameter, for which a value $k_{max} = 100$ has always been used so far. The minority clustering evaluation carried out in the previous chapter showed that parameters R and k_{max} of RSPLIT and RBC did not require careful tuning (see Section 5.6.4.3). This section attempts to provides a similar study for the relation detection task at hand.

Figure 6.11 contains plots of the relative values of precision, recall and F1 using the three consid-



(a) Prob algorithm

(b) Rbc algorithm

(c) RSvc algorithm

(d) Overall

Figure 6.12: Performance histograms, relative to <nw>, on the ACE subcollections (across all pairs, NSiz function, `w:t+c:t` feature set, Best threshold)

ered weak clustering algorithms, as a function of k_{max}. The values are relative to the performance obtained using the $k_{max} = 100$. As seen in plots (a) and (b), algorithms Rbc and RSvc are almost unaffected by the setting of k_{max}. The standard deviation of the results is barely 2%, and the difference from using $k_{max} = 2$ or $k_{max} = 100$ is lower than 1% in terms of precision, recall and F1. The algorithms thus seem portable without any special parameter tuning requirements.

However, plot (c) seems to show a different picture. Setting $k_{max} = 2$ instead of $k_{max} = 100$ can bring, in average, an almost 20% increase in recall at the expense of a slightly larger decrease in precision. Overall, F1 can drop some 5%. The standard deviation of these values is also in the order of 20%. Nevertheless, this variability is progressively reduced as k_{max} increases, and the values become stable from $k_{max} = 20$ on. Thus, even if k_{max} does indeed have an influence in the performance of Prob, the results suggest that coarsely setting the parameter to a relatively large value suffices to obtain competitive results, and that hence no fine-tuning of its value should be required.

6.3.3.6 Portability Across ACE Subcollections

Given that the pattern learning process uses a subset of the AQUAINT corpus consisting only of newswire documents, but the evaluation is performed on the ACE corpus which contains documents coming from heterogeneous sources, another open question is whether the obtained relation detection model is specific to newswire language—and how it performs over texts from different sources.

Figure 6.12 contain performance histograms for the Ewocs-based approaches over the different subcollections within the ACE corpus, relative to their performance on <nw>. The four plots present

	<all>	<bc>	<bn>	<nw>	<ts>	<un>	<wl>
BBN	**33.4**	24.7	34.0	33.7	42.6	31.7	34.8
UPC	**33.1**	24.1	38.2	33.2	43.6	20.5	27.8
LCC	**32.5**	25.5	42.3	41.0	41.2	54.5	4.6

Table 6.8: ACE scores on the Relation Mention Detection task, for the systems taking part in the ACE-2007 evaluation

a similar shape, independently of the used weak clustering algorithm, and all show how the change of subcollection does not necessarily imply a decrease of performance in terms of precision or recall—on the contrary, in many cases an improvement happens. Nevertheless, the change does produce an increase in the variance of the results, which is larger the more the nature of the documents differs from newswire text.

Curiously enough, the average performance on all other subcollections of ACE is higher than that on <nw> except for F1 on <un>. <bn> is the subcollection on which the results are better: all approaches obtain higher scores than on <nw>, and their variance is quite low. The results for <bc> are somewhat lower and more disperse, but nevertheless the comparison with <nw> remains positive. On the contrary, for the other three subcollections the variance of the results is large, becoming quite unwieldy over <wl> documents. Regarding the relative performances, the highest scores of all are achieved on <ts> conversations, whereas <un> seems the hardest section of all, and the F1 scores on it are lower than on <nw>.

Despite the variations across domains, we believe these results suggest that, even if the learning process was performed on newswire data, the learned patterns are general enough to be successfully applied on documents from other domains—and, in fact, often to better results than on the newswire domain itself.

For the sake of comparison, we have included, in Table 6.8, the official results reported by the organizers on the ACE-2007 Relation Mention Detection task, using the official value-based scores (ACE, 2007). The evaluation used the ACE-2005 corpus, which is a subset of our aggregated ACE collection, and considered *full runs*—i.e., relation extraction was performed using system-detected entity mentions. Despite the differences in evaluation metric and protocol, and the fact that the approaches are *supervised*, we believe information about the relative performance of the systems across the different subcollections can be relevant to the discussion. In particular, we want to emphasize how, with the exception of <bc>, the trends that we have observed about the average scores on each division are also present in the results obtained by BBN and UPC participants.

6.3.3.7 Runtime

The last aspect to be considered in the present evaluation is the computational cost of the Ewocs-based algorithms—not only their asymptotic behaviour, but also the actual runtime taken by the clustering pattern learning process.

Figure 6.13 contains a logarithmic scatter plot of runtime versus the size of the matrix containing all pair contexts (i.e., the number of collected contexts times the number of generated features)[8]. The plot confirms the expected linear behaviour of the Ewocs algorithm, but also puts in relief the different requirements of the three considered weak clustering algorithms. Whereas RSvc and, particularly, Rbc are cheap and allow matrices of sizes in the order of thousands of millions of entries to be processed in less than one hour, the required time clustering of the same matrices using Prob takes is more than one order of magnitude higher.

In addition to its runtime, Prob also has harder computational demands in terms of space: its parameter matrix cannot be represented in a sparse form, and its memory requirements can also become orders of magnitude higher than those of margin-based weak clustering algorithms. Even if this has not been an issue for problems the size of our evaluation collections, the trade-off of performance versus computational parsimony may need to be taken into account as datasets become larger and larger (such as Web-scale ones)—and hence may eventually tip the scales towards the use Rbc or RSvc.

[8] The matrices are stored in sparse format, so the actual memory footprint is much lower.

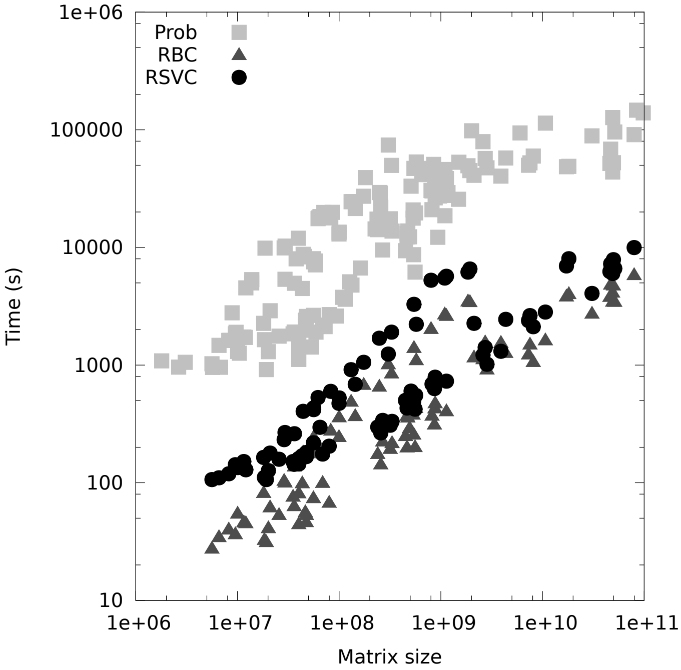


Figure 6.13: Runtime versus context matrix size scatter plot

6.3.4 Case Study

In order to give an intuition of the nature of the knowledge obtained using Ewocs, in this section we will examine a sample of the results obtained for one particular entity type pair and learning setting. This analysis is presented here for illustrative proposes only—it intends to be qualitative more than quantitative, and no general conclusions should be drawn from the behaviour of the algorithm in this particular case.

More specifically, we have considered the learning of relation detection for the Fac-Gpe pair of entity types, and the models obtained using:

- the Prob weak clustering algorithm,
- the NSiz scoring function,
- the `w:t+c:tl` feature set,
- and the Best threshold selection criterion.

This is one of the settings in which the best results were obtained. Moreover, `w:t+c:tl` contains most feature patterns we have considered (see Section 6.2.1)—in fact, all but the semantic ones, which were found in the evaluation to be too noisy to be useful for the task (see Section 6.3.3.2).

Next Section 6.3.4.1 examines the relevance of different feature types to the task, whereas Section 6.3.4.2 presents a small set of sample sentences from the corpus where correct, incorrect and missed extractions occur.

6.3.4.1 Feature Relevance

In order to quantify the impact of each individual feature in the results of the Ewocs algorithm, we have studied the conditional expectations of the scores assigned by the obtained model. Recalling Section 6.2.3, the score $s_x^\star$ assigned by Ewocs to an object x_x can be written as:

$$s_x^\star = \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_x, \pi_{rc}) \cdot \text{score}(\pi_{rc})$$

We can consider the expectation of this score across all possible objects, as well as its expectations conditioned to whether certain feature f is active or not in x_x. We can then define the *expected score offset* and *relevance* of a feature as

Definition 6.2 (Feature expected score offset and relevance)
The ***expected score offset*** *of a feature f in an* Ewocs *model is the difference between the conditional expectation of the score $s^\star_x$ of an object x_x given that f is active in x_x, and the expectation of the score given that f is inactive:*

$$\Delta_f = E^1_f - E^0_f = E[s^\star_x \mid x_{xf} = 1] - E[s^\star_x \mid x_{xf} = 0]$$

The ***relevance*** *of feature f is the absolute value of its expected score offset: $\overline{\Delta}_f = |\Delta_f|$*

Features with large relevances will be strong indicators of relatedness (if their expected score offset is positive) or non-relatedness (if it is negative).

We can also prove that

Proposition 6.3
When using the Prob *clustering algorithm within* Ewocs*, the expected score offset of feature f can be found from the model parameters as:*

$$\Delta_f = \frac{1}{N} \sum_{\Pi_r} \left(\frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot \vartheta_{rcf} \cdot \text{score}(\pi_{rc})}{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot \vartheta_{rcf}} - \frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot (1 - \vartheta_{rcf}) \cdot \text{score}(\pi_{rc})}{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot (1 - \vartheta_{rcf})} \right)$$

Proof See Appendix B. ■

These quantities thus defined, Table 6.9 contains the most relevant features found for one run of Ewocs on the Fac-Gpe entity type pair data, using the learning setting considered in this section. The 30 most relevant features with positive expected score offset, and the 15 most relevant features with negative offset are listed.

This set of most relevant features points to two patterns as the most common indicators of relatedness for this pair of entity types:

- Contexts in which the the Fac and Gpe entities are juxtaposed, the latter being a nominal complement of the former and the facility being a common noun. In this case:
  - The distance between the two entities is zero chunks—i.e., both entities belong to the same one (`dist=0ch`).
  - The distance between the two entities is one token—i.e., there is no token between them (`dist=1tk`).
  - The chunk is a noun phrase (`ch/common/type=NP`).
  - The head of the chunk is the Fac entity (`ch/common/head-tag=FAC`). Moreover, it is a common noun (`ch/common/head-tag=NN`), which corresponds to the facility type word (`ch/common/head-lemma=hospital`, `home`, `plant`...).
- Contexts in which the Fac and Gpe entities are separated by a preposition. In this case:
  - The distance between the two words is two chunks—i.e., there is one chunk between them (`dist=2ch`).
  - The distance between the two entities is two tokens—i.e., there is one token between them (`dist=2tk`).
  - The chunk in between is a prepositional phrase (`ch/before:right:1/type=PP`).
  - The head of this chunk is a preposition (`ch/before:right:1/head-tag=IN`). The strongest indicator of relatedness among them is *in*, but other ones like *near*, *outside*, *of*... are also possible (`ch/before:right:1/head-lemma=in`, `near`, `outside`...).

Concerning the relevant features with negative expected score offset, most of them express properties of chunks at a large distance, and are hence indicators of a long distance between the two entities. Therefore, the model is penalizing pairs in which the entities are far a part—matching the empirical evidence that relations are usually expressed using short-distance constructions (as mentioned in Section 4.3.1).

Moreover, for this particular entity type pair, the feature `ch/left/type=NP` is also a strong indicator of non-relatedness. Given that entities tend to occur within noun phrases, this feature

	Feature	E_f^1	E_f^0	Δ_f
+	ch/common/type=NP	161627.0	83451.6	78175.6
+	dist=0ch	161627.0	83451.6	78175.6
+	ch/common/head-tag=FAC	161435.0	85035.9	76399.2
+	ch/before:right:1/head-lemma=in	155162.0	82405.1	72756.7
+	ch/before:right:1/head-tag=IN	146606.0	75623.9	70982.0
+	ch/before:right:1/type=PP	144761.0	75747.1	69014.2
+	dist=1tk	154067.0	87335.1	66731.5
+	dist=2tk	145362.0	78827.7	66534.5
+	ch/common/head-lemma=hospital	153242.0	94010.6	59231.8
+	ch/common/head-lemma=home	150244.0	94034.4	56209.6
+	ch/common/head-lemma=plant	150089.0	94083.3	56005.6
+	ch/common/head-lemma=prison	149570.0	94153.7	55416.1
+	ch/common/head-tag=PER	148961.0	94181.4	54779.3
+	ch/common/head-tag=GPE	146703.0	94218.0	52485.1
+	ch/before:right:1/head-lemma=near	145274.0	93890.5	51383.8
+	ch/common/head-lemma=embassy	144723.0	94197.2	50525.9
+	ch/common/head-lemma=jail	144607.0	94271.0	50335.7
+	ch/common/head-lemma=building	144033.0	94284.0	49749.2
+	dist=2ch	126251.0	76782.1	49469.0
+	ch/common/head-lemma=base	143557.0	94249.7	49306.8
+	ch/common/head-tag=NN	142912.0	94267.4	48645.1
+	ch/common/head-lemma=station	142789.0	94290.8	48498.1
+	ch/common/head-lemma=airport	141110.0	94314.2	46796.2
+	ch/common/head-lemma=street	140978.0	94311.5	46666.2
+	ch/common/head-lemma=center	140819.0	94310.4	46508.4
+	ch/common/head-lemma=facility	140226.0	94308.8	45917.5
+	ch/common/head-lemma=headquarters	138602.0	94283.2	44318.5
+	ch/before:right:1/head-lemma=outside	137507.0	94309.4	43197.8
+	ch/before:right:1/head-lemma=of	136388.0	93392.6	42995.4
+	ch/common/head-lemma=hotel	137133.0	94360.9	42772.1
	...			
-	ch/left/type=NP	83768.7	153431.0	-69662.6
-	ch/before:right:9/head-lemma=mile	25301.7	94670.2	-69368.4
-	ch/after:left:4/head-lemma=mile	26666.2	94685.5	-68019.3
-	ch/before:right:9/head-tag=NNS	27376.3	94711.4	-67335.1
-	ch/before:right:10/type=ADVP	27973.4	94683.0	-66709.6
-	ch/before:right:10/head-tag=RB	28106.0	94679.1	-66573.1
-	ch/after:left:6/head-lemma=of	28758.2	94768.4	-66010.2
-	ch/before:right:11/head-lemma=of	28855.5	94777.2	-65921.7
-	ch/after:left:5/head-tag=RB	29053.4	94687.8	-65634.5
-	ch/before:right:6/head-lemma=in	30355.3	94871.3	-64516.0
-	ch/before:right:8/head-lemma=,	30760.4	95012.4	-64252.0
-	ch/before:right:8/head-tag=,	30760.4	95012.4	-64252.0
-	ch/after:left:5/type=ADVP	30784.9	94718.1	-63933.2
-	ch/before:right:7/head-tag=GPE	31557.8	95172.8	-63615.0
-	ch/before:right:12/head-tag=GPE	31145.9	94682.7	-63536.9
	...			

Table 6.9: Expected score offsets for the most relevant features
(Fac-Gpe pair, Prob algorithm, NSiz function, `w:t+c:tl` features, Best threshold)

(**that/FAC**$_1$)$_{\text{NP}}$ ('s/VBZ)$_{\text{VP}}$ (**new_york/GPE**$_{\text{A}}$)$_{\text{NP}}$
('s/POS **laguardia_airport/FAC**$_2$)$_{\text{NP}}$ ./.

