UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament de Ciències de la Computació

Document-Level Machine Translation

Ensuring Translational Consistency of Non-Local Phenomena

This dissertation has been submitted to
the *Computer Science Department (CS)* at
Universitat Politècnica de Catalunya (UPC)
in partial fulfilment of the requirements of
the *Artificial Intelligence Ph.D. programme*
for the degree of *Doctor of Philosophy*

by

Eva Martínez Garcia

with advisors

Cristina España-Bonet
Lluís Màrquez Villodre

Barcelona, September 10, 2019

Abstract

In this thesis, we study the automatic translation of documents by taking into account cross-sentence phenomena. This document-level information is typically ignored by most of the standard state-of-the-art Machine Translation (MT) systems, which focus on translating texts processing each of their sentences in isolation. Translating each sentence without looking at its surrounding context can lead to certain types of translation errors, such as inconsistent translations for the same word or for elements in a coreference chain. We introduce methods to attend to document-level phenomena in order to avoid those errors, and thus, reach translations that properly convey the original meaning.

Our research starts by identifying the translation errors related to such document-level phenomena that commonly appear in the output of state-of-the-art Statistical Machine Translation (SMT) systems. For two of those errors, namely inconsistent word translations as well as gender and number disagreements among words, we design simple and yet effective post-processing techniques to tackle and correct them. Since these techniques are applied a posteriori, they can access the whole source and target documents, and hence, they are able to perform a global analysis and improve the coherence and consistency of the translation. Nevertheless, since following such a two-pass decoding strategy is not optimal in terms of efficiency, we also focus on introducing the context-awareness during the decoding process itself. To this end, we enhance a document-oriented SMT system with distributional semantic information in the form of bilingual and monolingual word embeddings. In particular, these embeddings are used as Semantic Space Language Models (SSLMs) and as a novel feature function. The goal of the former is to promote word translations that are semantically close to their preceding context, whereas the latter promotes the lexical choice that is closest to its surrounding context, for those words that have varying translations throughout the document. In both cases, the context extends beyond sentence boundaries.

Recently, the MT community has transitioned to the neural paradigm. The final step of our research proposes an extension of the decoding process for a Neural Machine Translation (NMT) framework, independent of the model architecture, by shallow fusing the information from a neural translation model and the context semantics enclosed in the previously studied SSLMs. The aim of this modification is to introduce the benefits of context information also into the decoding process of NMT systems, as well as to obtain an additional validation for the techniques we explored.

The automatic evaluation of our approaches does not reflect significant variations. This is expected since most automatic metrics are neither context- nor semantic-aware and because the phenomena we tackle are rare, leading to few modifications with respect to the baseline translations. On the other hand, manual evaluations demonstrate the positive impact of our approaches since human evaluators tend to prefer the translations produced by our document-aware systems. Therefore, the changes introduced by our enhanced systems are important since they are related to how humans perceive translation quality for long texts.

Acknowledgments[1]

One of the most important things that I have learned in the long journey of this thesis is that you can make a good work alone but it will be better if you work in a team.

Primer de tot, vull agraïr als meus tutors Cristina España i Lluís Màrquez per tota la seva (santa) paciència durant tot aquest procés. Agraïr-vos totes les coses que m'heu donat l'oportunitat d'apendre. Al cap i a la fi el meu camí a MT va començar amb vosaltres. També vos heu convertit en un referent de com ser un bon professional, un bon "researcher". M'heu ensenyat com treballar amb gust pel que es fa, com donar-li la volta a un resultat dolent per aprendre i continuar treballant; sobretot això, com continuar treballant tot i que les circumstàncies no siguen les millors. El Lluís m'ha ensenyat com mai s'està massa ocupat per a no dedicar atenció a les coses importants, a saber distingir entre lo important i lo urgent, a no tindre por a preguntar i a no perdre mai la curiositat. La Cristina m'ha ensenyat que sempre es pot fer més, que sempre es pot treballar més i millor, que la constància n'és la clau per arribar a la meta.

Menció especial té el Lluís Padró, per ser el nostre "flawless" infiltrat a la UPC. Moltes gràcies per tenir sempre la millor de les disposicions per ajudar-nos, sempre a punt per trobar sol·lucions.

I would like to thank Joakim Nivre, Jörg Tiedemann, and Christian Hardmeier, who welcomed me at Uppsala Universitet back in 2013. Thanks for having me there, thanks for showing me other ways of making things, thanks for letting me learn from you. A great part of this thesis work began from the ideas I got during my research stay, it was a very productive experience for me. I would like to specially thank Christian for his infinite patience with the installation of DOCENT, and for being there for me as a valuable contact during the years. Also from this Swedish journey, I had the chance to meet very lovely and welcoming people. Thanks to Marie, Mojgan, Ali, Eva, Mats, Vera. My heart will always have a Swedish part because of you. Tack!

També volia agraïr al grup TALP de la UPC per tindre'm. Tot i que ara estiguem dispersos, sempre pense en vosaltres com el meu grup madriu: Meritxell, Jesús, Maria, Pere, Xavi Lluís, Edgar, Danielle, Audi, Pranava, Ali, Horacio. Sou gent molt maca i fàcil de trobar a faltar.

Tot i que el seu treball es queda sempre en un segon pla, també volia agraïr al personal de la secretaria del departament de LSI/CS per tot el suport durant la meva vinculació amb la UPC. Especialment a la Mercè Juan, perquè fa miracles burocràtics, fent senzill lo que sembla impossible i estant sempre de bon humor.

Mientras estuve en la UPC, también tuve la suerte de trabajar con el grupo IXA de la UPV/EHU: Kepa, Iñaki Alegría, Iñaki San Vicente, Nora, Mikel, Gorka. Con

[1]This work has been partially supported by an FPI 2010 grant from the Spanish Ministry of Science and Innovation (MICINN) within the OpenMT-2 project (ref. TIN2009-14675-C03-03, `http://ixa.si.ehu.es/openmt2/en/index.html`) of MICINN, a mobility EEBB 2013 grant from the Spanish Ministry of Economy and Competitiveness (MINECO) for a stay at the Department of Linguistics and Philology at the Uppsala University, and by the TACARDI project (ref. TIN2012-38523-C02-02, `http://ixa.si.ehu.es/tacardi/en/index.html`) of the MINECO.

algunos de vosotros compartí mi primera sagardotegi o mi primer congreso internacional, y eso son cosas que no se olvidan. Eskerrik asko denoi!

Los últimos tres años de este viaje los he pasado en San Sebastián. Mila esker Arantza por toda tu ayuda en la revisión del draft de esta tesis. Also, to the ELRI consortium (Helen, Jane, Andy, Teresa, Gary, Victoria, Hervé, Rui, Maite), thanks for letting me grow as professional and to believe in my skills again. En este periodo también tuve la suerte de colaborar con Christian Blum para la aplicación del ACO a MT. Muchas gracias por tu tiempo y tu buena disposición, Christian, fue un verdadero placer poder trabajar contigo.

Durante estos años he podido trabajar desde distintos sitios: 8 mudanzas, 5 ciudades y mucha gente involucrada en el éxito de este trabajo.

Els meus companys de màster i de fatigues "tesils": el Gabo i l'Àtia. Els companys de despatx de l'Omega: Jorge, Nikita, Àlex, Alberto, (Gran) Javier, Xuri, Dani, Sergi, Alessandra, Josep Lluís, Lucas, Eve, Albert, Mª Àngels, Lander, Jesús, Adrià, Guillem. Gràcies per compartir els "pastissos allibera estrés". Especialmente a Carles. Estuviste en este proceso desde el principio, y con paciencia infinita me has acompañado hasta el final. Gracias por todo, todo y todo.

En el sector vasco, mila esker a Haritz i Cristina per aguantar els baixons i els subidons i per compartir les nits de guacamole. Montse, moltes gràcies per tot, pels ànims i per alegrar els dies. También a Andoni, por recordarme que es importante saber lo que quiero y a veces aún más lo que no quiero, y enseñarme la parte más amable del País Vasco.

Al sector en la distancia, que no sois pocos. Teresa, Álex, Montse i Aleix, mis chicas de italiano, gracias por llenar mis días en Barcelona de un color especial. Javi, Sandra y Javi, el pequeño reducto castellonense que hace más dulce la vuelta a casa. Marina, Manu, sé que los dos desde Madrid habéis estado pendientes de cómo se desarrollaba esta aventura, gracias por estar ahí. Los informates (Ester, Marta, Jorge, Javi V., Javi A., Gerardo, Miguel Ángel, Luis, Jaime) me habéis visto crecer y, aunque desde lejos, me habéis acompañado a lo largo de todo este camino. Gracias por ser.

Finalmente, una de las partes más importantes de este trabajo, la familia. Sin esa red de seguridad no podría haber llegado hasta donde estoy hoy. Mamá, Rubén, aunque no hayáis participado en el trabajo técnico sí que habéis sufrido y compartido todos los momentos de estrés, el estar lejos, las prisas para entregar un artículo, la frustración porque algo no funciona. Sin vuestro apoyo y cariño incondicional esto no habría sido posible, esta tesis está dedicada a vosotros por ser la parte infalible de mi vida.