(a)

(**Israeli/GPE**$_{\text{A}}$ combat/NN helicopters/VEH)$_{\text{NP}}$
(are/VBP reported/VBN to/TO have/VB hit/VBN)$_{\text{VP}}$
(a/DT **hotel_and_casino/FAC**$_2$)$_{\text{NP}}$ (in/IN)$_{\text{PP}}$ (**Jericho/GPE**$_{\text{B}}$)$_{\text{NP}}$
and/CC (a/DT **building/FAC**$_2$)$_{\text{NP}}$ (of/IN)$_{\text{PP}}$
(the/DT **Palestinian_authority/GPE**$_{\text{C}}$)$_{\text{NP}}$ ./.

(b)

(An/DT explosion/NN)$_{\text{NP}}$ (struck/VBD)$_{\text{VP}}$ (the/DT Cole/NNP)$_{\text{NP}}$,/,
(as/IN)$_{\text{SBAR}}$ (it/PRP)$_{\text{NP}}$ (refueled/VBD)$_{\text{VP}}$ (in/IN)$_{\text{PP}}$
(the/DT **Yemeni/GPE**$_{\text{A}}$ **port/FAC**$_1$)$_{\text{NP}}$ (of/IN)$_{\text{PP}}$ (**Aden/GPE**$_{\text{B}}$)$_{\text{NP}}$ ./.

(c)

(The/DT authorities/PER)$_{\text{NP}}$ (have/VBP started/VBN building/VBG)$_{\text{VP}}$
(sandbag/NN **walls/FAC**$_1$)$_{\text{NP}}$ (to/TO protect/VB)$_{\text{VP}}$
(the/DT main/JJ **road/FAC**$_2$)$_{\text{NP}}$ (into/IN)$_{\text{PP}}$ (**Vietnam/GPE**$_{\text{A}}$)$_{\text{NP}}$
('s/POS commercial/JJ **capital/GPE**$_{\text{B}}$)$_{\text{NP}}$,/,
(**Ho_Chi_Min_city/GPE**$_{\text{C}}$)$_{\text{NP}}$ ./.

(d)

(in/IN)$_{\text{PP}}$ (many/JJ **cities/GPE**$_{\text{A}}$)$_{\text{NP}}$,/, (angry/JJ crowds/PER)$_{\text{NP}}$
(roam/VBP)$_{\text{VP}}$ (the/DT **streets/FAC**$_1$)$_{\text{NP}}$ ./.

(e)

Figure 6.14: Sample sentences from the ACE corpus

will be active in most of the cases where the two entities are not embedded in the same chunk (where a `ch/common/type=NP` would be active instead). This seems thus to reinforce the previous conclusion that Fac-Gpe relations are often expressed by means of a nominal complement, where both entities are contained inside the same chunk.

The analysis of individual feature relevance, however, can only uncover the strongest patterns of relatedness and non-relatedness. We must keep in mind that the model is complex, and the interactions between the features will lead to decisions far beyond the reach of these simple rules-of-thumb.

6.3.4.2 Sample Extractions

Figure 6.14 contains some sample sentences from the ACE corpus, drawn to illustrate the behaviour of the model learnt by Ewocs.

In the first sentence, (a), the model correctly predicts the relation between **new_york** and **laguardia_airport**, expressed by means of a possessive construction. Moreover, it judges **that** and **new_york** as non-related, which also matches the ACE annotation (even if, in this case, the existence of a relation may be subject to discussion).

In Sentence (b), there is a number of Fac and Gpe entities, of which only **hotel_and_casino** and **Jericho**, and **building** and **Palestinian_authority**, are related. The model classifies all the pairs correctly. In particular, it classifies **building** and **Jericho** as non-related—despite the fact that a strong indicator of relatedness such as feature `dist=2ch` is active. The model is hence able to successfully integrate the evidence supporting and rejecting relatedness into complex patterns.

On the flipside, in the case of Sentence (c) the outcome of the model differs from the ACE gold annotation. The sentence contains two Gpe entities, and one Fac entity, which the system classifies as related to the two of them. However, the relation existing between **port** and **Yemeni** was not annotated by the ACE annotators, and the pair counts as a false positive instead of a true one—an error which hurts the estimation of precision (see Section 4.5.1). Manual inspection shows that this kind of annotation errors are more frequent than we expected, and suggest that the actual precision figures for the methods might be higher than reported in our evaluation.

However, there is obviously a large number of actual misclassifications performed by the model. In Sentence (d), for instance, the system is unable to detect the relation between **road** and **capital** or **Ho_Chi_Min_city**—but finds one between **road** and **Vietnam**, which is not annotated as such. In this case, the fact that **road** and **Vietnam** are separated by a single preposition leads the model to an error, as it ignores the fact that the Gpe is followed by a possessive mark. For the other two cases, we believe the reason for the false negatives lies in the distance between **capital** and **Ho_Chi_Min_city** and **road**, together with the presence of other entities in between. The incorporation of features derived from the sentence parse tree could provide the additional information required to recover the relations present in these contexts.

Finally, in the case shown in Sentence (e), the relation between **cities** and **streets** is not detected by the model either. This time, in addition to the distance between the two entities, the fact that the prepositional phrase has been shifted to the beginning of the sentence makes it harder for the model to recognize the existent relation. Similarly to the previous sentence, in these and similar cases deep parsing information may bring an improvement—even if it would clearly pose a number of challenges of its own, too.

6.4 Conclusions

This chapter has presented our scheme for joint clustering and pattern learning combination. The proposed framework reduces pattern acquisition to the compilation of all entity pairs which co-occur in the same sentence across an unannotated corpus, and the minority clustering of the obtained context matrix.

To effectively put this scheme into action, firstly, the formalism previously used for sequential and collaborative clustering and learning combination, based on binary feature templates able to capture linguistic traits of the entity pair context, has been revisited to incorporate new sources of information to the potential detection patterns. Secondly, a number of weak clustering algorithms suitable for this binary feature vectors have been proposed, to be plugged into the Ewocs minority clustering algorithm developed in the previous chapter. Thirdly, two heuristics to reduce biases present in the scoring scheme of Ewocs have been devised.

In order to validate this novel approach, these components have been evaluated on an actual relation detection task, comparing them to the existing POS-sequence-based unsupervised system of Hassan et al. (2006) and a distance-based baseline.

The results of the evaluation have shown the superiority of Ewocs-based approaches. Specifically, the normalization of cluster scores by clustering cardinality within the Ewocs scoring process, the incorporation of chunk head POS-tags and lemmas into the patterns, and the usage of a naive-Bayes weak clustering algorithm based on Bernoulli distributions have been particularly successful, and made their results rise in terms of precision, recall and F1 of the detected relations up to levels which significantly improve those obtained by competing approaches. The usage of other feature sets, as well as that of margin-based weak clustering algorithms, has produced results that, despite not matching the performance of the most efficacious ones, also exceed the compared alternatives.

The approaches have been proven to be portable across texts of heterogeneous nature, and robust with respect to the setting of their internal parameters. Nevertheless, the larger computational requirements, in terms of time and space, of the better-performing probabilistic algorithm can make its choice unsuitable on very large textual collections, and opens the door to the use of margin-based weak clusterers—a choice which trades efficiency for a minor decrease in extraction performance.

At the light of these results, we believe that the proposed method for unsupervised learning of relation detection patterns can be considered a powerful alternative to existing alternatives, given its simplicity, efficiency, flexibility, non-supervision and competitive performance.

7 Conclusions

And this is what you waited for
But under lights, we're all unsure
So tell me
What would make you feel better?

LCD Soundsystem
Home

This chapter presents our conclusions and final thoughts at the end of the thesis. This recapitulation also gives us the chance to put into relief the main contributions of our work, and to sketch possible lines of future work.

Section 7.1 highlights the main contributions our work makes to the areas of clustering and information extraction. Section 7.2 draws a reduced set of final conclusions from all the work in our thesis. Finally, Section 7.3 contains our thoughts on the lines of research that could be followed to further explore areas that still remain open after our work.

In these pages, we have explored the task of unsupervised learning of relation detection patterns using clustering techniques. We have conducted research on the ways in which this combination can be accomplished, and we have also developed, adapted and compared clustering methods to suit our specific needs. Along the way, we have proposed a completely novel approach for minority clustering, based on ensemble methods.

7.1 Contributions of this Thesis

The research in this thesis thus spans over several areas within the fields of NLP and ML, and we believe that a number of distinguishable contributions are contained in our work. We want to highlight a small number of them, listing them below in what we consider their order of decreasing relevance:

- We have developed a novel unsupervised approach for learning of IE patterns using a minority clustering algorithm (Chapter 6). The approach not only presents a much lower degree of supervision than many other existing alternatives, but it is also more flexible and allows easier incorporation of additional linguistic features into the patterns. We have implemented a learning method based on this approach, and evaluated the produced patterns on the ACE relation detection task. The results of the evaluation have shown that the proposed approach outperforms comparable alternatives in the state of the art. Moreover, the experimentation has also provided insights into the influence of diverse feature sets and components of the minority clustering algorithm on the extraction pattern performance—as well as into the variations in performance experimented by the patterns across documents from different sources and nature.

- We have developed Ewocs, a novel minority clustering algorithm, and the first—to the best of our knowledge—to use ensemble methods for the task (Chapter 5). The algorithm consists of repeated application of an inner weak clustering algorithm and a cluster scoring scheme, and has been derived from the theoretical analysis of the distribution of the obtained scores. It is hence a statistically sound algorithm, under a set of conditions which we have found to be easily satisfied in practice. The algorithm contains a number of components (inner clustering algorithm, scoring scheme, threshold detection), for which we have considered different alternatives. All of them have been implemented and evaluated over a collection of geometrical datasets, in which the comparison to other approaches has been favourable to Ewocs. Moreover, the success of our pattern learning approach, which uses Ewocs at its core, must be regarded not only as a proof of the validity of the joint clustering and pattern learning strategy, but also of the effectiveness of the Ewocs algorithm itself.

- We have developed unsupervised approaches for learning of IE patterns using sequential and collaborative schemes for combination with document clustering (Chapter 4). The correctness of the approach has been studied using a double evaluation. In a first indirect evaluation on text categorization, the acquired patterns have been found to perform competitively with respect to those learned using manual sets of seeds. However, in the second direct evaluation on relation detection, the results have shown the inability of the framework to detect generic and transverse relations, such as the ones in the ACE evaluations. The low mutual information between the distributions of clusters and relation types has been pointed to as an explanation for the poor performance of the proposed approaches. The results have also raised doubts on the suitability of certain indirect text categorization evaluations themselves.

- We have performed an empirical comparison of a number of unsupervised ensemble approaches for the task of document clustering (Chapter 3). Two ensemble generation strategies and six clustering combination algorithms have been compared, together with individual methods, across a collection of real-world datasets. The comparison has shown the superiority of combination approaches over individual clustering algorithms; and of a massive generation strategy, based on randomization of a less informed algorithm, over a smaller ensemble of stronger individual clusterers.

- We have proposed a number of weak clustering algorithms: RSplit (Section 5.4.1), Rbc (Section 5.4.2) and RSvc (Section 6.2.2.2), the last two of them based on margin- and kernel-method theory. Their utility has been empirically assessed by their use within the Ewocs algorithm (Sections 5.6 and 6.3)—but they are open to usage in general weak clustering settings.

- We have devised an information-theoretical unsupervised clustering method, as an adaptation of a geometrical approach (Section 3.4.1.2). The method uses algorithms, measures and criteria coming from the field of information theory. Its performance has been compared to other individual and ensemble unsupervised clustering methods (Section 3.5).

7.2 General Conclusions

Being now at the terminus of our work, a number of general conclusions can be drawn from the results of our research. We want to highlight only three of them, as they are the strongest trends that we have observed along all our work.

We regard the leitmotif of the thesis itself—namely, that it is indeed possible to enhance the process of pattern learning with clustering techniques and reduce its elements of supervision—as the first and most significant one. In particular, using the Ewocs minority clustering algorithm we have obtained a virtually unsupervised method to learn patterns which detect ACE-style relations between entity mentions. As said, we consider this to be the most significant conclusion of our research, as it is the one that validates the whole thesis, and which justifies all work done. Even if other works had combined clustering and IE pattern acquisition, our approach has a number of distinguishing traits—such as the reduction of the task to a minority clustering problem, its flexibility in the incorporation of new features and the reduced supervision requirements—which set it apart from the rest.

Secondly, a trend which has recurrently occurred in our results is the superiority of ensemble methods with respect to individual approaches. We have observed such superiority in unsupervised document clustering (MAJOR method in Section 3.5), in collaborative pattern learning (MAJOR seeds in Section 4.5), in minority clustering (EWOCS algorithm in Section 5.6) and in joint pattern learning (Section 6.3). Even if we are only one more among the increasing number of works which praise combination approaches in both supervised and unsupervised learning settings, we believe the repetition of the trend across a number of different tasks is an interesting conclusion—which should reinforce the confidence in such approaches for all kind of ML problems.

Finally, a last pattern of behaviour consistently observed across our experiments is the prevalence of the ugly duckling theorem—and the problems inherent to providing learners with more information within unsupervised settings. Even if the issue is shared between unsupervised and supervised learning settings, in unsupervised learning tasks the lack of class information implies that drawing a line between useful and non-useful features can be hard—and makes it possible for the structure that we expect to uncover to end up buried in irrelevant information. In our case, this phenomenon lies behind the decrease of performance experimented by more complex feature sets with respect to simpler counterparts (such as `w:t+c:l` in Section 4.5, or `w:tl` in Section 6.3.3.2).

7.3 Future Work

Even if the writing of their PhD thesis is a major undertaking for any graduate student, it is also true that any work of research, even if it closes pending questions, always leaves new ones open. This thesis is no exception, and a number of ideas have not been thoroughly explored—including some which have only been scratched at the surface. This section tries to collect such possible future lines of research, grouping them by the chapter in which the work related to them is exposed.

Clustering Regarding our work on unsupervised clustering and clustering ensembles, it would be interesting to extend the comparison in Section 3.5 to consider more methods, both individual and ensemble-based. In particular, adding or replacing methods within the MINOR ensemble generation strategy may improve its results. Regarding the MAJOR strategy, the influence of the k_{max} parameter on the final output, even if small, is clearly a drawback of the approach (Section 3.5.3.1). Alternative ensemble generation and/or combination algorithms need to be explored, which be even less sensitive to the tuning of this and other parameters.

Finally, it is clear that the proposed INFO method is not as competitive as its GEO counterpart. However, the good performance of IT-based clustering algorithms has been proved in countless works, so finding a suitable implementation of the hybrid unsupervised clustering method is an open problem by itself.

Collaborative Learning As seen in Section 4.5.3, the performance of the sequential and collaborative approaches for pattern learning has been one of the biggest disappointments in this thesis. Nevertheless, even if we have been unable to detect ACE-style relations using these combination frameworks, we believe they may still be useful to detect domain-specific relations from document collections containing well distinguished categories. Experimentation with different relation types is clearly needed.

On the other hand, the introduction of the binary feature conjunction pattern formalism (Section 4.3.1.2) has required the separation of the pattern candidate generation and pattern selection steps for computational reasons. Given the poor results of the overall approach, it has not been possible to assess the impact of this decision. An evaluation of the fraction of interesting patterns that are missed because of not exceeding the frequency threshold can be of interest. Moreover, the development of a smart strategy to explore the pattern space during the pattern learning stage is by itself an algorithmic challenge—but might bring unexpected improvements on the quality of the obtained pattern base.