Aquesta tesi és tota vostra. This thesis is all yours. Tesi hau guztia zuena da. Esta tesis es toda vuestra.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Machine Translation (MT) can be defined as the use of a computer to translate a message from one natural language into another. It is a well-known Natural Language Processing (NLP) research area which has become quite popular nowadays. MT is very present in our daily lives. We use it to access information in other languages on the Internet or to figure out how to say something in languages we do not master for interaction and communication purposes. We are frequent users of the most popular online translation services (e.g., Google Translate[1], Bing[2], Reverso[3], or DeepL[4]) and we are also used to consuming the MT services provided by social networks (e.g., Facebook[5] or Twitter[6]), which allow us to access the published information in our preferred language. MT is also a common feature in email services, like Gmail[7] or Outlook Live[8], which use it to facilitate information exchange across language barriers. MT is present even in telecommunication applications like Skype[9], which offers video chats with real-time speech-to-speech translation services. This extended use of MT technology makes us familiarized with its advantages and drawbacks.

Although current MT systems have achieved good translation quality, even comparable with human translation quality in some cases (Hassan et al., 2018; Wu et al., 2016), they still hold a known limitation: they work at sentence level. MT systems translate a document sentence by sentence, taking into account a short context and ignoring document-level information. On the one hand, Rule Based Machine

[1] `https://translate.google.com`
[2] `https://www.bing.com/translator`
[3] `http://www.reverso.net`
[4] `https://www.deepl.com`
[5] `https://www.facebook.com`
[6] `https://twitter.com`
[7] `https://mail.google.com`
[8] `https://outlook.live.com`
[9] `https://www.skype.com/`

Translation (RBMT) systems (Bennett and Slocum, 1985) and Statistical Machine Translation (SMT) systems (Brown et al., 1990; Koehn et al., 2003) apply rules or statistical models, respectively, to estimate the best translation for a phrase only looking at most at the sentence context. On the other hand, Neural Machine Translation (NMT) approaches (Bahdanau et al., 2015), both those based on Recurrent Neural Networks (RNN) and those exploiting attentional networks (Transformer), work with vector representations for sentences, only taking into account the intra-sentence information. In either case, ignoring extra-sentential information is required due to performance concerns and to the difficulty of properly representing long-distance dependencies. SMT systems rely on local n-gram information, and for NMT systems it is still an open problem how to represent long sequences of words with fixed-length vectors. Thus, state-of-the-art systems perform translation assuming that every sentence can be translated in an isolated way.

However, texts contain relationships among words that hold their coherence, cohesion, and consistency across sentences. These linguistic properties establish the connectedness in a text and can be defined as follows (Dijk, 1977; Halliday and Hasan, 1976; Sanders and Pander Maat, 2006). Coherence is a semantic property of discourses, based on the interpretation of each sentence relative to the others; it is what makes a text a unified whole and semantically meaningful. Cohesion refers to relations that exist within the text; it is the grammatical and lexical linking within a document that holds it together. And lexical consistency is the quality of compatibility for the words in a document by, e.g., repeated use of the same words or lemmas. We consider that a good translation should reflect and maintain these qualities at the same degree as they appear in the source text. This is the motivation for our work, which explores techniques to improve the coherence and cohesion levels of the translations generated by state-of-the-art MT systems. We take as inspiration how human translators can resolve these phenomena naturally, by using the entire document's context information.

Some of the typical mistakes of current MT systems can be linked to the lack of contextual coherence present in the followed translation approaches. As an example to illustrate this phenomenon, consider using an MT system to translate a news item in English about a claim process in some office. The word "desk" can appear several times and it can be translated into Spanish as "mostrador", "ventanilla", "escritorio", or "mesa". These Spanish words are not synonyms. Where "mostrador" and "ventanilla" can both be a counter where a service is offered, "mesa" and "escritorio" refer to a piece of furniture. So, "desk" is a word with ambiguous translation into Spanish. Within the context of our example, "mesa" and "escritorio" are not correct translations for "desk". We address this as a problem of contextual coherence, because the aim of our work is to use inter-sentence context to help the system to choose a more adequate translation without the need of any knowledge from the domain.

Another typical issue is word agreement across translation segments. Coreference chains confer cohesion to a document, and it is desirable to see this property projected into the produced translations. Unfortunately, this is a property that is typically difficult to maintain for MT systems. Also, gender and number agreement between words is sometimes challenging for current MT approaches. For example, consider the following set of sentences in a source document in English: "She studied civil

engineering. [...] The civil engineer was the youngest in the company." These sentences can be translated into Spanish as "*Ella* estudió ingeniería civil. [...] *El ingeniero* era *el* más joven de la empresa." This translation is correct in Spanish if we look at it sentence by sentence in isolation. However, it is incorrect if we consider it in its entirety as part of the same document, since there is no gender agreement between the translations of "the engineer" and "she". Taking document context into account, the correct translation would be "*Ella* estudió ingeniería civil. [...] *La ingeniera* era *la* más joven de la empresa."

Our work is motivated by the idea that exploiting discourse information would help to improve the quality of the resulting machine translations at document level. All the techniques we explore in this thesis attempt to find the best way to exploit such kind of information within the current MT frameworks.

1.2 Context

SMT systems (Koehn et al., 2007) were state-of-the-art when the work we present in this thesis started. Also, the first document-level MT approaches were being built on the same principles and could be seen as a direct evolution within the SMT systems (Hardmeier et al., 2012). Thus, since SMT systems were the dominant paradigm in MT, both at phrase and document level, most of our research was naturally planned using these approaches.

Nevertheless, NMT systems (Bahdanau et al., 2015) have rapidly risen to become the new leading paradigm of the MT area, obtaining noticeably better translations than SMT systems, especially with respect to their fluency (Wu et al., 2016) and in some cases claiming to reach human parity (Hassan et al., 2018), although there is still room for improvement (Toral et al., 2018).

Thus, we expanded the initial research plan for this thesis to introduce some of the approaches tested on SMT systems into an NMT decoding framework. It is important to reiterate that standard systems from both approaches are usually designed at sentence or phrase level, sharing the limitation of ignoring inter-sentence contextual information. Hence, context-aware enhancements for SMT are also desirable for NMT.

1.3 Research Goals

The general goal of the work presented in this thesis is to improve MT quality by exploring the use of document-level information at different steps of the translation process in order to fix or prevent some of the errors made by sentence-level MT systems. In particular, the main goal is to improve machine translation coherence and cohesion by leveraging the information given by the relations of the words along a document.

In order to achieve this goal, we define a research strategy with the following steps:

1. *Analyzing translation errors related to document-level phenomena and designing simple methods to tackle them.* A first step towards improving document-level machine translation is to identify those phenomena that confer coherence and

cohesion to documents and are susceptible to be lost in the MT process. Before exploring ways to solve such mistakes as part of the MT process, it is interesting to implement a set of simple post-processing techniques and evaluate their impact.

2. *Capturing the semantic information of a document in a useful manner to aid the MT decoding process.* Undeniably, leveraging a document's semantic context should help improve the coherence and cohesion levels of its translation. Thus, it is necessary to explore ways to introduce contextual semantic information into the MT process. In particular, our final intention is to *extend a document-oriented decoder to incorporate document context semantics.*

3. *Enhancing an NMT framework using context-aware techniques.* To finalize the work of this thesis, one of our goals is to integrate the explored ideas into the NMT paradigm.

1.4 Main Contributions

The main contributions of this thesis are directly related to the research goals described in the previous section. Our work has resulted in several published works, mainly as conference papers. In fact, much of the content of this thesis is an update or extension of the published papers. Our set of contributions and related publications is as follows:

- *Analysis of translation errors related to document-level phenomena and the development of a set of simple, yet effective, post-processing techniques to handle them.* Since the particular document-level phenomena they handle are sparse, we need a manual evaluation to assess their effectiveness because the automatic evaluation metrics do not capture their improvements. These findings were published as a technical report (Martínez Garcia et al., 2014b) and presented in the SEPLN2014 conference (Martínez Garcia et al., 2014a).

- *Demonstrating that bilingual word embeddings are capable of modeling semantic relations that help the SMT process.* We observe that the quality of the translation and alignments previous to building the semantic models are crucial for the final performance of the embeddings. Word embeddings prove to be helpful in the task of lexical substitution for words that are ambiguously translated within a document. This work resulted in a publication in the SSST-8 conference (Martínez Garcia et al., 2014c).

- *Showing that the introduction of bilingual word embeddings guides document-oriented SMT decoders towards more coherent and cohesive translations.* Although we only observe a slight improvement in the results of automatic evaluation metrics, the improvement is consistent among metrics and is larger as we introduce more semantic information into the system, getting the best results when using the models with bilingual information. This approach was presented in the EAMT2015 conference (Martínez Garcia et al., 2015).

- *Designing new strategies that guide document-oriented decoders through the translation search space towards more consistent, coherent, and cohesive translations, focusing primarily on maintaining lexical consistency.* Our strategies based on word embeddings aid the decoder to assess the compatibility of the possible translations for ambiguous words with their context. This extension led to participating in the EAMT2017 conference (Martínez Garcia et al., 2017).

- *Enhancing the NMT decoding algorithm to include contextual semantics captured by a language model based on word embeddings.* Our experiments show how the semantic language models can help NMT systems to produce better translations. Our approach does not need to modify the training process, so we do not increase the training time, or document-level annotated data. This work is under review for publication at the time of submission of this thesis.

1.5 Outline

The remainder of the thesis is organized as follows. In order to contextualize our research, Chapter 2 revisits the state-of-the-art of the MT research area, focusing on the main technologies of the SMT and NMT paradigms, both at sentence and document level. Chapters 3 to 6 present the results of our research.

In Chapter 3, we analyze some of the translation errors related to document-level phenomena (Section 3.1) and present a set of post-processing strategies to handle them (Section 3.2).