Minority Clustering Even if our treatment of the Ewocs algorithm has been quite thorough, a number of issues concerning components of the algorithm remain open. The first and most obvious one is the improvement of the threshold detection algorithm. As mentioned in Section 5.6.4.3, there is still a gap between the threshold giving the maximum F1 score and the criteria that provide some level of supervision—and another one between these partially supervised criteria and the completely unsupervised ones. Similarly, in the application of Ewocs to pattern learning, inaccuracies in the threshold detection cause a significant loss in detection power (Section 6.3.3.4). Improving the detection of the threshold can thus have a considerable effect on the performance of Ewocs-based systems. Moreover, the presented detection methods are mostly heuristic in nature—research on the theoretical basis of the foreground-background separation is required, and may provide better criteria.

Concerning the evaluation of Ewocs, in this work we have only applied the method to synthetic data (Section 5.6) and to the linguistic data used within our pattern learning approach (Section 6.3). Given that Ewocs is devised as a generic algorithm for minority clustering, it would be interesting to test it in data coming from different areas.

Last, even if the procedure used to tune the Gaussian kernel within the Rbc algorithm (Section 5.4.2.3) has allowed us to obtain close-to-optimal performance on synthetic data, it is nevertheless a costly procedure. The development of a cheaper alternative, suitable to be used in large datasets, may be of interest not only for usage within Ewocs, but for any fuzzy clustering algorithm in general.

Joint Learning Finally, regarding the joint clustering and pattern learning approach, we believe much remains to be explored concerning the introduction of more kinds of linguistic information into the patterns. Parsing information, for instance, has been proven useful in a considerable number of relation extraction approaches. Moreover, one of the advantages of Ewocs is that it is not restricted to flat feature vectors. By using suitable kernel functions, alternative context representations, such as sequences or trees, could be employed.

Nevertheless, we believe that the most promising future line of research, for both this joint approach and the collaborative and sequential ones, is to extend the learning process to allow for full relation extraction instead of only detection—i.e., to include devices so as to classify the obtained relations into a number of classes, predefined or not. In particular, classification could allow the acquisition of patterns corresponding to a predefined scenario of extraction, allowing a system built using an otherwise unsupervised approach to take part, for instance, in KBP-style evaluations. Additionally, we also envision the incorporation of biases for the features taking part in the patterns—favouring, for instance, one particular type of construction (nominal, verbal...) above the others.

The possibility of incorporating an extraction scenario, or feature selectional preferences, as biases for the pattern learning process would require the extension of the existing framework to allow both completely unsupervised and slightly supervised learning. Such a unified framework would be of large interest by itself.

Appendices

A
Mathematical Background

This appendix contains the definitions of common mathematical concepts that we have omitted in the body of the thesis for fluidity of the exposition. We expect most readers to be familiar with them, but nonetheless decided to incorporate them for reference. We have tried to include, together with the definition, an authoritative source on the topic, often the very work in which the concept was first defined.

The concepts are grouped in three main areas. Section A.1 contains definitions of concepts from regular set theory, whereas Section A.2 deals with fuzzy sets, and Section A.3 is concerned with Information Theory.

A.1 Set Theory

Definition A.1 (Partition)
A ***partition*** *of a set $\mathcal{X}$ is a family of sets $\Pi = \{\pi_1 \ldots \pi_k\}$ such that*

- *The sets π_c are non-empty.*
$$\forall \pi_c \in \Pi : \; \pi_c \neq \varnothing$$
- *All sets are disjoint.*
$$\forall \pi_c \neq \pi_{c'} \in \Pi : \; \pi_c \cap \pi_{c'} = \varnothing$$
- *The union of all sets π_c is the total set $\mathcal{X}$.*
$$\bigcup_{\pi_c \in \Pi} \pi_c = \mathcal{X}$$

A.2 Fuzzy Set Theory

Definition A.2 (Fuzzy set)
A ***fuzzy set*** *over an ordinary set $\mathcal{X}$ is a pair $\tilde{\mathcal{X}} = (\mathcal{X}, f_{\tilde{X}})$, where $f_{\tilde{X}} : \mathcal{X} \to [0,1]$ is the* ***membership function*** *(or* ***characteristic function****) of $\tilde{\mathcal{X}}$. For $x_i \in \mathcal{X}$, $f_{\tilde{X}}(x)$ expresses the* ***grade*** *of membership of x_i to $\tilde{\mathcal{X}}$, and will often be denoted as $\mathrm{grade}(x_i, \tilde{\mathcal{X}})$*

(Zadeh, 1965) □

Definition A.3 (Fuzzy c-partition)
*A **fuzzy c-partition** (or **fuzzy pseudopartition**) of an ordinary set $\mathcal{X}$ is a family of fuzzy sets $\Pi = \{\pi_1 \dots \pi_k\}$ over $\mathcal{X}$ such that*

$$\forall x \in \mathcal{X} : \sum_{\pi_c \in \Pi} f_{\pi_c}(x) = 1$$
$$\forall \pi_c \in \Pi : \; 0 < \sum_{x \in \mathcal{X}} f_{\pi_c}(x) < \|\mathcal{X}\|$$

(Bezdek, 1981; Klir and Yuan, 1995) □

A.3 Information Theory

Definition A.4 (Entropy)
*The **entropy** of a discrete random variable X, following probability distribution $p(x)$ over set $\mathcal{X}$, is defined as:*

$$\mathrm{H}(X) = -\sum_{x \in \mathcal{X}} p(x) \cdot \log p(x)$$

(Shannon, 1948) □

Definition A.5 (Mutual Information)
*The **mutual information** between two discrete random variables X and Y, following a joint probability distribution $p(x,y)$ over sets $\mathcal{X}$ and $\mathcal{Y}$, is defined as:*

$$\mathrm{I}(X\,;Y) = \sum_{\substack{x \in \mathcal{X} \\ y \in \mathcal{Y}}} p(x,y) \cdot \log \frac{p(x,y)}{p(x) \cdot p(y)}$$

(Shannon, 1948) □

Definition A.6 (Cross-Entropy)
*The **cross-entropy** between two discrete probability distributions, $p(x)$ and $q(x)$, of the same random variable X over set $\mathcal{X}$, is defined as:*

$$\mathrm{H}^{\times}(p,q) = \sum_{x \in \mathcal{X}} p(x) \cdot \log q(x)$$

(Kullback and Leibler, 1951) □

Definition A.7 (Kullback-Leibler Divergence)
*The **Kullback-Leibler divergence** between two discrete probability distributions, $p(x)$ and $q(x)$, of the same random variable X over set $\mathcal{X}$, is defined as:*

$$\mathrm{KL}(p \mid q) = \sum_{x \in \mathcal{X}} p(x) \cdot \log \frac{p(x)}{q(x)}$$

(Kullback and Leibler, 1951) □

Definition A.8 (Jensen-Shannon Divergence)
*The **Jensen-Shannon divergence** between two discrete probability distributions $p(x)$ and $q(x)$, of the same random variable X over set $\mathcal{X}$, is defined as the average of the Kullback-Leibler divergence between each one of the distributions and their average, $m(x)$:*

$$\mathrm{JS}(p \parallel q) = \frac{1}{2} \cdot \mathrm{KL}(p \mid m) + \frac{1}{2} \cdot \mathrm{KL}(q \mid m)$$

where:

$$m(x) = \frac{1}{2} \cdot (p(x) + q(x))$$

(Lin, 1991) □

B

Proofs

This appendix contains the proofs of secondary propositions which we have not judged relevant enough to the discussion so as to include them in the main text—but which are not trivial and hence require a formal proof of their validity.

Each section below contains one of the proofs that have been deferred.

Proof of Proposition in Definition 3.5

In a subsumption tree Ψ over dataset $\mathcal{X}$, of cardinality $n = |\mathcal{X}|$, the root ψ_{2n-1} is equal to the complete dataset $\mathcal{X}$.

Proof The following definition will be needed for all proofs in the section:

> **Definition B.1 (Sequence prefix)**
> *Given a sequence $\Psi = (\psi_1 \ldots \psi_d)$ of length d, and $p \in \{1 \ldots d\}$, the **prefix** of length p of sequence Ψ is the subsequence*
>
> $$\Psi^p = (\psi_1 \ldots \psi_p)$$
>
> Remark Being a sequence of sets, the prefixes of a subsumption tree are its prefixes regarded as a sequence.

The proof uses the two propositions, which state properties of the subsumption tree prefixes.

> **Proposition B.2**
> *The number of non-subsumed nodes within the subsumption tree prefix Ψ^p of length p, with $p \in \{n \ldots 2n-1\}$, is $2n - l$.*
>
> Proof By induction on p:
>
> - If $p = n$, Ψ^n only contains the leaves $\psi_i = \{x_i\}$, for $x_i \in \mathcal{X}$. All the elements in $\mathcal{X}$ being distinct, none of the leaves subsumes another, so the number of non-subsumed nodes is $n = 2n - p = 2n - n$.
> - Let us assume that $p > n$ and the property is true for $p - 1$. Therefore, the number of non-subsumed nodes in Ψ^{p-1} is $2n - (p - 1) = 2n - p + 1$.

Let us consider node ψ_p. As $p > n$, ψ_p is the union of two preceding nodes ψ_a and ψ_b, with $a, b < p$. Hence, $\psi_a, \psi_b \in \Psi^{p-1}$.

Moreover, given that ψ_p subsumes ψ_a and ψ_b, and that all sets are subsumed by exactly one succeeding set, no set in Ψ^{p-1} can subsume ψ_a or ψ_b, and hence they are non-subsumed nodes in Ψ^{p-1}.

With the incorporation of ψ_p to Ψ^{p-1}, nodes ψ_a and ψ_b become subsumed in Ψ^p. But ψ_p cannot be subsumed by any succeeding one, being the last in the subsequence. Hence, the number of non-subsumed nodes is one less than in the previous prefix, becoming $(2n - p + 1) - 1 = 2n - p$. ■

Proposition B.3
The union of all non-subsumed nodes within the subsumption tree prefix Ψ^p of length p, with $p \in \{n \dots 2n-1\}$, is $\mathcal{X}$.

Proof By induction on p:

- If $p = n$, Ψ^n only contains the leaves $\psi_i = \{x_i\}$, for $x_i \in \mathcal{X}$. The union of all leaves is hence
$$\bigcup_{\psi_i \in \Psi^n} \psi_i = \bigcup_{x_i \in \mathcal{X}} \{x_i\} = \mathcal{X}$$
- Let us assume that $p > n$ and the property is true for $l - 1$. Therefore, the union of all non-subsumed nodes in Ψ^{p-1} is $\mathcal{X}$.

  Similarly to the previous proof, node ψ_p is the union of two preceding nodes ψ_a and ψ_b, which are non-subsumed within Ψ^{p-1}, but become so within Ψ^p. ψ_p is, in turn, non-subsumed within Ψ^p.

  Hence,
$$\begin{aligned} &\bigcup\{\psi_i \mid \psi_i \in \Psi^p \wedge \neg subsumed(\psi_i\,;\Psi^p)\} = \\ &\quad= \quad \psi_p \cup \left(\bigcup\{\psi_i \mid \psi_i \in \Psi^{p-1} \wedge \neg subsumed(\psi_i\,;\Psi^p)\}\right) \\ &\quad= \quad \psi_p \cup \left(\left(\bigcup\{\psi_i \mid \psi_i \in \Psi^{p-1} \wedge \neg subsumed(\psi_i\,;\Psi^{p-1})\}\right) \setminus \psi_a \setminus \psi_b\right) \\ &\quad= \quad (\psi_a \cup \psi_b) \cup \left(\left(\bigcup\{\psi_i \mid \psi_i \in \Psi^{p-1} \wedge \neg subsumed(\psi_i\,;\Psi^{p-1})\}\right) \setminus (\psi_a \cup \psi_b)\right) \\ &\quad= \quad \bigcup\{\psi_i \mid \psi_i \in \Psi^{p-1} \wedge \neg subsumed(\psi_i\,;\Psi^{p-1})\} \end{aligned}$$
  which equals $\mathcal{X}$ by the induction hypothesis. ■

Using these two propositions, the proof of the proposition is immediate. The whole dendrogram can be regarded as its own prefix of length $2n - 1 \in \{n \dots 2n - 1\}$. Within it, by Proposition B.2 there is only $2n - (2n - 1) = 1$ non-subsumed node, which must necessarily be the root ψ_{2n-1}. Given that, by Proposition B.3, the union of all non-subsumed nodes within the prefix is $\mathcal{X}$, the root $\psi_{2n-1} = \mathcal{X}$. ■

Proof of Proposition 3.8

A cut of tree Ψ over dataset $\mathcal{X}$ at any valid level l is a partition of $\mathcal{X}$.

Proof The proof uses the following propositions:

Proposition B.4
All nodes within the prefix of length p of a subsumption tree Ψ are non-empty.

Proof By induction on p:

- For $p \leq n$, the prefix of length p of the tree will only contain leaves which, by construction, are singletons containing the objects in set $\mathcal{X}$. They are hence non-empty.

- Let us assume $n < p \leq 2n - 1$ and the proposition holds for $p - 1$. Therefore, all nodes in Ψ^{p-1} are non-empty.

  Similarly to the previous proofs, node ψ_p is the union of two preceding nodes ψ_a and ψ_b, which are in Ψ^{p-1}, and are hence non-empty. Being the union of two non-empty nodes, ψ_p cannot be empty.

  Therefore, the prefix of length p does not contain any empty node either. ■

Corollary B.5
All nodes in a subsumption tree are non-empty.

Proof The whole tree is a prefix of itself, and, by previous proposition, prefixes of a tree do not contain empty nodes. ■

Proposition B.6
All non-subsumed nodes within the prefix of length p of a subsumption tree Ψ are disjoint.

Proof By induction on p:

- For $p \leq n$, the prefix of length p of the tree will only contain leaves which, by construction, are singletons containing the objects in set $\mathcal{X}$. All the elements in $\mathcal{X}$ being distinct, they are hence all disjoint.

- Let us assume $n < p \leq 2n - 1$ and the proposition holds for $p - 1$. Therefore, all non-subsumed nodes in Ψ^{p-1} are disjoint.

  Similarly to the previous proofs, node ψ_p is the union of two preceding nodes ψ_a and ψ_b, which are non-subsumed within Ψ^{p-1}, but become so within Ψ^p. ψ_p is, in turn, non-subsumed within Ψ^p.

  The only potential candidate for intersection with the other non-subsumed nodes in the prefix is ψ_p. However, ψ_p is the union of two nodes ψ_a and ψ_b, which were disjoint to all other nodes in Ψ^{p-1}. Hence, ψ_p is also be disjoint to them. The nodes with which ψ_p intersects are ψ_a and ψ_b, which are subsumed, or those which were subsumed in the first place by these nodes.

  Therefore, ψ_p is disjoint with all other non-subsumed nodes in Ψ^p. The rest of non-subsumed nodes remaining the same as in Ψ^{p-1}, all non-subsumed nodes in Ψ^p are disjoint. ■

Proposition B.7
The cut of tree Ψ at a level $l \in \{1 \dots n\}$ is the set of non-subsumed nodes within the prefix Ψ^p, with $p = 2n - l$.

Proof By induction on l:

- According to its definition, the cut at level $l = n$ is the set of nodes at level larger or equal than n which are not subsumed by another node at level larger or equal than n. This corresponds to the set of leaves, as only they have the largest level n, and none of them subsumes another.

  On the flipside, the prefix of length $p = 2n - n = n$ also corresponds, by definition, to the set of leaves of the tree. Hence, the two sets are equal.

- Let us assume that $1 \leq l < n$ and the property is true for $l + 1$. Therefore, the cut of tree Ψ at a level $l + 1$ is the set of non-subsumed nodes within the prefix Ψ^{2n-l-1} of length $2n - l - 1$.

  Consider now the cut for level l. It will contain the nodes at level larger or equal than l, which are not subsumed by another level larger or equal than l. In particular, it will contain all nodes in the cut for level $l+1$, plus the nodes at level l, minus the nodes in the previous cut which are subsumed by the nodes at level l.