Afterwards, in Chapter 4, we describe how to introduce word embeddings for decoding. First, we study the applicability of word embeddings to enhance the MT process in general (Sections 4.1 to 4.3). Then, we explain a method to enhance a document-oriented SMT decoder with word embeddings working as Semantic Space Language Models (Section 4.4).

Next, Chapter 5 describes our extension of a document-oriented SMT decoder to handle the particular document-level phenomenon of lexical choice consistency for a translation. We present a new feature function that guides the decoder towards more lexically consistent translation candidates (Section 5.2), as well as a new strategy to shortcut the exploration of the search space (Section 5.3).

Chapter 6 presents our approach to extend the usual NMT decoding process to take into account contextual semantics. In particular, we extend the beam search decoding algorithm by fusing the discourse information captured by the models we describe in Chapter 4 to work in tandem with the NMT model.

Finally, Chapter 7 draws the conclusions of the work of this thesis and describes possible avenues of future work.

Additionally, Appendix A describes a new document-level decoding strategy based on a swarm optimization algorithm, integrated into the decoder used in Section 4.4 and Chapter 5 as an alternative to its default hill climbing strategy.

Chapter 2

State of the Art

2.1 Machine Translation Background

Machine Translation is one of the earliest problems in Natural Language Processing and Artificial Intelligence. The origins of MT as a field itself can be dated in the late 1940s, with the end of World War II and the birth of the first electronic computers in the United States. In the 1950s, the original MT systems were very simple but obtained promising results. However, after the initial euphoria, there was an increasing acknowledgment of the linguistic difficulties involved that produced a lack of productivity in the area during the 1960s. This discouraging atmosphere ended up with the Automatic Language Processing Advisory Committee (ALPAC) report in 1966 (Pierce and Carroll, 1966). In such document, MT was qualified as useless and stated as a slower and more expensive procedure than human translation. Therefore, there was a recommendation not to invest in MT but to do it in other more basic NLP tasks instead. It was not until the late 1980s, with the rise of more powerful and faster computers, that the MT field emerged again. Statistical approaches appeared in the 1990s becoming the most successful MT systems until the rise of neural architectures in 2015.

MT approaches can be classified according to different criteria. According to their usage, there are systems designed for machine-aided translation, both for *human translation with machine support* and for *machine translation with human support*. On the other hand, there are *fully automated translation systems*, which typically prioritize speed over quality.

Regarding the level of linguistic analysis performed, MT systems can be classified in *direct*, *transfer*, and *interlingua*. The Vauquois triangle of Figure 2.1 shows this classification. The *direct* approach performs a word-by-word or phrase-by-phrase translation. The *transfer* approach makes a syntactic and/or semantic analysis of the input to build an abstract representation of the source. This abstract representation is then transferred to the abstract representation of the target language, from which the output is generated. The *interlingua* approach is similar to the *transfer* approach

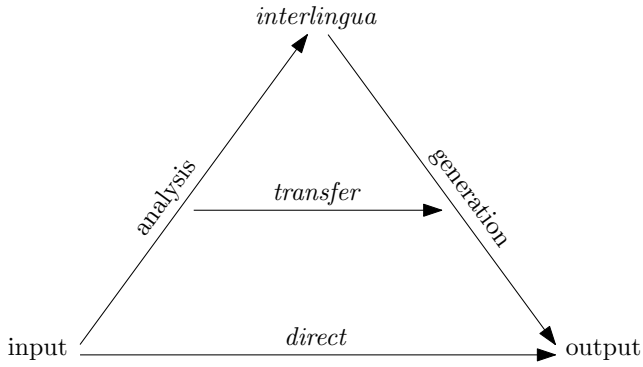

Figure 2.1: The Vauquois Triangle for the classification of MT systems according to the level of linguistic analysis.

but using a single abstract representation common to all languages.

According to the architecture of the system, we can distinguish between *rule-based systems* and *empirical systems*. *Rule-based systems* (RBMT) use a set of rules to describe the translation process. Typically, these rules are established by a group of human experts and drive the translation process. Although these systems obtain high quality syntactics of the translated output, the process followed is generally slow, expensive, not portable, and language dependent. RBMT systems are usually characterized also by a linguistic transfer process from a syntactic or semantic analysis.

On the other hand, *empirical systems* are data driven. They get knowledge automatically, typically from a sentence-aligned parallel corpus. Although it is not a common procedure, these systems can also include some linguistic analysis too. Since these systems learn automatically from data, there is no need for human interaction at least at translation time. *Example-based Machine Translation* (EBMT), *Statistical Machine Translation* (SMT), and *Neural Machine Translation* (NMT) systems are all *empirical systems*. Briefly, the EBMT systems build new translations using translations compiled previously as a basis. SMT systems consider that each sentence of the target language is a possible translation of a sentence in the source language and assign a probability to each of them. NMT systems use artificial neural networks to predict the likelihood of a sequence of words in the target language given a sequence of words in the source language.

2.1.1 Statistical Machine Translation

SMT systems assign a probability to every possible translation for each sentence and choose the final translation by finding the one that maximizes this probability. Formally, SMT systems estimate the probability of a target sentence $y = (y_1, \ldots, y_N)$ being the translation of a source sentence $x = (x_1, \ldots, x_M)$, and find the best translation $\hat{y}$ by selecting the target sequence that maximizes such probability (Brown et al., 1990):

$$\hat{y} = \arg\max_{y} p(y|x)$$

Making an analogy with the noisy channel model (Shannon, 1948), that posterior probability is rewritten with the Bayes' rule into:

$$\hat{y} = \arg\max_{y} p(x|y)p(y)$$

having already removed the divisor $p(x)$ as it does not affect the result of arg max. In this reformulation, $p(x|y)$ is the reverse translation probabilistic model. In other words, it represents the probability of seeing the sequence x of source words as the translation of the sequence y of target words. Also, the factor $p(y)$ represents a Language Model (LM) that estimates the probability of seeing the sequence of words y in the target language. This objective probability can be decomposed using a log-linear model:

$$\log p(y|x) \propto \log p(x|y)p(y) = \log p(x|y) + \log p(y)$$

and re-written in a more general way as the sum of different feature functions f_i that represent probabilistic models that capture different linguistic aspects:

$$\log p(y|x) = \sum_i w_i f_i(x, y) + C$$

The probabilistic models f_i are trained on large parallel or monolingual corpora, depending on the feature characteristics. In turn, the weight parameters w_i are tuned on smaller parallel development sets using standard generic algorithms to that end, such as MERT (Och, 2003), PRO (Hopkins and May, 2011), or adaptations of MIRA (Crammer et al., 2006) for SMT (Chiang et al., 2008; Watanabe et al., 2007). These probabilistic models can be applied at different translation unit levels: word, phrase, or even sentence.

The basic structure of a phrase-based SMT system (Koehn et al., 2003) is shown in Figure 2.2. First, there is a *pre-processing* step where the source text is normalized and tokenized. Then, there is a process of *phrase extraction* from the parallel corpus used to train the system. In this stage, the source language phrases are *aligned* with their corresponding target language ones. From these alignments, a probabilistic *translation model* is built capturing the information about the most probable translations of a given source phrase. Each of these translation pairs is assigned a probability, estimated by frequency counts in the training corpus. In a similar way, a stochastic *language model* is derived from a monolingual corpus to represent the information of the target language that must be present in a good translation output. Using that information, the *decoder* builds several possible translations from a source sentence and ranks them using a scoring function. This function can be defined using different language features depending on which characteristics to reinforce in the translation system. Finally, there could be a post-processing step where some translation errors can be fixed.

The most commonly used decoding algorithm in phrase-based SMT is the *beam search* (Koehn et al., 2003). It constructs the translation of each sentence incrementally: it starts with an empty hypothesis and expands it iteratively with possible phrases, thus producing diverging hypotheses of varying quality. Since the amount of possible hypotheses grows exponentially as the sentence building advances, extensive

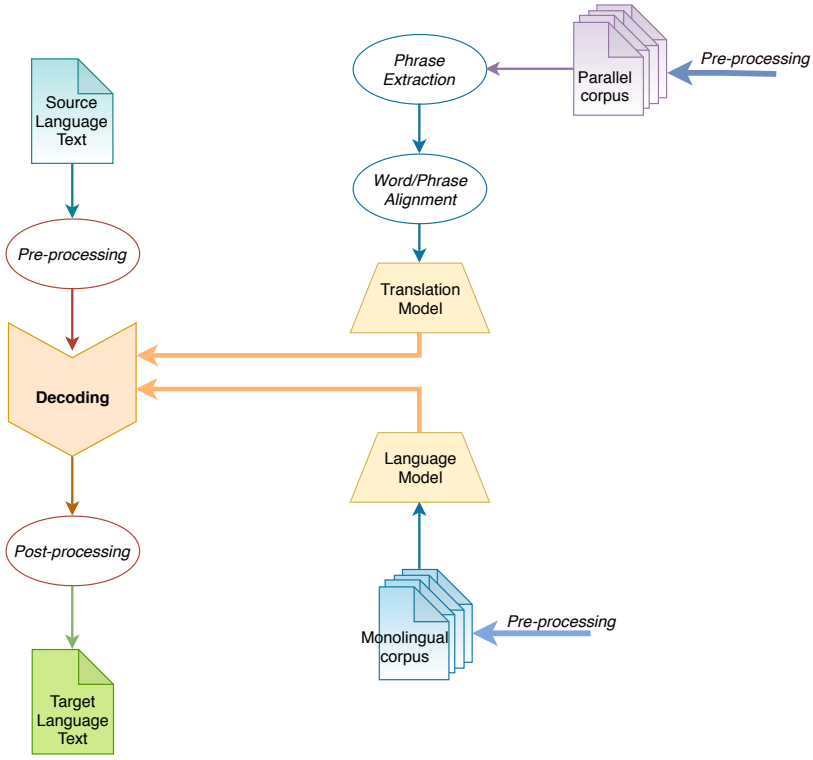

Figure 2.2: Structure of a state-of-the-art SMT system.

pruning of the search space must be performed. To this end, two distinct techniques are applied. First, groups of similar hypotheses are merged in a recombination process like the one proposed by Och et al. (2001), which consists in discarding the hypothesis with the worst score from any pair of hypotheses that cannot be distinguished according to the language and translation models. In particular, indistinguishable hypotheses are essentially those that have translated the exact same part of the input sentence and their trailing translated words coincide. And second, only a handful of the hypotheses remaining after the recombination is retained for further expansion. In particular, those that seem most promising according to the score of the already constructed part and an estimated score for the remaining part. The set of hypotheses the algorithm works on conforms the *beam*, which is of a fixed size, usually small.