In particular, as $l < n$, from the definition of node level, it follows that only one branch in a tree may have level l, and that is node ψ_{2n-l}. By appending this node to prefix Ψ^{2n-l-1}, we obtain prefix Ψ^{2n-l}—and the nodes that are subsumed by ψ_{2n-l} are the only ones in Ψ^{2n-l-1} which are subsumed by nodes at level l. Hence, the set of non-subsumed nodes within Ψ^{2n-l} equals the cut of tree Ψ at level l. ■

The proof now is reduced to checking that the set of nodes in $\text{cut}(\Psi, l)$ satisfy the three requirements for being a partition[1]:

- *The sets in the cut are not empty.* Being a subset of the nodes of the tree, which by Corollary B.5 contains no empty nodes, the cut cannot contain any empty nodes.
- *The sets in the cut are disjoint.* Being by Proposition B.7 equal to the set of non-subsumed nodes within a prefix of length $p = 2n - l \in \{n \dots 2n-1\}$ of the tree, all sets are disjoint, according to previous Proposition B.6.
- *The union of all sets in the cut is the total set $\mathcal{X}$.* Being by Proposition B.7 equal to the set of non-subsumed nodes within a prefix of length $p = 2n - l \in \{n \dots 2n-1\}$ of the tree, the union of all sets is equal to the complete set $\mathcal{X}$, according to previous Proposition B.3. ■

Proof of Theorem 6.1

When $k_{max} = 2$, Rbc *using the Gaussian kernel $K_\phi(x,y) = \alpha \cdot e^{-\gamma\|x-y\|^2}$ and* RSvc *using the Gaussian kernel $K'_\phi(x,y) = 2\alpha \cdot e^{-\gamma\|x-y\|^2} = 2K_\phi(x,y)$ are equivalent algorithms.*

Proof If $k_{max} = 2$ the only possible value for the sampled k in both Rbc and RSvc, and it is $k = 2$. In the case of Rbc, this means that two samples $\hat{x}_1, \hat{x}_2 \in \mathcal{X}$ will be sampled, and that the grade of membership of another object x_i to cluster π_1 (resp., π_2) is

$$\text{grade}(x_i, \pi_1) = \frac{e^{-D_\phi(\hat{x}_1, x_i)}}{\sum_{q=1}^{k} e^{-D_\phi(\hat{x}_q, x_i)}} = \frac{e^{-D_\phi(\hat{x}_1, x_i)}}{e^{-D_\phi(\hat{x}_1, x_i)} + e^{-D_\phi(\hat{x}_2, x_i)}}$$

In particular, using a kernel-induced squared Euclidean distance (Equation 5.9)

$$D_\phi(x,y) = K_\phi(x,x) + K_\phi(y,y) - 2K_\phi(x,y)$$

the formula for the grade of membership becomes

$$\begin{aligned}
\text{grade}(x_i, \pi_1) &= \frac{e^{-(K_\phi(\hat{x}_1,\hat{x}_1) + K_\phi(x_i,x_i) - 2K_\phi(\hat{x}_1,x_i))}}{e^{-(K_\phi(\hat{x}_1,\hat{x}_1) + K_\phi(x_i,x_i) - 2K_\phi(\hat{x}_1,x_i))} + e^{-(K_\phi(\hat{x}_2,\hat{x}_2) + K_\phi(x_i,x_i) - 2K_\phi(\hat{x}_2,x_i))}} \\
&= \frac{\frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{K_\phi(\hat{x}_1,\hat{x}_1)} e^{K_\phi(x_i,x_i)}}}{\frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{K_\phi(\hat{x}_1,\hat{x}_1)} e^{K_\phi(x_i,x_i)}} + \frac{e^{2K_\phi(\hat{x}_2,x_i)}}{e^{K_\phi(\hat{x}_2,\hat{x}_2)} e^{K_\phi(x_i,x_i)}}} \\
&= \frac{\frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{K_\phi(\hat{x}_1,\hat{x}_1)}}}{\frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{K_\phi(\hat{x}_1,\hat{x}_1)}} + \frac{e^{2K_\phi(\hat{x}_2,x_i)}}{e^{K_\phi(\hat{x}_2,\hat{x}_2)}}}
\end{aligned}$$

And in the case of a Gaussian kernel, in which $K_\phi(\hat{x}_1, \hat{x}_1) = K_\phi(\hat{x}_2, \hat{x}_2) = \alpha$

$$\text{grade}(x_i, \pi_1) = \frac{\frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{\alpha}}}{\frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{\alpha}} + \frac{e^{2K_\phi(\hat{x}_2,x_i)}}{e^{\alpha}}} = \frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{2K_\phi(\hat{x}_1,x_i)} + e^{2K_\phi(\hat{x}_2,x_i)}} \tag{B.1}$$

[1] See Definition A.1 in Appendix A.1.

Regarding RSvc, for the same two samples $\hat{x}_1, \hat{x}_2$, the reduced SVM problem becomes finding the

$$\hat{T} = \begin{bmatrix} \hat{\tau}_{11} & \hat{\tau}_{12} \\ \hat{\tau}_{21} & \hat{\tau}_{22} \end{bmatrix}$$

such that

$$\hat{T} = \arg\max_{T} -\frac{1}{2} \sum_{i,j=1}^{2} \sum_{c=1}^{2} K'_\phi(\hat{x}_i, \hat{x}_j) \tau_{ic} \tau_{jc} + \beta \sum_{i=1}^{k} \tau_{ii}$$

and subject to

$$\tau_{11} + \tau_{12} = 1 \qquad \tau_{21} + \tau_{22} = 1$$
$$\tau_{11} \leq 1 \qquad \tau_{12} \leq 0 \qquad \tau_{21} \leq 0 \qquad \tau_{22} \leq 1$$

However, in this case there is no need to solve the optimization problem: the constraints imply that $\tau_{12} = 1 - \tau_{11}$, thus $\tau_{12} = 1 - \tau_{11} \leq 0$, and $\tau_{11} \geq 1$—forcing that $\tau_{11} = 1$ and $\tau_{12} = 0$, and, *mutatis mutandis*, $\tau_{22} = 1$ and $\tau_{21} = 0$. Therefore, the only feasible point, and hence also the optimal, is

$$\hat{T} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

The grade of membership of another object x_i to cluster π_1 (resp., π_2) assigned by RSvc will then be

$$\begin{aligned}
\text{grade}(x_i, \pi_1) &= \frac{e^{M'_\phi(x_i,1)}}{\sum_{q=1}^{2} e^{M'_\phi(x_i,q)}} \\
&= \frac{e^{\tau_{11} K'_\phi(\hat{x}_1,x_i) + \tau_{21} K'_\phi(\hat{x}_2,x_i)}}{e^{\tau_{11} K'_\phi(\hat{x}_1,x_i) + \tau_{21} K'_\phi(\hat{x}_2,x_i)} + e^{\tau_{12} K'_\phi(\hat{x}_1,x_i) + \tau_{22} K'_\phi(\hat{x}_2,x_i)}} \\
&= \frac{e^{K'_\phi(\hat{x}_1,x_i)}}{e^{K'_\phi(\hat{x}_1,x_i)} + e^{K'_\phi(\hat{x}_2,x_i)}} \\
&= \frac{e^{2K_\phi(\hat{x}_1,x_i)}}{e^{2K_\phi(\hat{x}_1,x_i)} + e^{2K_\phi(\hat{x}_2,x_i)}}
\end{aligned}$$

which is equal to the grade of membership determined by Rbc (Equation B.1). Therefore, the produced clusterings—and also the two methods—are equivalent. ■

Proof of Proposition 6.3

When using the Prob *clustering algorithm within* Ewocs*, the expected score offset of feature f can be found from the model parameters as:*

$$\Delta_f = \frac{1}{N} \sum_{\Pi_r} \left(\frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot \vartheta_{rcf} \cdot \text{score}(\pi_{rc})}{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot \vartheta_{rcf}} - \frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot (1 - \vartheta_{rcf}) \cdot \text{score}(\pi_{rc})}{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot (1 - \vartheta_{rcf})} \right)$$

Proof Applying the definition of expected score offset (Definition 6.2), we obtain:

$$\begin{aligned}
\Delta_f &= E[s_x^\star \mid x_{xf} = 1] - E[s_x^\star \mid x_{xf} = 0] \\
&= E[\frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_x, \pi_{rc}) \cdot \text{score}(\pi_{rc}) \mid x_{xf} = 1] \\
&\quad - E[\frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} \text{grade}(x_x, \pi_{rc}) \cdot \text{score}(\pi_{rc}) \mid x_{xf} = 0] \\
&= \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} E[\text{grade}(x_x, \pi_{rc}) \cdot \text{score}(\pi_{rc}) \mid x_{xf} = 1] \\
&\quad - \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} E[\text{grade}(x_x, \pi_{rc}) \cdot \text{score}(\pi_{rc}) \mid x_{xf} = 0] \\
&= \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} E[\text{grade}(x_x, \pi_{rc}) \mid x_{xf} = 1] \cdot \text{score}(\pi_{rc}) \\
&\quad - \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} E[\text{grade}(x_x, \pi_{rc}) \mid x_{xf} = 0] \cdot \text{score}(\pi_{rc})
\end{aligned}$$

For the Prob algorithm, we know that grades of membership are identified with *a posteriori* class probabilities (Section 6.2.2.1). Hence, applying the probabilistic model:

$$
\begin{aligned}
E[\mathrm{grade}(x_x, \pi_{rc}) \mid x_{xf} = 1] &= E[p(\pi_{rc} \mid x_x, x_{xf} = 1)] = p(\pi_{rc} \mid x_x, x_{xf} = 1) \\
&= \frac{p(x_{xf} = 1 \mid \pi_{rc})}{p(x_{xf} = 1)} = \frac{\alpha_{rc} \cdot \vartheta_{rcf}}{\sum_{\pi_{rc'}} \alpha_{rc'} \cdot \vartheta_{rc'f}} \\
E[\mathrm{grade}(x_x, \pi_{rc}) \mid x_{xf} = 0] &= E[p(\pi_{rc} \mid x_x, x_{xf} = 0)] = p(\pi_{rc} \mid x_x, x_{xf} = 0) \\
&= \frac{p(x_{xf} = 0 \mid \pi_{rc})}{p(x_{xf} = 1)} = \frac{\alpha_{rc} \cdot (1 - \vartheta_{rcf})}{\sum_{\pi_{rc'}} \alpha_{rc'} \cdot (1 - \vartheta_{rc'f})}
\end{aligned}
$$

Plugging back these quantities into the definition of Δ_f:

$$
\begin{aligned}
\Delta_f &= \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} E[\mathrm{grade}(x_x, \pi_{rc}) \mid x_{xf} = 1] \cdot \mathrm{score}(\pi_{rc}) \\
&\quad - \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} E[\mathrm{grade}(x_x, \pi_{rc}) \mid x_{xf} = 0] \cdot \mathrm{score}(\pi_{rc}) \\
&= \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} \frac{\alpha_{rc} \cdot \vartheta_{rcf}}{\sum_{\pi_{rc'}} \alpha_{rc'} \cdot \vartheta_{rc'f}} \cdot \mathrm{score}(\pi_{rc}) - \frac{1}{N} \sum_{\Pi_r} \sum_{\pi_{rc} \in \Pi_r} \frac{\alpha_{rc} \cdot (1 - \vartheta_{rcf})}{\sum_{\pi_{rc'}} \alpha_{rc'} \cdot (1 - \vartheta_{rc'f})} \cdot \mathrm{score}(\pi_{rc}) \\
&= \frac{1}{N} \sum_{\Pi_r} \frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot \vartheta_{rcf} \cdot \mathrm{score}(\pi_{rc})}{\sum_{\pi_{rc}} \alpha_{rc} \cdot \vartheta_{rcf}} - \frac{1}{N} \sum_{\Pi_r} \frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot (1 - \vartheta_{rcf}) \cdot \mathrm{score}(\pi_{rc})}{\sum_{\pi_{rc}} \alpha_{rc} \cdot (1 - \vartheta_{rcf})} \\
&= \frac{1}{N} \sum_{\Pi_r} \left(\frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot \vartheta_{rcf} \cdot \mathrm{score}(\pi_{rc})}{\sum_{\pi_{rc}} \alpha_{rc} \cdot \vartheta_{rcf}} - \frac{\sum_{\pi_{rc} \in \Pi_r} \alpha_{rc} \cdot (1 - \vartheta_{rcf}) \cdot \mathrm{score}(\pi_{rc})}{\sum_{\pi_{rc}} \alpha_{rc} \cdot (1 - \vartheta_{rcf})} \right)
\end{aligned}
$$

as we wanted to prove. ■

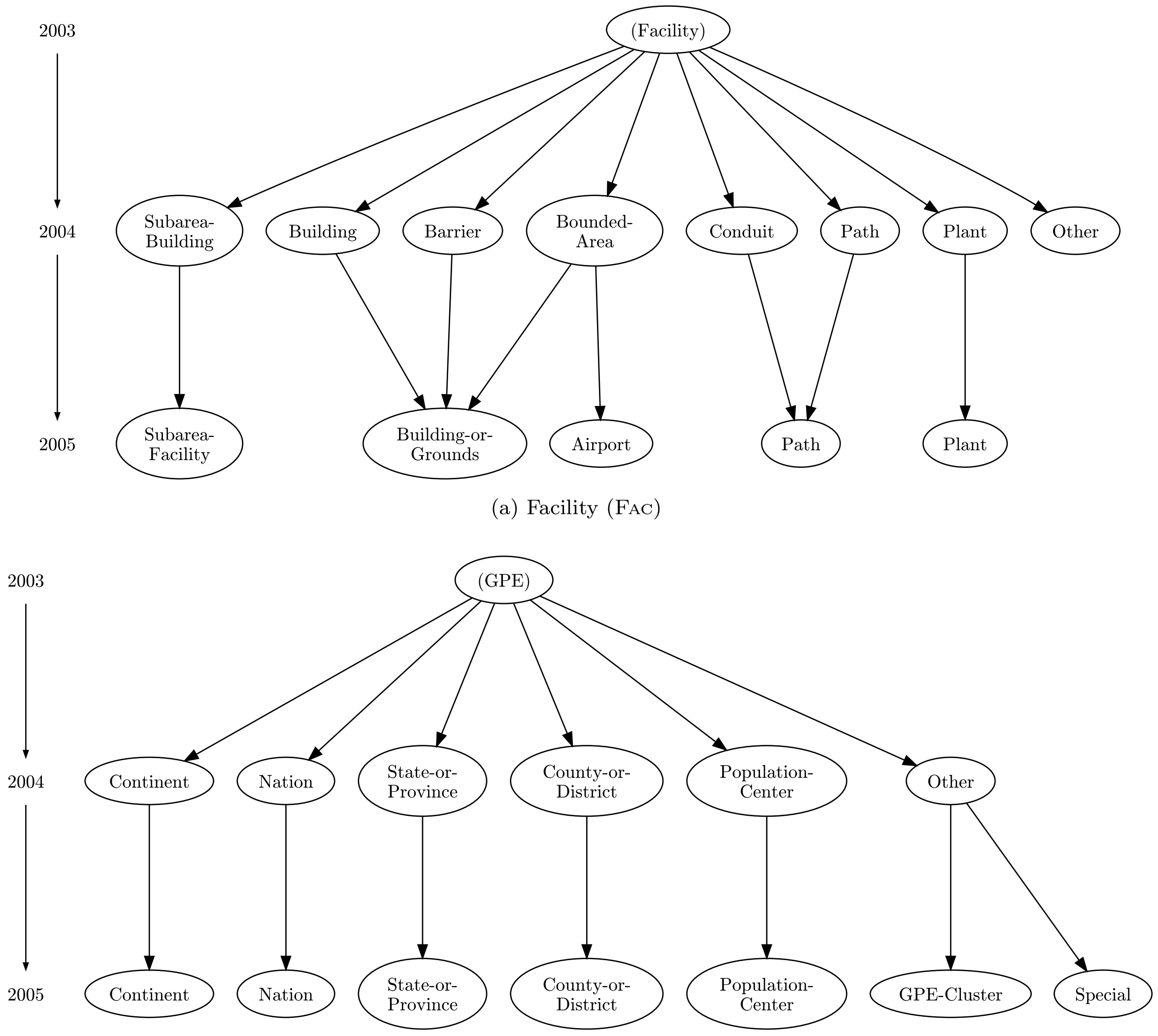
C
ACE Annotation

This appendix contains the full set of figures representing, in a diagrammatic fashion, the entity and relation types and subtypes hierarchies used in the ACE-2003, ACE-2004 and ACE-2005 evaluations—as well as the evolution of subtypes into others as the annotation schemes changed from year to year.