2.1.2 Neural Machine Translation

Although SMT systems have shown several advantages, such as training speed and easy adaptation to new domains, with the pass of time their limitations have also been identified, such as the generation of local translations that do not model entire sentences.

Neural Machine Translation arises from the comeback of deep learning methods

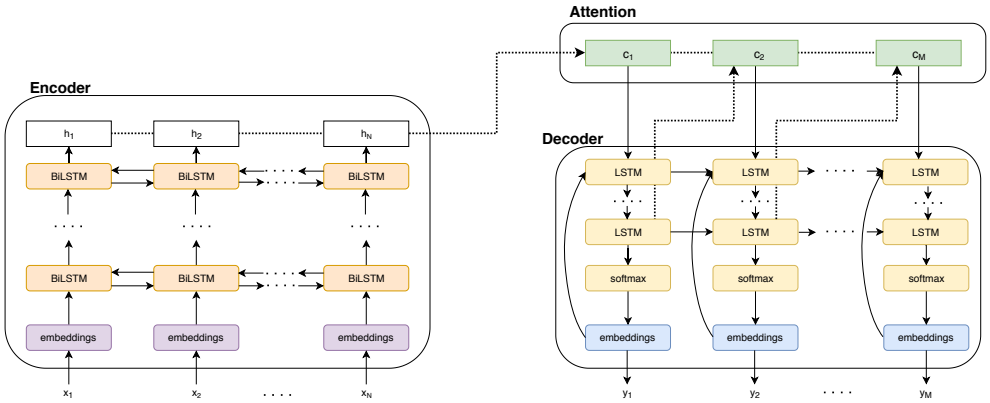

Figure 2.3: Encoder-Decoder NMT system with Attention mechanism.

based on artificial neural networks (NN). Thus, NMT is based on developing neural-based end-to-end translation systems.

Some first attempts of pure NMT appeared in the late 1990s in Spain. Forcada and Ñeco (1997) proposed a recursive hetero-associative memory (RHAM) model that was able to learn general translations from examples. Also, Castaño and Casacuberta (1997) proposed an NN-based system with promising results. However, these approaches reported that the size of the neural networks required for the MT task, and thus also their training time, was excessive for the computation power available at that time.

Modern pure NMT systems appear in the 2010s integrating word embeddings (Mikolov et al., 2013a,c) and neural language modeling techniques into MT systems. The first approaches tried to combine neural components with SMT approaches either in a chain (Schwenk et al., 2006) or introducing them as a new component of the MT systems (Devlin et al., 2014). End-to-end neural MT systems are designed from the "sequence-to-sequence" perspective. They are devised to be able to generate an output sequence from an input one. Sutskever et al. (2014), Cho et al. (2014a,b), and Bahdanau et al. (2015) proposed systems that implement a neural architecture based on two main components: the *Encoder* and the *Decoder* (see Figure 2.3). The *Encoder* projects a source sentence with variable length into a set of continuous vectors with a fixed length, and then, the *Decoder* takes this continuous representation to generate a target sentence. These systems are trained to maximize the conditional log-likelihood of the bilingual training corpus:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} \log p_\theta(y_n|x_n)$$

where θ is the set of the model parameters. For an encoder-decoder model the conditional probability of the next word is:

$$y_t = \text{softmax}\left(\text{DNN}\left(\vec{H}_t^{TM}, y_{t-1}, \vec{C}_t\right)\right) \tag{2.1}$$

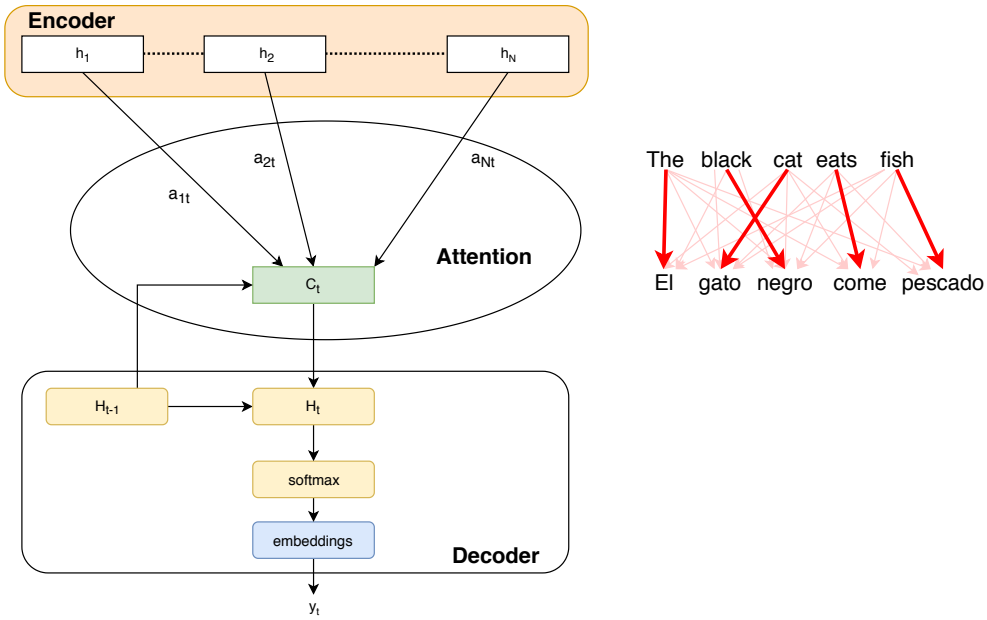

Figure 2.4: Attention mechanism functionality as described by Bahdanau et al. (2015). The example depicts the soft alignments handled by the NMT system on a particular sentence.

where DNN represents the deep neural network that conforms the decoder, $\vec{H}_t^{TM}$ is the hidden state from the decoder, and $\vec{C}_t$ is the context vector from the encoder. The context vector is the encoder hidden state and represents a summary of the whole input sentence. Finally, the output is a softmax layer that draws a probability distribution over the target vocabulary. In these approaches, the encoder and the decoder are built using Recurrent Neural Networks (RNNs) (Bengio et al., 2013). These neural nets are designed specifically to deal with sequence problems. They are able to take into account the previously processed/generated context before generating the next step in a sequence. RNNs are mainly built using LSTM (Hochreiter and Schmidhuber, 1997) or GRU (Cho et al., 2014a) neural units. Both of these neuron types include gating mechanisms that allow the neurons to remember the information from the previous step with a certain probability.

Introducing bidirectional networks in the NMT systems architecture shows a significant improvement. The NMT architecture by Bahdanau et al. (2015) includes a bidirectional encoder (BiEncoder) that projects the source sentence into a continuous vector by processing it from left to right but also from right to left, capturing in this way sentence context from both sides of sentence words.

The main contribution to NMT systems was the introduction of the *Attention Mechanism* (Bahdanau et al., 2015). This module regularizes the output projection by learning soft alignments among the source sentence words and the generated target words. So, the system not only learns to translate, but it also learns a probability

distribution of the alignments among the input and the output words. Figure 2.3 depicts the attentional encoder-decoder NMT architecture, and Figure 2.4 shows how the attention mechanism relates source and target words through a probability distribution. For these systems, the probability of the following predicted word in a sequence is calculated as described in Equation 2.1 but being $\hat{C}_t$ a weighted sum of the *annotations* where the encoder maps the input sentence. In particular, the context vector is computed as follows:

$$\hat{C}_t = \sum_{j=1}^{n} \alpha_{tj} \vec{h}_j$$

with α_{tj} being the weight:

$$\alpha_{tj} = \frac{\exp(e_{tj})}{\sum_{k=1}^{n} \exp(e_{tk})}$$

where the soft-alignment weights e_{tj} reflect how well the inputs around position j and the output at position t match. They are computed by:

$$e_{tj} = a(\vec{H}_{t-1}^{TM}, \vec{h}_j)$$

depending on the hidden state $\vec{H}_{t-1}^{TM}$ previous to generating y_t and on the jth annotation $\vec{h}_j$ of the input sentence. The alignment model a is a feed-forward neural network which is trained jointly with the rest of the components of the system. These attention weights then perform normalization on the output taking into account the alignment information among source and target words.

Typically, in order to generate the target sentence, the NMT systems implement a beam search algorithm to explore the space of translation candidates and finally propose the sequence that maximizes the translation probability.

The latest NMT systems propose the use of encoder-decoder implementations based on the Attention Mechanism (Vaswani et al., 2017). In particular, the Transformer architecture claims that attention is all you need to have a good MT system. The Transformer also follows the encoder-decoder architecture, but it does not include any recurrent or convolutional network. Instead, this approach implements a set of techniques that allow knowing the sequence. First, it employs positional embeddings to be aware of the position of the tokens in the sequence. Also, it includes position-wise feed-forward layers, that can be understood as two convolutions with a kernel of dimension 1. Finally, it implements multi-head self-attention layers. These layers allow attending to information that comes from different representation subspaces, acting as a set of attention layers working in parallel. The Transformer is the NMT architecture that achieves the best results across many language pairs nowadays, producing very fluent translations and reducing training times. However, it seems to be sensitive to parameter tuning and to require high usage of memory when translating (Popel and Bojar, 2018).

Independently of their neural architecture, NMT systems cannot afford an open vocabulary translation, i.e., they can only perform translations managing a fixed source and target vocabularies. One of the most used approaches to solve this problem is sub-unit segmentation by Byte Pair Encoding (BPE). Sennrich et al. (2016) apply this encoding algorithm to segment infrequent words without taking into account

any linguistic information, just their character pair frequencies. Some morphological based segmenting approaches exist (Cotterell et al., 2015; Virpioja et al., 2013) that have proved to be effective when dealing with morphologically rich languages like Turkish (Ataman et al., 2017), Basque (Etchegoyhen et al., 2018), or Finnish (Ding et al., 2016). However, handling open vocabularies is still an open problem for NMT systems.

NMT systems have proved their high performance in a short time, providing more fluent translations than SMT systems. However, they also have several limitations. For instance, they need a huge amount of data to build translation models with good performance and they seem to be more sensitive to data quality than SMT systems (Khayrallah and Koehn, 2018; Koehn and Knowles, 2017). Also, similarly to SMT approaches, they are typically designed to perform translations at sentence level. They do take into account intra-sentence context information due to the vector projections they handle, but they are not capable of transferring information across different sentences, as they handle each sentence in isolation.

2.2 Document-Level Machine Translation

SMT and NMT systems typically translate documents sentence by sentence, separately, ignoring many document-level phenomena. RBMT systems define their transfer rules at most at the sentence level. Thus, although any document usually follows a structure, they are generally translated as a collection of independent sentences, ignoring in the process the document-level information. This behavior is pervasive in the current MT approaches.

Limitations of the sentence-by-sentence translations appear in coherent discursive pieces as news, encyclopedic articles, etc. Potential advantages of working at document level in these cases could be, for instance, the resolution of lexical inconsistencies between translations of the same word and the agreement among coreferent mentions of the same entity. Lexical ambiguities could be better resolved by using the information provided by the surrounding sentences since topical coherence should be maintained (e.g., if the word “desk” is translated under an administrative topic as “ventanilla” or “mostrador”, it makes no sense to translate it as “mesa” or “escritorio” in Spanish).

Moreover, it is difficult to correctly translate coreferent mentions of an entity without knowing its antecedent one. For instance, if the following sentences appear in a source document “Maria won the first prize. She was very happy. The winner was the youngest one.”. It is easy to see that the words “*Maria*”, “*She*”, and “*The winner*” define a coreference chain, and must agree in number and gender. One of its possible correct translations into Spanish is “Maria ganó el primer premio. Ella estaba muy contenta. La ganadora era la más joven.”, where the corresponding target words in the projected coreference chain are coherent in gender and number. However, using a state-of-the-art SMT system[1] we obtain the following translation “Maria ganó el primer premio. Ella estaba muy feliz. *El ganador* fue el más joven.”, where the agreement in gender is lost in the last sentence.

[1] By using Google Translate or Bing Translator.

These are two classic examples of the need for translating a document as a whole, but the benefits should increase when handing more phenomena related to the discourse information within a document.

2.2.1 Document-Level Statistical Machine Translation

The beam search algorithm presented by Koehn et al. (2003) and implemented in the Moses decoder (Koehn et al., 2007) is considered the state of the art in the SMT area. Nevertheless, this approach translates each sentence independently, ignoring its surrounding context within the document. Furthermore, adapting the beam search to consider linguistic phenomena that go beyond sentence boundaries, such as coreference or discourse markers, is quite challenging.

Beam search uses the assumption of sentence independence at its very core: it is key to prune the search space. This pruning is done while decoding and consists of recombining partial translation hypotheses that seem similar and discarding those that have a low estimated score. To select the recombinations and to compute the estimations it is necessary that the scoring function can be computed by analyzing just a small preceding context, without exploiting inter-sentence dependencies. For this reason, works based on beam search that focus on document-level phenomena have usually resorted to using ad-hoc workarounds to the sentence independence assumption. For instance, pronominal anaphora is tackled by Le Nagard and Koehn (2010) by, previous to decoding, performing an automatic annotation of pronoun genders by resolving coreferences with preceding sentences of the source document. Similarly, pronominal anaphora is approached by Hardmeier and Federico (2010) with a driver that annotates pronouns with gender and number before decoding each sentence. The annotation is performed by leveraging the already translated sentences, but allowing efficient parallel decoding between sentences without coreference dependencies. Limitations of both methods are discussed by Guillou (2012). Tiedemann (2010) employs a cache of the word translations used in preceding sentences to influence the decoding decisions of the next sentence. A drawback of this technique is that the cache increases the propagation of translation errors. More refined cache techniques are used by Gong et al. (2011) and Louis and Webber (2014) for topic cohesion. Carpuat (2009) and Xiao et al. (2011) address lexical consistency with a post-process to re-translate source words that were inconsistently translated within a document. Additionally, Xiao et al. (2011) also propose an alternative two-pass decoding where, previous to the second decoding pass, undesirable translation options in the phrase table are removed. Similarly, a two-pass technique is also used by Ture et al. (2012) for lexical consistency but, instead of pruning the phrase table as Xiao et al. (2011), the second decoding pass uses additional features that analyze counts obtained in the first pass.

Alternatively, several works have extended the model of Koehn et al. (2003) by, instead of performing workarounds to its limitations, replacing the beam search algorithm altogether. For instance, Arun et al. (2010) use a Gibbs sampler to draw samples from the posterior distribution. The sampler consists of three operators that, applied probabilistically, explore the distribution. This allows, in particular, more general feature functions in the scoring. Nevertheless, as with the approach of Koehn et al. (2003), this method assumes sentence independence. Moreover, since

all possible ways of applying the sampler operations are considered at each iteration of the process, the cost of treating full documents as translation units would be prohibitively high. Langlais et al. (2007) propose a so-called greedy decoder. It is, in fact, a decoder based on local search that performs a steepest-ascend hill-climbing strategy at sentence level. The idea is to produce an initial translation for the sentence and then iteratively refine it into a local optimum. The initial translation is either obtained by the beam search algorithm of Koehn et al. (2003) or by segmenting the source sentence into the minimal amount of segments and choosing for each of them the translation option with the highest score in the phrase table. At each iteration, a new translation for the sentence is produced from the current one by taking the best translation in its neighborhood, i.e., the best among all possible translations derived from the current one with a single, simple modification. This approach allows the decoder to have features with fewer restrictions than the one by Koehn et al. (2003) since the full sentence translation is available for scoring. Nevertheless, it is still not possible to have document-level features: it is not feasible to handle full documents as translation units since each step of the decoding process explores the whole neighborhood, which is in general excessively large for full documents. Hardmeier et al. (2012) solve this drawback by changing the steepest-ascend hill-climbing strategy for a first-choice hill-climbing one. In this way, the neighborhood is not fully explored at each step and instead, it is randomly enumerated until finding a better translation (or until exhausting a maximum amount of tries). Thanks to this, the decoding is able to handle full documents as translation units, and thus, it is a suitable framework for integrating features capturing properties of document-level phenomena. Since this approach is key to our work presented in part of Chapter 4 and in Chapter 5, we describe it with greater detail in Section 2.2.2. More recently, Douib et al. (2016) use a genetic algorithm (Holland, 1973) for decoding full documents. This approach bears some resemblance to the one by Hardmeier et al. (2012): the process starts by randomly generating a set of translations and then iteratively improving them by randomly exploring their neighborhoods. Nevertheless, genetic algorithms also provide an operation that allows escaping the vicinity of a neighborhood: the crossover operation randomly combines two translations to obtain a new one, composed entirely with pieces of either translation.