We want to reiterate that the correspondences are established by ourselves after comparison of the annotation guidelines, and do not necessarily correspond to the official view of the ACE organizers.

Figures C.1 to C.3 contain the entity types and subtypes. Each subfigure concerns one type, and each ellipse corresponds to a subtype. Solid and dashed arrows try to capture what we have judged, respectively, as a full or partial correspondence from one subtype to another.

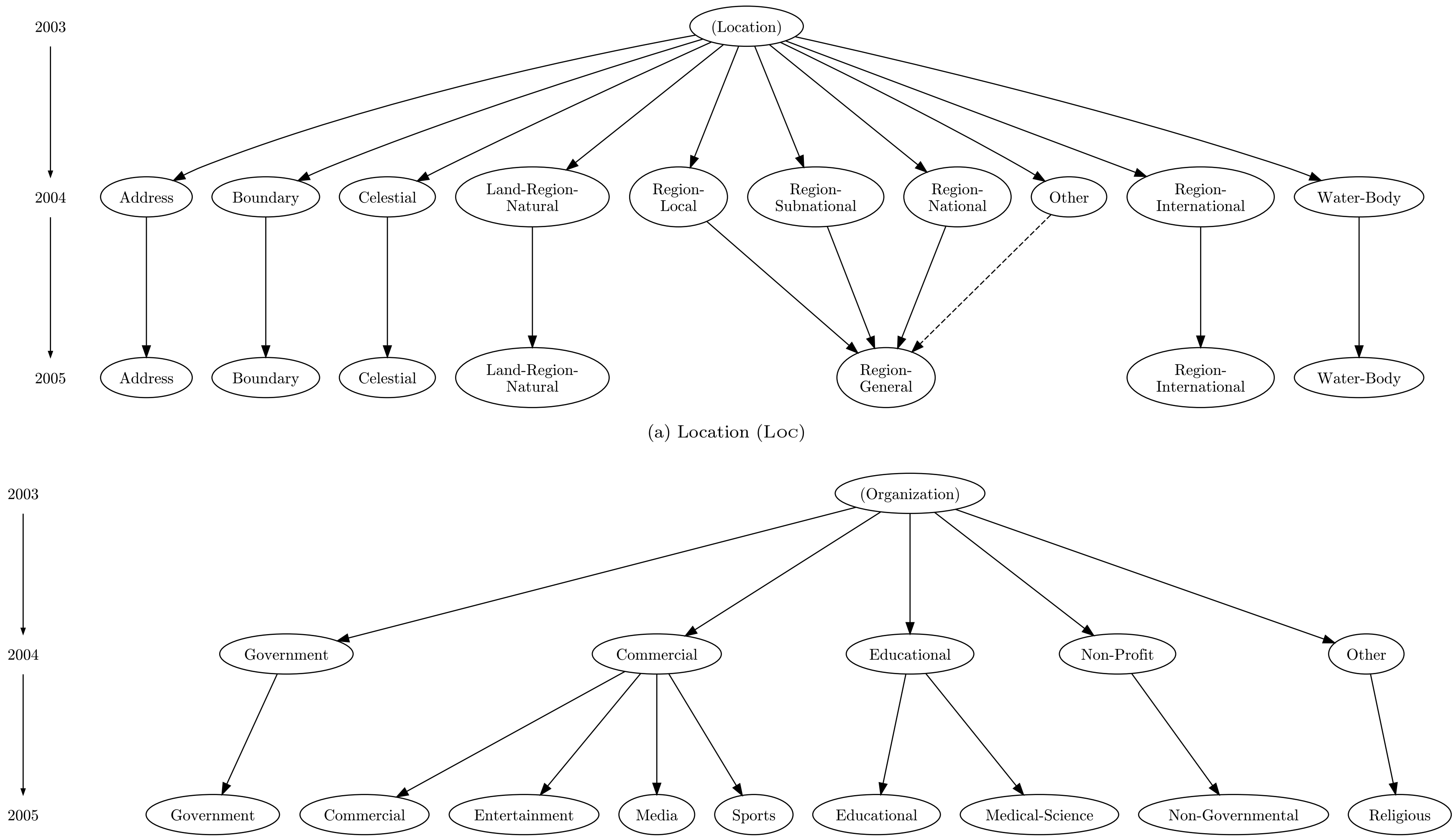
In turn, Figures C.4 to C.6 contain the relation types and subtypes. In each plot, named boxes correspond to types and ellipses to subtypes. Similarly, solid and dashed arrows depict full or partial subtype correspondence. A number of boxes and ellipses appear dotted—this is to indicate that only some of the subtypes in the type are represented in the figure, and that the full fragment of the hierarchy is to be found elsewhere.



(a) Facility (Fac)

(b) Geo-Political Entity (Gpe)

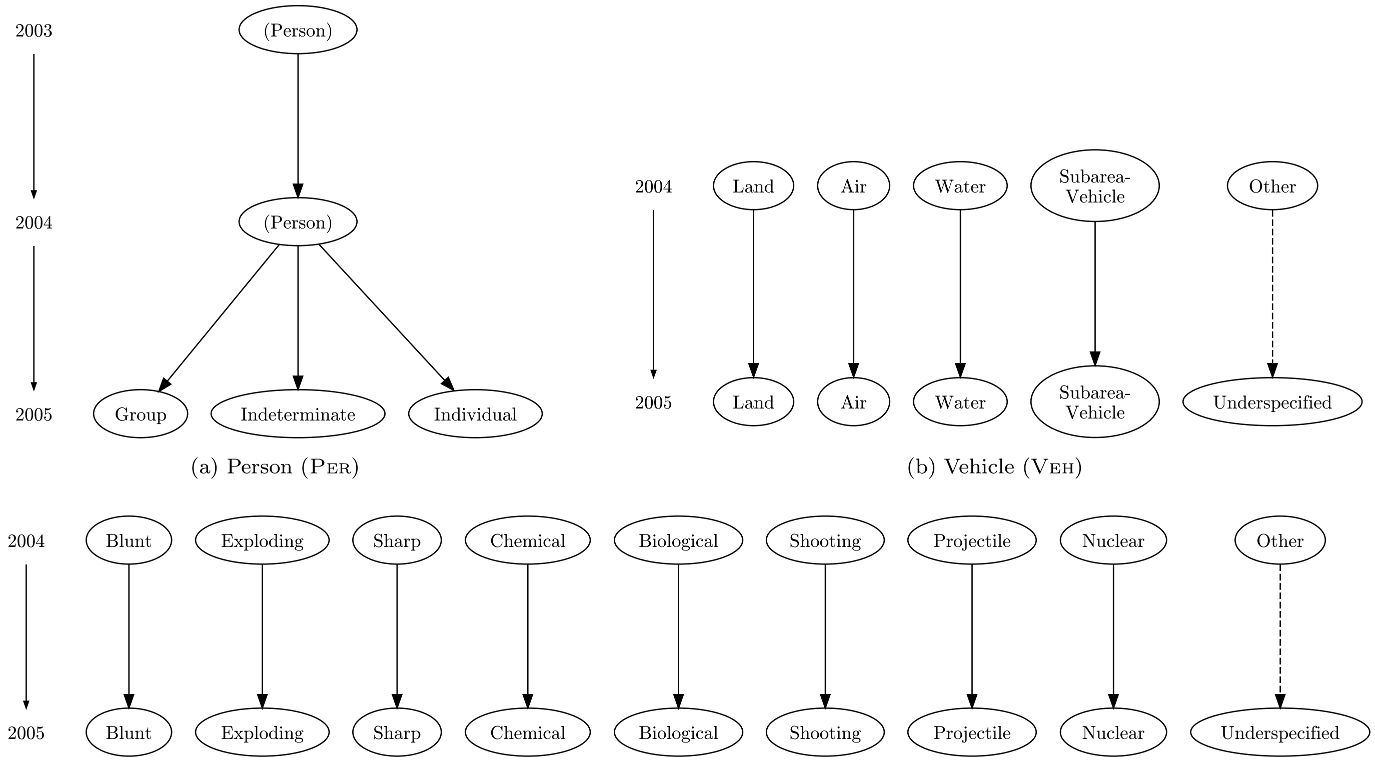
Figure C.1: Entity types in the ACE 2003–2005 evaluations (I)



(a) Location (Loc)

(b) Organization (Org)

Figure C.2: Entity types in the ACE 2003–2005 evaluations (II)



(a) Person (Per)

(b) Vehicle (Veh)

(c) Weapon (Wea)

Figure C.3: Entity types in the ACE 2003–2005 evaluations (III)

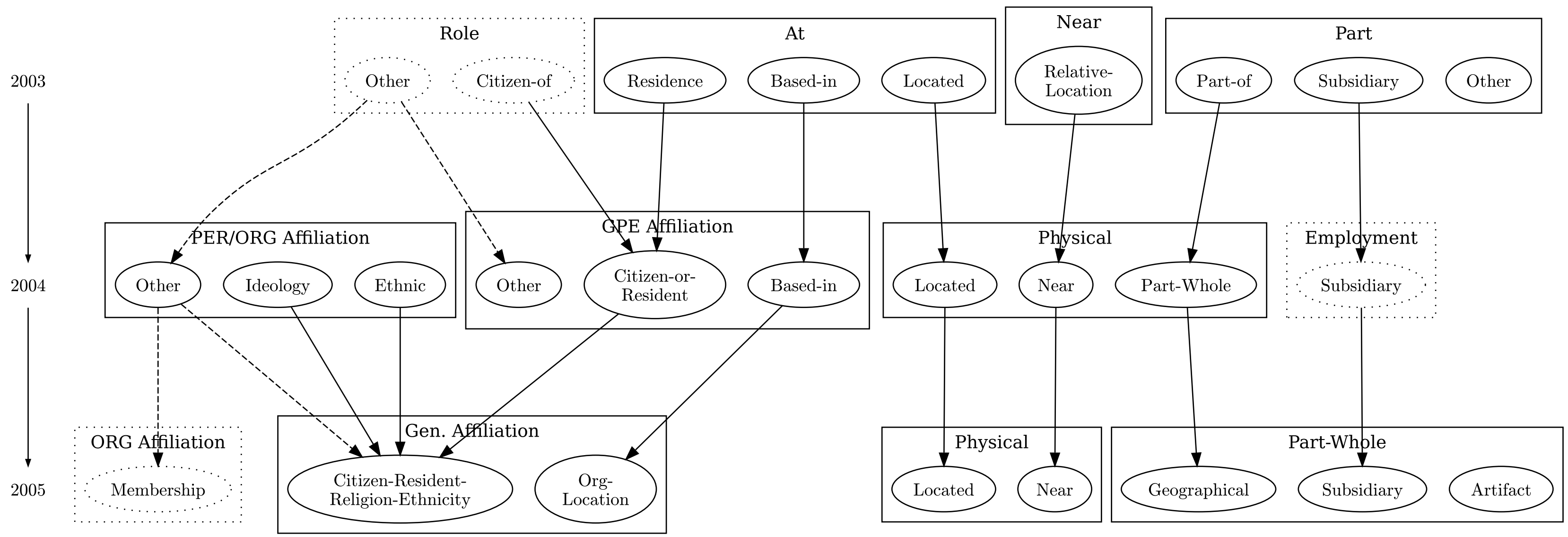


Figure C.4: Relation types in the ACE 2003–2005 evaluations (I)

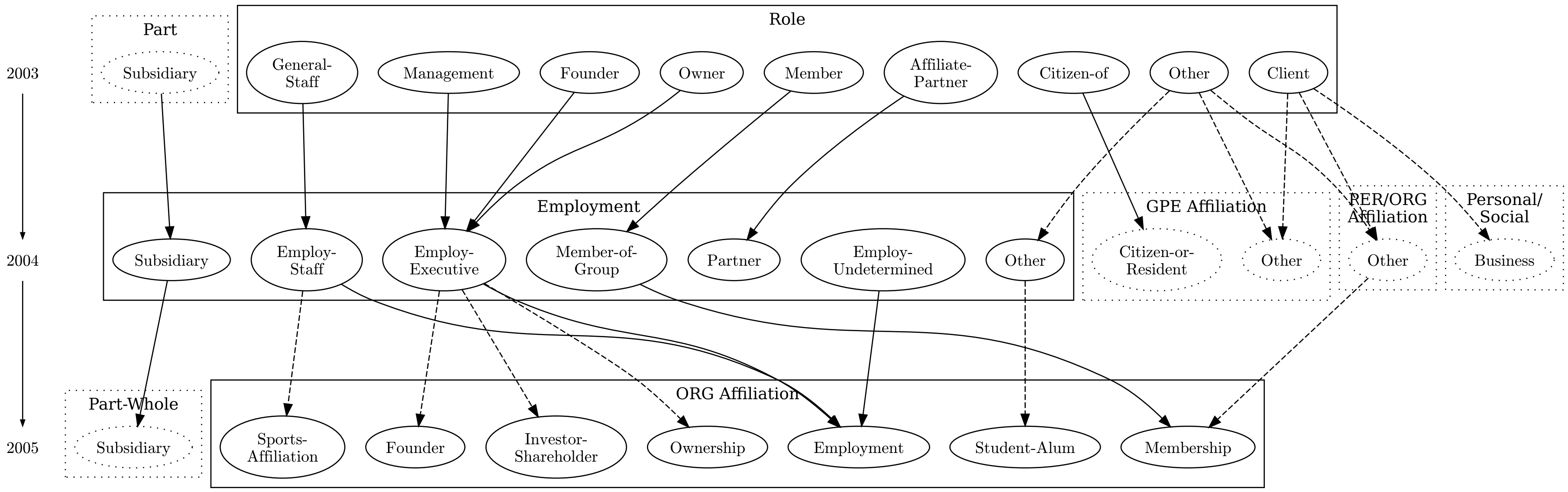


Figure C.5: Relation types in the ACE 2003–2005 evaluations (II)

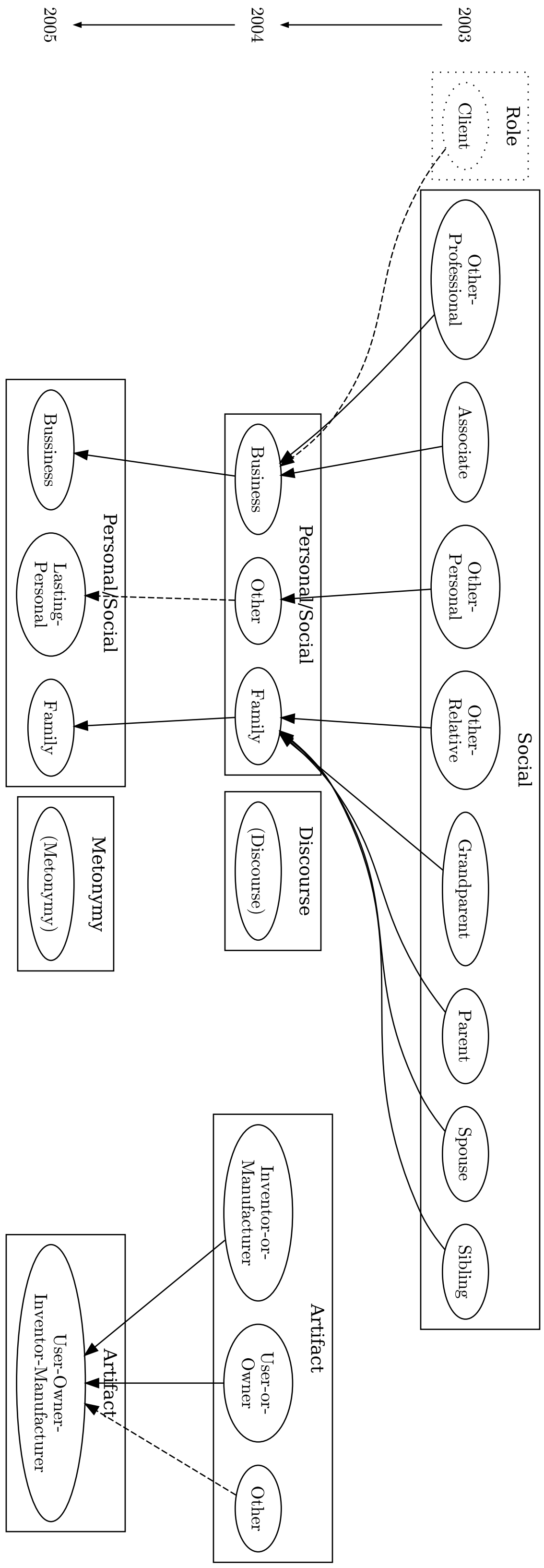


Figure C.6: Relation types in the ACE 2003–2005 evaluations (III)

D List of Publications

This appendix contains a list of the publications which have been produced with the research in this thesis. For each one of them, the sections of this document which present the work there contained are referred.