2.2.2 Docent and Lehrer

Hardmeier et al. (2013) introduced the Docent[2] document-level decoder implementing the local search algorithm described by Hardmeier et al. (2012) and building on the phrase-based SMT model of Koehn et al. (2003). This local search follows a first-choice hill-climbing strategy and it is performed in a space that can be seen as a graph: nodes are full-document translations and an edge connects two nodes when one translation can be transformed into the other by applying a single change operation (see Figure 2.5). More precisely, the search proceeds as follows. Initially, a node is chosen as a starting point. Then, at each iteration, the decoder tries to move into a node adjacent to the current one, but only performs the move when the score of the destiny adjacent node strictly improves the score of the current one. The adjacent

[2]Source code available at: `https://github.com/chardmeier/docent/`

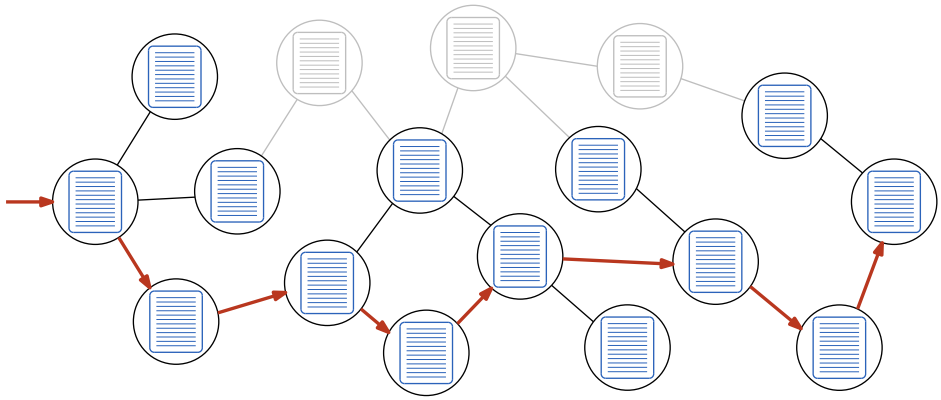

Figure 2.5: Example depiction of DOCENT's search space and the path it follows through it (starting at the left-most node, proceeding along the thick, red-colored edges, and ending at the right-most node, which corresponds to the final refined translation). The three greyed-out nodes at the top are not considered during the search, since they are not in the immediate neighborhood of any of the visited nodes.

nodes are tried in random order,[3] possibly with repetitions. The process terminates when a limit on the amount of iterations is reached, or when a certain amount of successive iterations have not found a suitable adjacent node to move into. Therefore, the three basic ingredients of the search algorithm are (*i*) the selection of the starting node, (*ii*) how adjacent nodes are obtained, and (*iii*) the scoring function that guides the search.

For the selection of the starting point, a *deterministic* and a *random* method are proposed. The former consists in simply using MOSES to construct an initial translation. The latter consists of constructing a translation by, first, randomly segmenting each sentence of the source document into phrases that occur in the underlying phrase table, and second, for each such segment randomly choosing a translation option from the phrase table.

For obtaining adjacent nodes, the decoder randomly applies to the current translation one of the change operations available, thus constructing a translation belonging to its neighborhood. Three basic change operations are provided: (*i*) *change-phrase-translation* changes the translation of a segment by another one in the phrase table, (*ii*) *swap-phrases* swaps in the target the order of the translations corresponding to two segments of the same source sentence, and (*iii*) *resegment* takes a contiguous span of segments of the same source sentence, satisfying that their corresponding translations also appeared contiguously in the target, segments anew the whole span and, for each resulting segment, chooses a translation option from the phrase table. Note that, in all the cases, the minimal translation units handled by the decoder are

[3]The traditional definition of hill climbing does not involve random decisions. In this case, however, the neighborhood is too big to enumerate, and thus, it is simply explored randomly until finding one translation with a higher score or exhausting the quota of tries.

phrases from the phrase table. Also notice that, by repeated applications of these three change operations, any translation can be transformed into any other translation, i.e., the available change operations guarantee that the whole search space is reachable regardless of the starting point.[4] Nevertheless, since the search uses a hill-climbing strategy, the starting point does determine which local maxima are attainable. To be able to search the whole space and reach global maxima, a simulated annealing strategy (Kirkpatrick et al., 1983) could be used instead. This is discussed by Hardmeier (2014), who reports problems with using such strategy.

For the scoring, the decoder uses the log-linear model from Koehn et al. (2003). In particular, the score of a translation is computed as the (weighted) addition of several feature functions. Seven basic features are provided, implemented to be compatible with the analogous ones in Moses; roughly: (*i*) *geometric-distortion-model* favours translations that closely follow the phrase order from the source document, (*ii*) *word-penalty* favours translations with fewer words, (*iii*) *oov-penalty* favours translations with fewer out-of-vocabulary words, (*iv*) *phrase-penalty* favours translations with more phrase pairs, (*v*) *ngram-model* favours translations that employ usual constructions of the target language, and (*vi*) *phrase-table* favours for each segment of the source the most usual translations. The latter feature is the only obligatory one, as it defines the underlying phrase table that determines the search space. Note that all these basic features assume sentence-independence, as they are inherited from Moses. Nevertheless, the decoder also takes advantage of the fact that the full-document translation is available for the scoring, and provides some additional document-level features.

In the experiments carried out within this thesis, an in-house re-implementation of this decoder was used. As a homage to Docent, it was named Lehrer.[5]

2.2.3 Decoding with Ants

As an alternative to the simulated annealing strategy discussed by Hardmeier (2014) to generalize the hill-climbing decoding strategy of Hardmeier et al. (2012), we have explored the benefits of using a new strategy building on a different metaheuristic for optimization problems: ant colony optimization (Colorni et al., 1992; Dorigo et al., 1996), ACO for short. Intuitively, our method[6] consists in reducing the problem of finding a translation that optimizes the score function to the problem of finding an optimal path through a graph. Similar reductions have already been considered in the literature, e.g.: Knight (1999) relates MT decoding to the traveling salesman problem. Tackling the construction of an optimal path by means of ACO proceeds as follows (see Figure 2.6). Iteratively, sets of paths (each path corresponding to a full-document translation) are randomly constructed by ants walking through the graph (each of its nodes corresponding to a phrase from the phrase table). The randomness

[4] In fact, this property already holds with *swap-phrases* and *resegment* alone. Moreover, note that *change-phrase-translation* is subsumed by the case where *resegment* operates on a span of one single segment.

[5] The word "lehrer" is German for "teacher". Source code available at: `https://www.cs.upc.edu/~emartinez/lehrer.tgz`

[6] Source code available at: `https://www.cs.upc.edu/~emartinez/lehrer-aco.tgz`

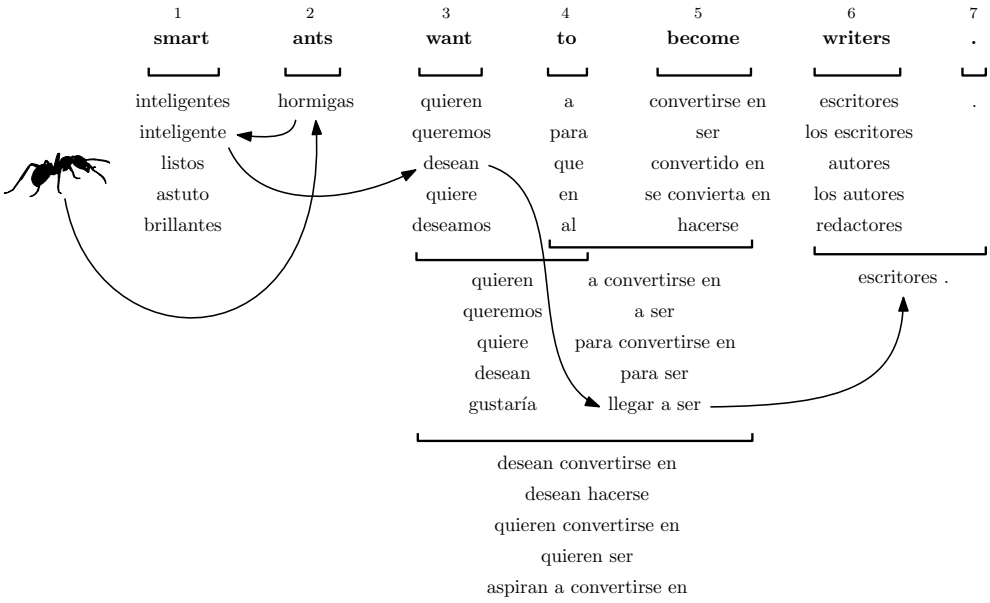

Figure 2.6: Sketch of the graph constructed for the source sentence "*smart ants want to become writers.*" (each of the Spanish phrases corresponds to a node of the graph; edges are omitted to avoid clutter) and path that an ant follows through it to form the translation "*hormigas inteligente desean llegar a ser escritores.*" with some gender and number disagreements. For each phrase of the source sentence we show up to 5 translations, listed by decreasing probability. Note that only 3 two-token phrases ("*want to*", "*to become*", and "*writers .*") and 1 three-token phrase ("*want to become*") have translations. Also note that the nodes visited in the path translate each of the 7 source tokens, and do it just once.

of the ant's movements is influenced by the amount of pheromone in the graph, which gets updated at the end of each iteration: the pheromone amount is increased along the best discovered paths, whereas in the rest of the graph it is decreased. Eventually, the distribution of pheromone is expected to evolve such that all the ants converge to the same, optimal path through the graph. Nevertheless, unless certain conditions are met, it cannot be guaranteed that the method will actually converge to the global optimal solution. If global optimality is not strictly required and just an approximation is enough, ACO has been shown to obtain good solutions in competitive runtime for many problem domains (Dorigo and Blum, 2005).

Unfortunately, the experiments conducted to assess this new decoding strategy have ultimately been unsuccessful: we have been unable to outperform a baseline Moses. Furthermore, in the best results we have achieved, we used the ACO decoding variants that closest resemble the hill-climbing approach of Hardmeier et al. (2012), but with worse memory requirements and runtime in our case. Our prototype implementation is not fully optimized yet and has some margin for improvement (e.g.,

the highly parallelizable nature of ACO could be further exploited), but nevertheless, it is unlikely that we could get any significant asymptotic improvement.

Appendix A presents a detailed description of the approach, together with a report on the conducted experiments and the obtained results.

2.2.4 Document-Level Neural Machine Translation

NMT systems have proved their good performance in a short time, beating SMT systems broadly. However, they inherited the limitation of SMT systems of not being able to handle extra-sentence context information since they are typically designed to treat each sentence in an isolated and independent way.

The interest in making NMT systems able to include wider context information in the translation process has increased in recent years (Jean et al., 2017; Popescu-Belis, 2019), exposing the necessity of exploring new approaches of document-level machine translation (Läubli et al., 2018).

There are several approaches that tried to extend the context beyond the sentence information by modifying the system's input. Tiedemann and Scherrer (2017) concatenate the previous source sentence to the current one, whereas Bawden et al. (2018) also concatenate the previous predicted target sentence. More sophisticated approaches propose to modify the NMT model itself to make it able to handle context information beyond the sentence scope.

Wang and Cho (2016) present an approach to include document-level context into language modeling by implementing fusion approaches that help the LSTM maintain separated the inter- and the intra-sentence context dependencies. They report that using a wider context helps a neural LM capture better the semantics of a document.

There are several approaches that extend the context handled by an NMT taking into account the previously encoded source sentences. Jean et al. (2017) analyze whether NMT systems can also benefit from larger contexts. They propose a variation of an attentional RNN NMT system to model the surrounding text in addition to the source sentence. They include an additional encoder and attentional model to encode as context sentence the previous source sentence. Wang et al. (2017a) propose a cross-sentence context-aware approach that integrates the historical contextual information within the NMT system in three different ways: by initializing the encoder or the decoder or both with the history representation, by using the history representation as static inter-sentence context in combination with the source sentence context produced by the attention model and, finally, by adding a gating to the amount of context information used to generate the next word. However, these approaches only extend the source context but ignore the target side context.

In contrast, Tu et al. (2018) take into account the target side context by using a lightweight cache-like memory network which stores bilingual hidden representations as translation history, showing the utility of using this context information.

More recent approaches implement system extensions that handle both source and target side contexts. While Maruf and Haffari (2018) extend an RNN-based NMT system using memory networks to capture global source and target document context, Voita et al. (2018) present a variation of the Transformer that extends the handled context by taking in the input both the current and previous sentences. And Jean

and Cho (2019) extend it by including a context-aware regularization.

The importance of document-level neural machine translation is also seen in the recent WMT2019[7] news translation shared task, where for the first time a specific track for document-level MT was included. The systems presented at the shared task follow the previously explained strategies: introducing the inter-sentence context information into the NMT system by augmenting the training data including document-level information, i.e., including coreference information (España-Bonet et al., 2019), or just by increasing the training-sequence length in order to capture a larger data context (Junczys-Dowmunt, 2019; Popel et al., 2019; Talman et al., 2019), or introducing variations in the NMT architecture to take into account document-level information (Stahlberg et al., 2019; Talman et al., 2019).

In summary, many NMT approaches have been explored to include document-wide context or, at least, to make an NMT system able to handle a wider context than the intra-sentence one. Some of these approaches only exploit the context information from the source side ignoring the valuable information from the target side, although more recent variation proposals also include the target context information. The most successful approaches propose complex variations in the NMT models by including new document-oriented modules or regularization mechanisms to model document context. However, they do not show much success in exploiting the inter-sentence context information, presenting relative improvements in the final translation quality.

2.3 Automatic Evaluation

Automatic evaluation of machine translation quality deals with computing the similarity between an MT system's output and one or several reference translations for a given source text. Automatic machine translation evaluation metrics are not only useful to provide a quality measure for machine translation results but also are an important guidance for MT development and tuning.

The first approaches for automatic MT evaluation were based on lexical similarity, that is, designing lexical measures. These measures work by rewarding lexical matches between automatic translations and a set of reference translations.

The most popular and representative measure here is BLEU. This measure has been widely accepted as a de facto standard for years but it has several well-known drawbacks:

1. It has been shown that lexical similarity is neither a sufficient nor a necessary condition for two sentences to convey the same meaning (Callison-Burch et al., 2006; Coughlin, 2003; Culy and Riehemann, 2003).

2. It is also the case that BLEU and current BLEU-like metrics (Doddington, 2002; Lavie and Agarwal, 2007) perform well on low-quality machine translation results, but worse for high-quality ones.

3. It has been shown that BLEU has trouble distinguishing raw, inadequate machine translation output from fully fluent and adequate translation obtained from them through professional post-editing (Denkowski and Lavie, 2012b).

[7] `http://www.statmt.org/wmt19/translation-task.html`

4. In particular, string-based metrics are not able to capture the syntax or semantic structure of sentences; therefore, they are not sensitive to the improvement of machine translation systems on these aspects.

5. The reliability of lexical metrics depends very strongly on the heterogeneity and representativity of reference translations.

6. String-based metrics tend to favour statistical MT systems when compared to rule-based MT or other paradigms on a particular data set.

In order to cope with these issues, a number of authors have suggested exploiting linguistic information beyond the lexical level to increase robustness. Some have used additional linguistic knowledge to extend the reference lexicon. For instance, ROUGE, METEOR, and TER allow for morphological variations via stemming. TER and METEOR may perform an additional dictionary-based lookup for synonyms and paraphrases (Denkowski and Lavie, 2012a; Snover et al., 2009b). Russo-Lassner et al. (2005), Kauchak and Barzilay (2006), Owczarzak et al. (2006), and Zhou et al. (2006) have also studied the use of automatically-generated paraphrases to find potential phrase matchings. Surprisingly little work has actually been done in tuning the parameters of automatic evaluation metrics to correlate with actual assessments of quality, an exception being the work by Denkowski and Lavie (2012a) just cited.

More recent approaches have been designed focusing on performing a semantic evaluation rather than only looking for lexical and syntactical features, resulting in more adequacy-oriented evaluation metrics. The MEANT metrics family (Lo, 2017; Lo and Wu, 2011; Lo et al., 2014) compute the similarity of the semantic frames and their role fillers between the human reference and machine translations, relying on semantic parsers, and some of them weighting the importance of a word by inverse document frequency when computing the phrasal similarity score. However, although these metrics correlate well with human accuracy judgements, they are not widely used. Furthermore, there exist also Adequacy-Fluency oriented metrics (Banchs et al., 2015; D'Haro et al., 2019) that are designed, although at sentence level, to take into account the syntactic and the semantic information in a decoupling way in order to provide a more balanced view of the translations quality.

2.3.1 Automatic Evaluation Metrics

Throughout this thesis, the Asiya[8] toolkit by Giménez and Màrquez (2010) and Gonzàlez et al. (2012) is used to carry out the automatic evaluations. Depending on the experiment, we use a certain selection of the following metrics on lexical similarity:

- Three metrics that compute some variant of the edit distance by Levenshtein (1966). First, the *word error rate*, WER (Nießen et al., 2000), is the minimal amount of changes (i.e., substitutions, deletions, and insertions of tokens) needed to transform the generated translation into the reference. Second, the *position-independent word error rate*, PER (Tillmann et al., 1997), is similar to WER but does not take into account the token order in the sentences. And

[8] http://asiya.cs.upc.edu/

third, the *translation edit rate*, TER (Snover et al., 2006, 2009a), is also similar to WER but adding as a possible change to shift forward or backward a phrase within the sentence. Furthermore, TER performs stemming and synonymy lookup when matching tokens from the translation to the reference. We denote by TER_{base} the version that performs only exact matching.

Since the previous metrics measure the number of changes needed to coincide with the reference, their value is lower the closer the translation is to the reference. In the following chapters, we may append $\downarrow$ to the names of these metrics as a reminder of the fact that lower values are better. This is in contrast to all the remaining metrics that we describe below, whose values are higher the closer the translation is to the reference. For them, we will use $\uparrow$ as a reminder.

- Two metrics based on lexical precision.[9] First, the *bilingual evaluation understudy*, BLEU (Papineni et al., 2002), measures the amount of n-grams of varying lengths in the generated translation that appear in the reference. We use the variant that considers up to 4-grams and is smoothed as described by Lin and Och (2004a). And second, the *NIST* metric (Doddington, 2002) is an evolution of BLEU which puts more weight on the infrequent n-grams, especially the short ones. In this case, we score up to 5-grams.

- We also use three metric families based on the F-measure.[10] First, the *general text matcher*, GTM (Melamed et al., 2003; Turian et al., 2003), has several variants depending on the value given to the e parameter appearing in the exponent and the root of its formulation: GTM_1 counts the matching unigrams between the translation and the reference, whereas GTM_2 and GTM_3 reward longer matchings. Second, from the *ROUGE* metric (Lin and Och, 2004b) we use several of its variants, differing on how the translation and reference are matched: the ROUGE_{L} looks for the longest common subsequence (allowing gaps) between them, whereas ROUGE_{W} additionally penalizes gaps; on the other hand, $\text{ROUGE}_{\text{S}*}$ counts the matching bigrams (allowing gaps) between them, whereas $\text{ROUGE}_{\text{SU}*}$ additionally counts the matching unigrams. And third, the *METEOR* metric (Banerjee and Lavie, 2005; Denkowski and Lavie, 2010) counts the amount of words of the generated translation that match the reference, either *ex*actly, with *st*emming, with *sy*nonyms, or with *pa*raphrasing, adding penalties for reorderings.