D.1 Conference Papers

- (Gonzàlez and Turmo, 2005)

  Edgar Gonzàlez and Jordi Turmo. Unsupervised clustering of spontaneous speech documents. In *9th European Conference on Speech Communication and Technology (EuroSpeech/InterSpeech)*, 2005.

  This paper contains our first experiments on the task of unsupervised document clustering—in particular, the application of the GEO method on the SWB corpus. The method is presented, and evaluated on manual transcripts of spontaneous conversations from the Switchboard corpus. The results show how GEO finds a suitable estimation of the number of clusters in the collection, and exceeds the performance of its two components, HAC and EM.

  The work in the paper is subsumed by the much comprehensive evaluation presented in Section 3.5.

- (Gonzàlez and Turmo, 2008b)

  Edgar Gonzàlez and Jordi Turmo. Comparing non-parametric ensemble methods for document clustering. In *13th International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 245–256, 2008.

  This paper contains another series of experiments on unsupervised document clustering—concerning, in this case, the comparison of individual and combination methods, and between different ensemble generation strategies. The MAJOR and MINOR strategies are presented, as well as the triad of individual methods GEO, HI and INFO (referred to in the paper as IT). The AGGLO+L algorithm is used for clustering combination. The experiments included in the evaluation are hence a subset of those included in Section 3.5. The conclusions, nevertheless, remain the same: the superiority of ensemble methods—and, in particular, the MAJOR strategy—over individual methods, and that of GEO among the latter.

  Most parts of Sections 3.4 and 3.5 of this document correspond to extended versions of the work in the paper.

- (Gonzàlez and Turmo, 2009)

  Edgar Gonzàlez and Jordi Turmo. Unsupervised relation extraction by massive clustering. In *9th IEEE International Conference on Data Mining (ICDM)*, pages 782–787, 2009.

  This paper contains our joint approach for unsupervised learning of relation detection pattern. Specifically, it contains the first presentation of the Ewocs algorithm—albeit without this name, and in a slightly different form—using the Prob algorithm and the Dist threshold detection criterion, on the APW and ACE corpora. However, only patterns based on word POS tags are considered (feature set `w:t`).

  Chapter 6 contains most of the ideas presented in this paper, especially in Sections 6.2.2.1 and 6.3. However, the evaluation therein subsumes that in the paper, as additional algorithms and feature sets are included. Regarding the Ewocs algorithm itself, it has been formalized and evaluated in isolation in Chapter 5.

D.2 Journal Articles

- (Gonzàlez and Turmo, 2008a)

  Edgar Gonzàlez and Jordi Turmo. Non-parametric document clustering by ensemble methods. *Procesamiento del Lenguaje Natural*, 40:91–98, 2008.

  This article is a journal version of the paper presented the same year in the International Conference on Applications of Natural Language to Information Systems. Its contents are hence also included in Sections 3.4 and 3.5.

D.3 Technical Reports

- (Gonzàlez and Turmo, 2006)

  Edgar Gonzàlez and Jordi Turmo. Unsupervised document clustering by weighted combination. Technical Report LSI-06-17-R, Department de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 2006.

  This report contains our proposal of a method for weighted clustering ensemble combination, based on the unweighted approaches of Strehl and Ghosh (2002) and Topchy et al. (2005). The report describes the proposed algorithm, and evaluates it on ensembles built from the Geo, Hi (referred to as HiIT) and Info (referred to as IT) clusterers, following what we have subsequently named the Minor strategy. The results show how an improvement in clustering performance can often be achieved with the incorporation of weighting. However, the computational cost of the approach eventually made us drop it in favour of the simpler Agglo, Balls and Furth algorithms.

Bibliography

Kenji Abe, Shinji Kawasoe, Tatsuya Asai, and Hiroki Arimura. Optimized substructure discovery for semi-structured data. In Tapio Elomaa, Heikki Mannila, and Hannu Toivonen, editors, *Principles of Data Mining and Knowledge Discovery*, volume 2431 of *Lecture Notes in Computer Science*, pages 57–100. Springer, 2002.

Steven Abney. Understanding the Yarowsky Algorithm. *Computational Linguistics*, 30(3), 2004.

ACE. *The ACE 2003 Evaluation Plan*, 2003. URL ftp://jaguar.ncsl.nist.gov/ace/doc/ace_evalplan-2003.v1.pdf.

ACE. *The ACE 2004 Evaluation Plan*, 2004. URL http://www.itl.nist.gov/iad/mig/tests/ace/2004/doc/ace04-evalplan-v7.pdf.

ACE. *The ACE 2005 (ACE05) Evaluation Plan*, 2005a. URL http://www.itl.nist.gov/iad/mig/tests/ace/2005/doc/.

ACE. *ACE (Automatic Content Extraction) English Annotation Guidelines for Relations*, 2005b. URL http://www.ldc.upenn.edu/Projects/ACE/docs/English-Relations-Guidelines_v5.8.3.pdf.

ACE. Nist 2007 ace evaluation results, 2007. URL http://www.itl.nist.gov/iad/mig/tests/ace/2007/doc/ace07_eval_official_results_20070402.html.

Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *ACM Conference on Digital Libraries (DL)*, pages 85–94, 2000.

Eneko Agirre and Oier López de Lacalle. Clustering WordNet word senses. In *EuroConference Recent Advances in Natural Language Processing (RANLP)*, pages 121–130, 2003.

Rakesh Agrawal and Ramakrishnan Srikant. Fast algorithms for mining association rules in large databases. In *20th International Conference on Very Large Data Bases (VLDB)*, pages 487–499, 1994.

Hirotsugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723, 1974.

Shin Ando. Clustering needles in a haystack: An information theoretic analysis of minority and outlier detection. In *7th IEEE International Conference on Data Mining (ICDM)*, pages 13–22, 2007.

Shin Ando and Einoshin Suzuki. An information theoretic approach to detection of minority subsets in database. In *6th IEEE International Conference on Data Mining (ICDM)*, pages 11–20, 2006.

Nicholas Andrews and Naren Ramakrishnan. Seeded discovery of base relations in large corpora. In *46th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 591–599, 2008.

Douglas Appelt and David Israel. Introduction to information extraction technology: A tutorial prepared for IJCAI-99, 1999. URL `http://www.ai.sri.com/~appelt/ie-tutorial`.

Suguru Arimoto. An algorithm for computing the capacity of arbitrary discrete memoryless channels. *IEEE Transactions on Information Theory*, 18(1):14–20, 1972.

Arindam Banerjee, Srujana Merugu, Inderjit S. Dhillon, and Joydeep Ghosh. Clustering with Bregman divergences. *Journal of Machine Learning Research*, 6:1705–1749, 2005.

Jeffrey D. Banfield and Adrian E. Raftery. Model-based gaussian and non-gaussian clustering. *Biometrics*, 49(3):803–821, 1993.

Michele Banko and Oren Etzioni. The tradeoffs between open and traditional relation extraction. In *46th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2008.

Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. In *43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 238–247, 2002.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56: 89–113, 2004.

Roberto Basili, Maria Teresa Pazienza, and Michele Vindigni. Corpus-driven learning of event recognition rules. In *Machine Learning for Information Extraction Workshop (ECAI)*, 2000.

Roberto J. Bayardo, Jr. Efficiently mining long patterns from databases. In *ACM SIGMOD International Conference on Management of Data*, pages 85–93, 1998.

Richard Ernest Bellman. *Adaptive Control Processes*. Princeton University Press, 1961.

Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2002.

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71, 1996.

Beate Bergmann and Gerhard Hommel. Improvements of general multiple test procedures for redundant systems of hypotheses. In Peter Bauer, Gerhard Hommel, and Eckart Sonnemann, editors, *Multiple Hypothesenprüfung—Multiple Hypotheses Testing*, pages 100–115. Springer, 1988.

Jim C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.

Christophe Biernacki, Gilles Celeux, and Gérard Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

Richard E. Blahut. Computation of channel capacity and rate-distortion functions. *IEEE Transactions on Information Theory*, 18(4):460–473, 1972.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

Avrim Blum and Alexis Mitchell. Combining labeled and unlabeled data with co-training. In *11th Annual Conference on Computational Learning Theory (COLT)*, pages 92–100, 1998.

Joseph Bockhorst and Mark Craven. Markov networks for detecting overlapping elements in sequence data. In *17th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 193–200, 2004.

Roger E. Bohn and James E. Short. How much information? 2009 report on american consumers, 2009.

Constantinos Boulis and Mari Ostendorf. Combining multiple clustering systems. In *European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 63–74, 2004.

David M. Boulton and Christopher Stewart Wallace. The information content of a multistate distribution. *Journal of Theoretical Biology*, 23:269–278, 1969.

Lev M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7:200–217, 1967.

Leo Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

Sergei Brin. Extracting patterns and relations from the World-Wide Web. In *International Workshop on the Web and Databases (WebDB)*, 1998.

Razvan C. Bunescu and Raymond J. Mooney. Subsequence kernels for relation extraction. In *18th Annual Conference on Neural Information Processing Systems (NIPS)*, 2005.

Razvan C. Bunescu and Raymond J. Mooney. Learning to extract relations from the web using minimal supervision. In *45th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 576–583, 2007.

Lorna Byrne and John Dunnion. UCD IIRG at TAC 2010 KBP slot filling task. In *Text Analysis Conference*, 2010.

Mary Elaine Califf. *Relational Learning Techniques for Natural Language Information Extraction.* PhD thesis, University of Texas at Austin, 1998.

Tadeusz Caliński and Joachim Harabasz. A dendrite method for cluster analysis. *Communications in Statistics*, 3:1–27, 1974.

Nicola Cancedda, Eric Gaussier, Cyril Goutte, and Jean Michel Renders. Word sequence kernels. *Journal of Machine Learning Research*, 3:1059–1082, 2003.

Claire Cardie. Empirical methods in information extraction. *AI Magazine*, 18(4):65–79, 1997.

Vittorio Castelli, Radu Florian, and Ding jung Han. Slot filling through statistical processing and inference rules. In *Text Analysis Conference*, 2010.

Gilles Celeux and Gérard Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14(3):315–332, 1992.

Daniel Chada, Christian Aranha, and Carolina Monte. An analysis of the Cortex method at TAC 2010 KBP slot-filling. In *Text Analysis Conference*, 2010.

Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41:15:1–58, 2009.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Unsupervised feature selection for relation extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 262–267, 2005.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. Relation extraction using label propagation based semi-supervised learning. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 2006.

Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, and Heng Ji. CUNY-BLENDER TAC-KBP2010: Entity linking and slot filling system description. In *Text Analysis Conference*, 2010.

Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

Nancy A. Chinchor. MUC-4 evaluation metrics. In *4th Message Understanding Conference (MUC)*, pages 22–29, 1992.

Nancy A. Chinchor. Overview of MUC-7/MET-2. In *7th Message Understanding Conference (MUC)*, 1998.

Timothy Chklovski and Patrick Pantel. Verbocean: Mining the web for fine-grained semantic verb relations. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2004.

Grzegorz Chrupała, Saeedeh Momtazi, Michael Wiegand, Stefan Kazalski, Fang Xu, Benjamin Roth, Alexandra Balahur, and Dietrich Klakow. Saarland University Spoken Language Systems at the slot filling task of TAC KBP 2010. In *Text Analysis Conference*, 2010.

Stephen Clark and David Weir. Class-based probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206, 2002.

Michael Collins and Nigel Duffy. New ranking algorithms for parsing and tagging: Kernels over discrete structures, and the voted perceptron. In *40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 263–270, 2002.

Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 100–110, 1999.

Koby Crammer and Gal Chechik. A needle in a haystack: Local one-class optimization. In *21st International Conference on Machine Learning (ICML)*, pages 26–33, 2004.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

Koby Crammer, Partha Pratim Talukdar, and Fernando C. Pereira. A rate-distortion one-class model and its applications to clustering. In *25th International Conference on Machine Learning (ICML)*, pages 184–191, 2008.

Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

Aron Culotta, Andrew Kachites McCallum, and Jonathan Betz. Integrating probabilistic extraction models and data mining to discover relations and patterns in text. In *International Conference on Human Language Technology Research and Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 296–303, 2006.

Manoranjan Dash, Kiseok Choi, Peter Scheuermann, and Huan Liu. Feature selection for clustering — a filter solution. In *2nd IEEE International Conference on Data Mining (ICDM)*, pages 115–122, 2002.

Rajesh N. Davé. Characterization and detection of noise in clustering. *Pattern Recognition Letters*, 12(11):657–664, 1991.

Rajesh N. Davé and Raghu Krishnapuram. Robust clustering methods: A unified view. *IEEE Transactions on Fuzzy Systems*, 5(2), 1997.

Dmitry Davidov and Ari Rappoport. Efficient unsupervised discovery of word categories using symmetric patterns and high frequency words. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 2006.

Jesse Davis and Mark Goadrich. The relationship between Precision-Recall and ROC curves. In *23rd International Conference on Machine Learning (ICML)*, pages 233–240, 2006.

Gerald Francis de Jong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3:251–273, 1979.

Gerald Francis de Jong. An overview of the FRUMP system. In Wendy G. Lehnert and Martin H. Ringle, editors, *Strategies for Natural Language Processing*, pages 146–176. Lawrence Erlbaum Associates, 1982.

Peter J. Deer and Peter Eklund. A study of parameter values for a Mahalanobis distance fuzzy classifier. *Fuzzy Sets and Systems*, 137:191–213, 2003.

Arthur Pentland Dempster, Nan McKenzie Laird, and Donald Bruce Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Royal Statistical Society, Series B*, 39(1), 1977.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30, 2006.

Inderjit S. Dhillon and Yuqiang Guan. Information theoretic clustering of sparse co-occurrence data. In *3rd IEEE International Conference on Data Mining (ICDM)*, pages 517–520, 2003.

Inderjit S. Dhillon, Subramanyam Mallela, and Rahul Kumar. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3: 1265–1287, 2003.

Thomas G. Dietterich, Richard H. Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.

Evgenia Dimitriadou. *Exploratory Data Analysis and Applications.* PhD thesis, Technische Universität Wien, 2003.

Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik. A combination scheme for fuzzy clustering. In Nikhil Pal and Michio Sugeno, editors, *Advances in Soft Computing*, volume 2275 of *Lecture Notes in Computer Science*, pages 405–414. Springer, 2002.

George Doddington, Tom Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program: Tasks, data, and evaluation. In *International Conference on Language Resources and Evaluation (LREC)*, 2004.

Pedro Domingos and Michael J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29:103–130, 1997.

Doug Downey, Oren Etzioni, and Stephen Soderland. A probabilistic model of redundancy in information extraction. In *19th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1034–1041, 2005.

Doug Downey, Matthew Broadhead, and Oren Etzioni. Locating complex named entities in web text. In *20th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2733–2739, 2007.

Richard Dubes and Anil Kumar Jain. Clustering techniques: The user's dilemma. *Pattern Recognition*, 8:247–260, 1976.

Richard Dubes and Anil Kumar Jain. Clustering methodologies in exploratory data analysis. In Marshall C. Yovits, editor, *Advances in Computers*, volume 19, pages 113–228. Elsevier, 1980.

Joe C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. *Cybernetics and Systems: An International Journal*, 3:32–57, 1973.

Jennifer G. Dy and Carla E. Brodley. Feature selection for unsupervised learning. *Journal of Machine Learning Research*, 5:845–889, 2004.

Dave E. Eckhardt, Jr. and Larry D. Lee. A theoretical basis for the analysis of multiversion software subject to coincident errors. *IEEE Transactions on Software Engineering*, 11(12):1511–1517, 1985.

Kathrin Eichler, Holmer Hemsen, and Günter Neumann. Unsupervised relation extraction from web documents. In *International Conference on Language Resources and Evaluation (LREC)*, 2008.

Michael B. Eisen, Paul T. Spellman, Patrick O. Brown, and David Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95:14863–14868, 1998.

Hafida Essaqote, Nour-eddine Zahid, Mohammed Limouri, and Abderrahman Essaid. A new approach for unsupervised classification. *4OR: A Quarterly Journal of Operations Research*, 3: 39–49, 2005.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *2nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

Oren Etzioni, Michael J. Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in KnowItAll. In *International World Wide Web Conference (WWW)*, 2004.

Katti Faceli, Andre C.P.L.F. de Carvalho, and Marcilio C.P. de Souto. Multi-objective clustering ensemble. In *6th International Conference on Hybrid Intelligent Systems (HIS)*, pages 51–54, 2006.

Tom Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.

Ronen Feldman and Benjamin Rosenfeld. Boosting unsupervised relation extraction by using ner. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

Christiane Fellbaum, editor. *WordNet: An electronic lexical database.* e:mit, 1998.

Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *20th International Conference on Machine Learning (ICML)*, 2003.

Xiaoli Zhang Fern and Carla E. Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *21st International Conference on Machine Learning (ICML)*, pages 36–43, 2004.

Mario A.T. Figueiredo and Anil Kumar Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

Vladimir Filkov and Steven Skiena. Integrating microarray data by consensus clustering. In *15th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 418–426, 2003.

Chris Fraley and Adrian E. Raftery. How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588, 1998.

Ana Luisa N. Fred. Finding consistent clusters in data partitions. In Josef Kittler and Fabio Roli, editors, *Multiple Classifier Systems*, volume 2096 of *Lecture Notes in Computer Science*, pages 309–318. Springer, 2001.

Ana Luisa N. Fred and Anil Kumar Jain. Evidence accumulation clustering based on the K-Means algorithm. In Terry Caelli, Adnan Amin, Robert P.W. Duin, Dick de Ridder, and Mohamed Kamel, editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 2396 of *Lecture Notes in Computer Science*, pages 442–451. Springer, 2002.

Dayne Freitag. *Machine Learning for Information Extraction in Informal Domains.* PhD thesis, Computer Science Department, Carnegie Mellon University, 1998.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *2nd European Conference on Computational Learning Theory (EuroCOLT)*, 1995.

Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296, 1999.

Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

Hichem Frigui and Raghu Krishnapuram. A robust competitive clustering algorithm with applications in computer vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21 (5):450–465, 1999.

Keinosuke Fukunaga and Larry D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on Information Theory*, 21(1):32–40, 1975.

Gabriel Pui Cheong Fung, Jeffrey Xu Yu, and Hongjun Lu. Discriminative category matching: Efficient text classification for huge document collections. In *2nd IEEE International Conference on Data Mining (ICDM)*, pages 187–194, 2002.

Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *22nd National Conference on Artificial Intelligence (AAAI)*, 2006.

Sanyuan Gao, Yichao Cai, Si Li, Zongyu Zhang, Jingyi Guan, Yan Li, Hao Zhang, Weiran Xu, and Jun Guo. PRIS at TAC2010 KBP track. In *Text Analysis Conference*, 2010.

Salvador García and Francisco Herrera. An extension on "Statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9: 2677–2694, 2008.

Joydeep Ghosh and Alexander Strehl. Similarity-based text clustering: A comparative study. In Jacob Kogan, Charles Nicholas, and Marc Teboulle, editors, *Grouping Multidimensional Data*, pages 73–97. Springer, 2006.

Joydeep Ghosh, Alexander Strehl, and Srujana Merugu. A consensus framework for integrating distributed clusterings under limited knowledge sharing. In *NSF Workshop on Next Generation Data Mining*, pages 99–108, 2002.

Jim Giles. Internet encyclopaedias go head to head. *Nature*, 438:900–901, 2005.

Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. In *21st IEEE International Conference on Data Engineering (ICDE)*, pages 341–352, 2005.

Mark Girolami. Mercer kernel-based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.

Erhan Gokcay and Jose C. Principe. Information theoretic clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):158–171, 2002.

Edgar Gonzàlez and Jordi Turmo. Unsupervised clustering of spontaneous speech documents. In *9th European Conference on Speech Communication and Technology (EuroSpeech/InterSpeech)*, 2005.

Edgar Gonzàlez and Jordi Turmo. Unsupervised document clustering by weighted combination. Technical Report LSI-06-17-R, Department de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 2006.

Edgar Gonzàlez and Jordi Turmo. Non-parametric document clustering by ensemble methods. *Procesamiento del Lenguaje Natural*, 40:91–98, 2008a.

Edgar Gonzàlez and Jordi Turmo. Comparing non-parametric ensemble methods for document clustering. In *13th International Conference on Applications of Natural Language to Information Systems (NLDB)*, pages 245–256, 2008b.

Edgar Gonzàlez and Jordi Turmo. Unsupervised relation extraction by massive clustering. In *9th IEEE International Conference on Data Mining (ICDM)*, pages 782–787, 2009.

Derek Greene, Alexey Tsymbal, Nadia Bolshakova, and Pádraig Cunningham. Ensemble clustering in medical diagnostics. In *IEEE Symposium on Computer-Based Medical Systems (CBMS)*, pages 576–581, 2004.

Mark A. Greenwood and Mark Stevenson. Improving semi-supervised acquisition of relation extraction patterns. In *Workshop on Information Extraction Beyond the Document*, pages 29–35, 2006.

Nizar Grira, Michel Crucianu, and Nozha Boujemaa. Unsupervised and semi-supervised clustering: A brief survey. In *A Review of Machine Learning Techniques for Processing Multimedia Content*. Rapport du Réseau d'Excellence MUSCLE, 2004.

Ralph Grishman and Bonan Min. New York University KBP 2010 slot-filling system. In *Text Analysis Conference*, 2010.

Frank E. Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11: 1–21, 1969.

Régis Guillemaud and Michael Brady. Estimating the bias field of MR images. *IEEE Transactions on Medical Imaging*, 16(3):238–251, 1997.

Francesco Gullo, Andrea Tagarelli, and Sergio Greco. Diversity-based weighting schemes for clustering ensembles. In *SIAM International Conference on Data Mining (SDM)*, pages 437–448, 2009.

Gunjan Gupta and Joydeep Ghosh. Robust one-class clustering using hybrid global and local search. In *22nd International Conference on Machine Learning (ICML)*, pages 273–280, 2005.

Gunjan Gupta and Joydeep Ghosh. Bregman bubble clustering: A robust, scalable framework for locating multiple, dense regions in data. In *6th IEEE International Conference on Data Mining (ICDM)*, pages 232–243, 2006.

Stefan Hadjitodorov and Ludmila Kuncheva. Selecting diversifying heuristics for cluster ensembles. In Michal Haindl, Josef Kittler, and Fabio Roli, editors, *Multiple Classifier Systems*, volume 4472 of *Lecture Notes in Computer Science*, pages 200–209. Springer, 2007.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(10):993–1001, 1990.

Sanda M. Harabagiu and Steven J. Maiorano. Acquisition of linguistic patterns for knowledge-based information extraction. In *International Conference on Language Resources and Evaluation (LREC)*, 2000.

Zellig Sabbettai Harris. Distributional structure. *Word*, 10:146–162, 1954.

John A. Hartigan. *Clustering Algorithms*. John Wiley and Sons, 1975.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

Hany Hassan, Ahmed Hassan, and Ossama Emam. Unsupervised information extraction approach using graph mutual reinforcement. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

David Heckerman. A tutorial on learning with bayesian networks. Technical report, Microsoft Research, 1996.

Victoria Hodge and Jim Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22:85–126, 2004.

Ander Intxaurrondo, Oier Lopez de Lacalle, and Eneko Agirre. UBC at slot filling TAC-KBP 2010. In *Text Analysis Conference*, 2010.

Paul S. Jacobs, editor. *Text-Based Intelligent Systems: Current Research and Practice in Information Extraction and Retrieval.* Lawrence Erlbaum Associates, 1992.

Anil Kumar Jain, M. Narasimha Murty, and Patrick Joseph Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

Edwin Thompson Jaynes. Information theory and statistical mechanics i. *Physical Review, Series II*, 106(4):620–630, 1957.

Jay J. Jiang and David W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference on Research in Computational Linguistics (ROCLING)*, 1997.

Michael I. Jordan, editor. *Learning in Graphical Models.* MIT Press, 1998.

Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *42nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2004.

Leonard Kaufman and Peter J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis.* John Wiley and Sons, 2005.

Yeong-Seog Kim, W. Nick Street, and Filippo Menczer. Feature selection in unsupervised learning via evolutionary search. In *6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 365–369, 2000.

Benjamin King. Step-wise clustering procedures. *Journal of the American Statistical Association*, 62(317):86–101, 1967.

Scott Kirkpatrick, C. Daniel Gelatt, Jr., and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598), 1983.

Josef Kittler, Mohamad Hatef, Robert P.W. Duin, and Jiri Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

John Kleinberg. Authoritative soruces in a hyperlinked envirornment. *Journal of the Association for Computing Machinery*, 46:604–632, 1999.

George J. Klir and Bo Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications.* Prentice Hall, 1995.

Leon G. Kraft. A device for quantizing, grouping, and coding amplitude modulated pulses. Master's thesis, Electrical Engineering Department, Massachussetts Institute of Technology, 1949.

Solomon Kullback. *Information Theory and Statistics.* John Wiley and Sons, 1959.

Solomon Kullback and Richard A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86, 1951.

John Lafferty, Andrew Kachites McCallum, and Fernando C. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *18th International Conference on Machine Learning (ICML)*, pages 282–289, 2001.

John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. LCC approaches to knowledge base population at TAC 2010. In *Text Analysis Conference*, 2010.

Wendy G. Lehnert. Symbolic/subsymbolic sentence analysis: Exploiting the best of two worlds. In Jordan B. Pollack and John A. Barnden, editors, *Advances in Connectionist and Neural Computation Theory*, volume 1, pages 135–164. Ablex Publishing, 1991.

Friedrich Leisch. Bagged clustering. Technical Report Working Paper 51, SFB "Adaptive Information Systems and Modelling in Economics and Management Science", WU Vienna University of Economics and Business, 1999.

Erel Levine and Eytan Domany. Resampling method for unsupervised estimation of cluster validity. *Neural Computation*, 13(11):2573–2593, 2001.

David D. Lewis. Evaluating text categorization. In *Workshop on Speech and Natural Language Processing*, pages 312–318, 1991.

Hang Li and Naoki Abe. Generalizing case frames using a thesaurus and the MDL principle. *Computational Linguistics*, 24(2):217–244, 1998.

Tao Li, Sheng Ma, and Mitsunori Ogihara. Document clustering via adaptive subspace iteration. In *27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2004a.

Tao Li, Mitsunori Ogihara, and Sheng Ma. On combining multiple clusterings. In *Conference on Information and Knowledge Management (CIKM)*, 2004b.

Tao Li, Chris Ding, and Michael I. Jordan. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *7th IEEE International Conference on Data Mining (ICDM)*, pages 577–582, 2007.

Dekang Lin. An information-theoretic definition of similarity. In *15th International Conference on Machine Learning (ICML)*, pages 296–304, 1998.

Dekang Lin and Patrick Pantel. DIRT - discovery of inference rules from text. In *7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2001.

Jianhua Lin. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991.

Shin-Yee Lu and King S. Fu. A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(5):381–389, 1978.

Hui-Lan Luo, Xiao-Bing Xie, and Kang-Shun Li. A new method for constructing clustering ensembles. In *International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR)*, volume 2, pages 874–878, 2007.

Peter Lyman and Hal R. Varian. How much information, 2003. URL `http://www2.sims.berkeley.edu/research/projects/how-much-info-2003/`.

James B. MacQueen. Some methods for classification and anlysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

Prasanta Chandra Mahalanobis. On the generalised distance in statistics. *Proceedings of the National Institute of Sciences of India*, 2(1):49–55, 1936.

Inderjeet Mani, George Wilson, Lisa Ferro, and Beth Sundheim. Guidelines for annotating temporal information. In *International Conference on Human Language Technology Research (HLT)*, pages 1–3, 2001.

Christopher D. Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.

Andrew Kachites McCallum and Kamal Nigam. A comparison of event models for naive Bayes text classification. In *AAAI Workshop on Learning for Text Categorization*, pages 41–48, 1998.

Diana McCarthy. Using semantic preferences to identify verbal participation in role switching alternations. In *1st Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 256–263, 2000.

Brian McLernon and Nicholas Kushmerick. Transductive pattern learning for information extraction. In *EACL Workshop on Adaptive Text Extraction and Mining (ATEM)*, 2006.

Brockway McMillan. Two inequalities implied by unique decipherability. *IEEE Transactions on Information Theory*, 2(4):115–116, 1956.

Marina Meilă. Comparing clusterings by the variation of information. In Bernhard Schölkopf and Manfred Warmuth, editors, *Learning Theory and Kernel Machines*, volume 2777 of *Lecture Notes in Computer Science*, pages 173–187. Springer, 2003.

Marina Meilă and David Heckerman. An experimental comparison of model-based clustering methods. *Machine Learning*, 42:9–29, 2001.

James Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Philosophical Transactions of the Royal Society A*, 209:415–446, 1909.

Adam Meyers, Michiko Kosaka, Satoshi Sekine, Ralph Grishman, and Shubin Zhao. Parsing and GLARFing. In *EuroConference Recent Advances in Natural Language Processing (RANLP)*, 2001.

Glenn W. Milligan and Martha C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrica*, 50(2):159–179, 1985.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *47th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1003–1011, 2009.

Mary M. Moya, Mark W. Koch, and Larry D. Hostetler. One-class classifier networks for target recognition applications. In *World Congress on Neural Networks*, pages 797–801, 1993.

M. Narasimha Murty and Girish Krishna. A computationally efficient technique for data clustering. *Pattern Recognition*, 12:153–158, 1980.

Ion Muslea. Extraction patterns for information extraction tasks: A survey. In *AAAI Workshop on Machine Learning for Information Extraction*, 1999.

Dávid Nemeskey, Gábor Recski, Attila Zséder, and András Kornai. BUDAPESTACAD at TAC 2010. In *Text Analysis Conference*, 2010.

Nam Nguyen and Rich Caruana. Consensus clusterings. In *7th IEEE International Conference on Data Mining (ICDM)*, pages 607 –612, 2007.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Alexis Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3), 2000.

Francis Okeke and Arnon Karnieli. Linear mixture model approach for selecting fuzzy exponent value in fuzzy c-Means algorithm. *Ecological Informatics*, 1:117–124, 2006.

Patrick Pantel and Dekang Lin. Document clustering with committees. In *25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 199–206, 2002.

David Peel and Geoffrey J. McLachlan. Robust mixture modelling using the t distribution. *Statistics and Computing*, 10:339–348, 2000.

Jose Manuel Peña, Jose Antonio Lozano, and Pedro Larrañaga. An empirical comparison of four initialization methods for the k-means algorithm. *Pattern Recognition Letters*, 20(10):1027–1040, 1999.

Wim Peters, Ivonne Peters, and Piek Vossen. Automatic sense clustering in EuroWordNet. In *International Conference on Language Resources and Evaluation (LREC)*, pages 409–416, 1998.

John C. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In Alexander J. Smola, Peter L. Bartlett, Bernhard Schölkopf, and Dale Schuurmans, editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press, 2000.

Kunal Punera and Joydeep Ghosh. Soft cluster ensembles. In José Valente de Oliveira and Witold Pedrycz, editors, *Advances in Fuzzy Clustering and its Applications*, pages 69–91. John Wiley and Sons, 2007.

Longhua Qian and Guodong Zhou. Clustering-based stratified seed sampling for semi-supervised relation classification. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 346–355, 2010.

Longhua Qian, Guodong Zhou, Fang Kong, and Qiaoming Zhu. Semi-supervised learning for semantic relation classification using stratified sampling strategy. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1437–1445, 2009.

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Vijay V. Raghavan, Peter Bollmann, and Gwang S. Jung. A critical investigation of recall and precision as measures of retrieval system performance. *ACM Transactions on Information Systems*, 7:205–229, 1989.

Soumya Ray and Mark Craven. Supervised versus multiple instance learning: An empirical comparison. In *22nd International Conference on Machine Learning (ICML)*, pages 697–704, 2005.

Yoram Reich and Steven J. Fenves. The formation and use of abstract concepts in design. In Douglas H. Fisher, Jr, Michael J. Pazzani, and Pat Langley, editors, *Concept Formation: Knowledge and Experience in Unsupervised Learning*, pages 323–354. Morgan-Kaufmann, 1991.

Ellen Riloff. Automatically constructing a dictionary for information extraction tasks. In *11th National Conference on Artificial Intelligence (AAAI)*, pages 811–816, 1993.

Ellen Riloff. Automatically generating extraction patterns from untagged text. In *13th National Conference on Artificial Intelligence (AAAI)*, pages 1044–1049, 1996.

Ellen Riloff and Rosie Jones. Learning dictionaries for information extraction by multi-level bootstrapping. In *16th National Conference on Artificial Intelligence (AAAI)*, 1999.

Jorma J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.

Dennis Ritchie. The evolution of the Unix time-sharing system. In *Symposium on Language Design and Programming Methodology*, pages 25–35, 1979.

Ronald L. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

Stephen J. Roberts. Parametric and non-parametric unsupervised cluster analysis. *Pattern Recognition*, 30(2):261–272, 1997.

Benjamin Rosenfeld and Ronen Feldman. Ures: an unsupervised web relation extraction system. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 2006.

Benjamin Rosenfeld and Ronen Feldman. Clustering for unsupervised relation identification. In *Conference on Information and Knowledge Management (CIKM)*, 2007.

Benjamin Rosenfeld, Ronen Feldman, Moshe Fresko, Jonathan Schler, and Yonatan Aumann. TEG: a hybrid approach to information extraction. In *Conference on Information and Knowledge Management (CIKM)*, pages 589–596, 2004.

Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Computational and Applied Mathematics*, 20:53–65, 1987.

Binjamin Rozenfeld and Ronen Feldman. High-performance unsupervised relation extraction from large corpora. In *6th IEEE International Conference on Data Mining (ICDM)*, 2006.

Naomi Sager. *Natural Language Information Processing: A Computer Grammar of English and its Applications.* Addison-Wesley, 1981.

Tomoya Sakai, Atsushi Imiya, Takuto Komazaki, and Shiomu Hama. Critical scale for unsupervised cluster discovery. In Petra Perner, editor, *Machine Learning and Data Mining in Pattern Recognition*, volume 4571 of *Lecture Notes in Computer Science*, pages 218–232. Springer, 2007.

Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5:197–227, 1990.

Bernhard Schölkopf, John C. Platt, John Shawe-Taylor, Alexander J. Smola, and Robert C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13 (7):1443–1471, 2001.

Veit Schwämmle and Ole Nørregaard Jensen. A simple and fast method to determine the parameters for fuzzy c-Means cluster analysis. *Bioinformatics*, 26(22):2841–2848, 2010.

Gideon E. Schwartz. Estimating the dimension of a model. *Annals of Statistics*, 6(2):461–464, 1978.

Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34:1–47, 2002.

Satoshi Sekine. Automatic paraphrase discovery based on context and keywords between NE pairs. In *International Workshop on Paraphrasing: Paraphrase Acquisition and Applications (IWP)*, 2005.

Satoshi Sekine. On-demand information extraction. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 2006.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. Extended named entity hierarchy. In *International Conference on Language Resources and Evaluation (LREC)*, 2002.

Xavier Sevillano, Germán Cobo, Francesc Alías, and Joan Claudi Socoró. Robust document clustering by exploiting feature diversity in cluster ensembles. *Procesamiento del Lenguaje Natural*, 37, 2006.

Xavier Sevillano, Joan Claudi Socoró, and Francesc Alías. Fuzzy clusterers combination by positional voting for robust document clustering. *Procesamiento del Lenguaje Natural*, 43:245–253, 2009.

Claude Elwood Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:279–423, 623–656, 1948.

John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, 2004.

Yusuke Shinyama and Satoshi Sekine. Preemptive information extraction using unrestricted relation discovery. In *International Conference on Human Language Technology Research and Annual Meeting of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, 2006.

John E. Shore and Robert M. Gray. Minimum cross-entropy pattern classification and cluster analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 4(1):11–17, 1982.

Judith F. Silverman and David B. Cooper. Bayesian clustering for unsupervised estimation of surface and texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):482–495, 1988.

Noam Slonim and Naftali Tishby. Agglomerative information bottleneck. In *12th Annual Conference on Neural Information Processing Systems (NIPS)*, 1999.

Alan F. Smeaton. Using NLP or NLP resources for information retrieval tasks. In Tomek Strzalkowski, editor, *Natural Language Information Retrieval*, pages 99–111. Kluwer Academic Press, 1999.

Peter H.A. Sneath and Robert R. Sokal. *Numerical Taxonomy.* Freeman, 1973.

Stephen Soderland. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 34:233–272, 1999.

Robert R. Sokal and Charles Duncan Michener. A statistical method for evaluating systematic relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958.

Yang Song, Zhengyan He, and Houfeng Wang. ICL KBP approaches to knowledge base population at TAC2010. In *Text Analysis Conference*, 2010.

Kent A. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In *6th International Workshop on Machine Learning*, pages 160–163, 1989.

Karen Spärck-Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28:11–21, 1972.

Karen Spärck-Jones. What is the role of NLP in text retrieval? In Tomek Strzalkowski, editor, *Natural Language Information Retrieval.* Kluwer Academic Press, 1999.

Mark Stevenson. An unsupervised WordNet-based algorithm for relation extraction. In *Beyond Named Entity: Semantic Labelling for NLP Tasks Workshop at LREC*, pages 37–42, 2004.

Mark Stevenson and Mark A. Greenwood. A semantic approach to IE pattern induction. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 379–386, 2005.

Alexander Strehl and Joydeep Ghosh. Cluster ensembles — a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.

Kiyoshi Sudo, Satoshi Sekine, and Ralph Grishman. An improved extraction pattern representation model for automatic IE pattern acquisition. In *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 224–231, 2003.

Mihai Surdeanu and Massimiliano Ciaramita. Robust information extraction with perceptrons. In *NIST Automatic Content Extraction Workshop (ACE)*, 2007.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. A hybrid unsupervised approach for document clustering. In *11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2005.

Mihai Surdeanu, Jordi Turmo, and Alicia Ageno. A hybrid approach for the acquisition of information extraction patterns. In *EACL Workshop on Adaptive Text Extraction and Mining (ATEM)*, pages 49–56, 2006.

Mihai Surdeanu, David McClosky, Julie Tibshirani, John Bauer, Angel X. Chang, Valentin I. Spitkovsky, and Christopher D. Manning. A simple distant supervision approach for the TAC-KBP slot filling task. In *Text Analysis Conference*, 2010.

Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to Data Mining.* Addison-Wesley, 2005.

Dimitris K. Tasoulis and Michael N. Vrahatis. Unsupervised distributed clustering. In *IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN)*, 2004.

David M.J. Tax and Robert P.W. Duin. Support vector data description. *Machine Learning*, 54 (1):45–66, 2004.

Naftali Tishby, Fernando C. Pereira, and William Bialek. The information bottleneck method. In *37th Allerton Conference on Communication, Control, and Computing*, 1999.

Noriko Tomuro. Tree-cut and a lexicon based on systematic polysemy. In *2nd Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 1–8, 2001.

Alexander Topchy, Anil Kumar Jain, and William Punch. Combining multiple weak clusterings. In *3rd IEEE International Conference on Data Mining (ICDM)*, pages 331–338, 2003.

Alexander Topchy, Anil Kumar Jain, and William Punch. A mixture model for clustering ensembles. In *SIAM International Conference on Data Mining (SDM)*, pages 379–390, 2004.

Alexander Topchy, Anil Kumar Jain, and William Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005.

Robert Choate Tryon. *Cluster Analysis: Correlation Profile and Orthometric Analysis for the Isolation of Unities of Mind Personality.* Edward Brothers, 1939.

Jordi Turmo, Alicia Ageno, and Neus Català. Adaptive information extraction. *ACM Computing Surveys*, 38, 2006.

Peter Turney. Expressing implicit semantic relations without supervision. In *International Conference on Computational Linguistics and Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, 2006.

Stijn van Dongen. A cluster algorithm for graphs. Technical Report INS-R 0010, Centrum Wiskunde & Informatica, 2000.

Cornelis Joost van Rijsbergen. Foundation of evaluation. *j:jd*, 30(4):365–373, 1974.

Vladimir Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay Bharath Reddy, Sudheer Kovelamudi, Srikanth Reddy Vaddepally, Radheshyam Nanduri, Kiran Kumar N, Santhosh Gsk, and Prasad Pingali. IIIT Hyderabad in guided summarization and knowledge base population. In *Text Analysis Conference*, 2010.

Ellen Voorhees. Natural language processing and information retrieval. In Maria Teresa Pazienza, editor, *Information Extraction*, volume 1714 of *Lecture Notes in Computer Science*, pages 32–48. Springer, 1999.

Lipo Wang, editor. *Support Vector Machines: Theory and Applications.* Springer, 2005.

Satosi Watanabe. *Knowing and Guessing: A Quantitative Study of Inference and Information.* John Wiley and Sons, 1969.

Satosi Watanabe. *Pattern Recognition: Human and Mechanical.* John Wiley and Sons, 1985.

Peter A. Whigham. Search bias, language bias and genetic programming. In *1st Annual Conference on Genetic Programming (GECCO)*, pages 230–237, 1996.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

Rui Xu and Donald C. Wunsch, II. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

Yulan Yan, Naoaki Okazaki, Yutaka Matsuo, Zhenglu Yang, and Mitsuru Ishizuka. Unsupervised relation extraction by mining wikipedia texts using information from the web. In *47th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1021–1029, 2009.

Yiming Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1:69–90, 1999.

Roman Yangarber. Counter-training in discovery of semantic patterns. In *41st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 343–350, 2003.

Roman Yangarber and Ralph Grishman. Issues in corpus-trained information extraction. In *International Symposium on Spontaneous Speech: Toward the Realization of Spontaneous Speech Engineering*, 2000.

Roman Yangarber and Ralph Grishman. Customization of information extraction systems. In *International Workshop on Lexically Driven Information Extraction*, 1997.

Roman Yangarber, Ralph Grishman, Pasi Tapanainen, and Silja Huttunen. Automatic acquisition of domain knowledge for information extraction. In *Conference on Applied Natural Language Processing (ANLP-NAACL)*, pages 940–946, 2000.

Roman Yangarber, Winston Lin, and Ralph Grishman. Unsupervised learning of generalized names. In *Conference on Computational Linguistics (COLING)*, 2002.

Limin Yao, Aria Haghighi, Sebastian Riedel, and Andrew Kachites McCallum. Structured relation discovery using generative models. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1456–1466, 2011.

David Yarowsky. Unsupervised word sense disambiguation rivaling unsupervised methods. In *33rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 1995.

Alexander Yates and Oren Etzioni. Unsupervised resolution of objects and relations on the web. In *8th Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 121–131, 2007.

Steve Young and Gerritt Bloothoft, editors. *Corpus-Based Methods in Language and Speech Processing*. Kluwer Academic Press, 1997.

Jiang Yu, Qiansheng Cheng, and Houkuan Huang. Analysis of the weighting exponent in the FCM. *IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics*, 34, 2004.

Jingtao Yu, Omkar Mujgond, and Rob Gaizauskas. The University of Sheffield system at TAC KBP 2010. In *Text Analysis Conference*, 2010.

Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.

Mohammed J. Zaki. Efficiently mining frequent trees in a forest. In *8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 71–80, 2002.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106, 2003.

Harry Zhang. The optimality of naive bayes. In *International Florida Artificial Intelligence Research Society Conference Conference (FLAIRS)*, pages 562–567, 2004a.

Min Zhang, Jian Su, Danmei Wang, Guodong Zhou, and Chew Lim Tan. Discovering relations between named entities from a large raw corpus using tree similarity-based clustering. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 378–389, 2005.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. BIRCH: an efficient data clustering method for very large databases. In *ACM SIGMOD International Conference on Management of Data*, pages 103–114, 1996.

Zhu Zhang. Weakly-supervised relation classification for information extraction. In *Conference on Information and Knowledge Management (CIKM)*, pages 581–588, 2004b.

Bin Zhao, Fei Wang, and Changshui Zhang. Efficient multiclass maximum margin clustering. In *25th International Conference on Machine Learning (ICML)*, pages 1248–1255, 2008.

Shubin Zhao and Ralph Grishman. Extracting relations with integrated information using kernel methods. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 419–426, 2005.

Ying Zhao and George Karypis. Criterion functions for document clustering: Experiments and analysis. Technical Report 01-40, University of Minnesota, 2001.

Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55:311–331, 2004.

Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: A comparative study. *Knowledge and Information Systems*, 8:374–384, 2005.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. Exploring various knowledge in relation extraction. In *43rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 427–434, 2005.

Guodong Zhou, Min Zhang, Donghong Ji, and Qiaoming Zhu. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 728–736, 2007.

Guodong Zhou, Junhui Li, Longhua Qian, and Qiaoming Zhu. Semi-supervised learning for relation extraction. In *International Joint Conference on Natural Language Processing (IJCNLP)*, pages 32–39, 2008.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107, Carnegie Mellon University, 2002.

Xinhua Zhuang, Yan Huang, Kannappan Palaniappan, and Yunxin Zhao. Gaussian mixture density modeling, decomposition, and applications. *IEEE Transactions on Image Processing*, 5(9):1293–1302, 1996.

Zoran Zivkovic and Ferdinand van der Heijden. Recursive unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(5):651–656, 2004.

List of Algorithms

List of Figures

List of Tables