- The *lexical overlap*, O_l, is based on the Jaccard coefficient (Jaccard, 1912) to quantify the similarity between sets. In particular, it is defined as the ratio of distinct lexical items common to both translation and reference to the total amount of distinct lexical items among translation and reference together. In other words, given the set of lexical items in the translation and the analogous

[9] Remember the standard definitions of $\mathit{precision}(T|R) = |T \cap R|/|T|$ and of $\mathit{recall}(T|R) = |T \cap R|/|R|$, where T stands for the set of obtained translation items and R for the set of reference translation items. Metrics based on either concept usually differ on how T and R are specifically defined and, especially, how the intersection $T \cap R$ must be handled.

[10] The F-measure is a combination of precision and recall through their harmonic mean (Rijsbergen, 1979).

set for the reference, O_l is computed as the division of the cardinal of their intersection by the cardinal of their union.

On syntactic similarity, we use metrics of two broad families:

- The metrics building on *shallow parsing* work on annotations for part-of-speech, word lemmas, and base phrase chunks, all obtained with automatic tools. We use six distinct metrics of this class. The *overlap on part-of-speech*, SP-$O_p(\star)$, computes the lexical overlap restricted to tokens belonging to a certain part-of-speech, and averages over all parts of speech. The *overlap on base phrase chunk*, SP-$O_c(\star)$, is analogous, but working on base phrase chunks instead of parts-of-speech. Finally, four metrics that use the NIST measure for computing the accumulated scores over sequences of up to 5 elements, which may be lemmas (*SP-NIST*$_l$), parts of speech (*SP-NIST*$_p$), base phrase chunks (*SP-NIST*$_c$), or chunk type and inside/outside/beginning-position labels (*SP-NIST*$_{iob}$).

- Three metrics working on the trees obtained through automatic *constituent parsing* on the translation and reference. The *overlap according to part-of-speech*, CP-$O_p(\star)$, is similar to SP-$O_p(\star)$, i.e., it computes the lexical overlap according to the part-of-speech, averaging over all parts of speech. The *overlap according to phrase constituent type*, CP-$O_c(\star)$, is similar to SP-$O_c(\star)$ but working on phrase constituents instead of base phrase chunks, which in particular allows to consider phrase embedding and overlap. The *syntactic tree matching*, CP-STM (Liu and Gildea, 2005), computes the ratio of matching subtrees of a certain height, averaging the results for heights from 1 to 9.

Finally, we also use the *uniformly-averaged linear combination*, ULC (Giménez and Màrquez, 2008), which combines the values of other metrics to give a general quality ranking. Thus, its precise definition depends on the list of metrics selected to be averaged. Furthermore, the reported ULC value for a system is relative to the other systems appearing in the same ranking. Hence, the exact same system may obtain different ULC values depending on which other systems it is ranked against. Note that, in particular, this implies that ULC values are not directly comparable across evaluations on different sets of systems.

Chapter 3

Towards Document-Level Machine Translation

In this chapter, we analyze some of the problems of translating texts sentence by sentence ignoring inter-sentence context information. In particular, we focus on frequent errors made by SMT systems that can be handled by exploiting context information across sentences. Then, we present a simple approach that uses context information to improve the coherence and cohesion levels of translations. This approach takes the form of a post-process strategy, which takes the output of an SMT system, detects words affected by some of the identified document-level issues, and suggests a better re-translation.

3.1 Document-Level Phenomena

We analyze SMT system outputs in order to identify and categorize those translation errors that can be related to document-level phenomena, caused by incorrectly managing inter-sentence and document-level information.

For such purpose, we selected a set of newswire articles. Journalistic texts exemplify the translation setting we are interested in, as the discourse in such formal texts holds a high level of coherence, cohesion, and lexical consistency since their objective is to describe a series of facts about a certain topic. In particular, we chose the newsCommentary corpus, which is a free corpus that is released every year for the WMT translation shared tasks.[1] We carried out our analysis on the test set from the 2011 release. It contains documents in English, Spanish, French, German, and Czech. However, only the English-Spanish parallel corpus was analyzed because our experiments focus on English to Spanish translations. This test set has 110 news items on different topics, with a total of 3,003 sentences. Each text is labeled with XML tags

[1] Available at: `http://www.statmt.org/wmt13/training-parallel-nc-v8.tgz`

that identify the news title and mark the limits of the document, paragraphs, and sentences, reflecting the document structure.

To generate automatic translations of the newswire corpus under analysis, we built a baseline SMT system based on the Moses (Koehn et al., 2003) decoder, using the Europarl-v7 English-Spanish parallel corpus. Then, in order to recognize patterns that could help identify errors in the generated translations, we used several NLP tools[2] to analyze the selected source document and, then, the gathered information was projected into the machine translations. Special attention was devoted to annotating the most typical features linked to document-level information, such as coreference or lexical consistency. Coreference directly correlates with the cohesion level of a document, while translations of named entities throughout a document are a direct indicator of its translation quality. Thus, coreference mistakes in the translations were studied to assess whether the cohesion from the source text had been transferred to the translation by the SMT system or not. Similarly, the relations of the same named entity in the source document were followed in the translation to identify and study those cases in which it varied.

As a result of the analysis, the following errors were identified as the most relevant translation phenomena related to the lack of document-level information management:

- **Inconsistent translation of "ambiguous" words:** One of the first detected errors were mistakes that break the semantic coherence of a document. Taking the "one-sense-per-discourse" assumption, a translation is better if its words appear translated in a consistent way, that is, the same source word appears translated into the same target word or, at least, into words with similar meanings within a document. In the analyzed translation examples, there were words that appeared translated into different and semantically incompatible forms within a document. This kind of error noticeably hinders the final translation quality.

  So, we want to identify these words and design a strategy to correct their incoherent translations. Choosing the right translation for these ambiguous words is equivalent to disambiguate the word in its context, but, in contrast to a word sense disambiguation problem, we are facing here instead a lexical choice problem, since our aim is to correct the bad translation choices made by the decoder.

  Consider, for example, the English word "desk". It can be translated into Spanish as "ventanilla", "escritorio", "mesa", or "mostrador", depending on the context where it is used. These translation options cover two basic meanings: "desk" as a piece of furniture ("mesa" and "escritorio") and "desk" as a counter ("ventanilla" or "mostrador"). The aim of our work is to make the decoder able to translate "desk" homogeneously throughout the document regarding its context.

[2] In particular, we employed the tools integrated in the Asiya toolkit (`http://asiya.cs.upc.edu`), the PoS tagger provided by the FreeLing library (`http://nlp.cs.upc.edu/freeling/`) by Padró et al. (2010), the RelaxCor (`http://nlp.cs.upc.edu/relaxcor/`) coreference resolutor of Sapena et al. (2010), and the BIOS (`http://www.surdeanu.info/mihai/bios/`) named entity recognizer of Surdeanu et al. (2005).

A general strategy to follow in order to handle this phenomenon is to identify those words in the source document that appear translated into different forms, detect wrong translations, and correct them by disambiguating these words in their contexts thus improving the translation consistency and coherence.

The already mentioned example when translating the word "desk" illustrates the situation. Another example of this phenomenon is the word "nickname"; in the context of a news item about the names used among colleagues, this word can be translated as "denominación", "sobrenombre", or "apodo". Since "sobrenombre" and "apodo" are synonyms given a context, the option of using "denominación" is not as accurate as the others. Then, we want to filter out this option to not use this term in the final translation to improve its lexical cohesion.

- **Incoherent translation of coreference mentions:** A word corefers with another in a text if both of them refer to the same entity. It is easy to find this kind of relation between names and pronouns through a text. If it is a well-formed text, these words must agree in gender and number, and these relations confer cohesion to the document. Moving to the side of a document translation, an indicator of its level of coherence is how this intra- and inter-sentential agreement among the corefered words is preserved. Studying the appearance of correlations in translations and how the agreement is maintained by the MT systems gave us hints about handling errors related to this document-level phenomenon. In particular, we would like to keep the translations coherence through coreference chains at translation time or fix the disagreements by a post-process strategy.

  An example of this phenomenon is the translation of the term "the engineer". In a document where the term refers to a woman called Ana, in Spanish it would become "la ingeniera" and not "el ingeniero". Another example, in the context of a news item talking about several councilors in the city hall of Prague, we find the sentence "... the Councilor for the environment, Lukas Plachy. He also guesses the right meaning." translated as "... *la consejera* de medioambiente, Lukas Plachy. *Él* también adivinó el significado correcto." This example shows a disagreement among the noun phrase "la consejera" and the "Él" pronoun. If the system is able to identify *Lukas* as a male name, a good translation would be "... *el consejero* de medioambiente, Lukas Plachy. *Él* también adivinó el significado correcto." where gender agreement is maintained for the mentions that corefer.

  The cohesion level of a translation is directly related to how the source text coreference information is projected into the produced translation. Monitoring the gender and number agreement of the coreference chains gives a good idea of how to identify translation errors that can be handled by using the document context information.

- **Disagreement in gender, number, and person:** Going further analyzing the agreement among words within a document, we realized that SMT systems can lose agreement among words in a noun phrase, among the persons of a subject and a verb, or even the verb tense being used in a part of a document.

A well written text usually does not present gender nor number disagreements and coherent verbal tenses are used to present different and distinguishable parts of itself.

We propose to not only design strategies to identify and fix incoherences in gender or number inside coreference chains, but also to take advantage of the described scenario and correct agreement errors in the intra-sentential scope or in the close inter-sentences context which is a more global and simpler problem than the one we presented related to the coreference information.

We use this categorization to design several strategies to correct translation mistakes by using document-level information. Through the rest of this chapter, we propose and analyze approaches to improve translations without modifying the inner decoder functionality.

3.2 Post-Process Strategies

Our approach can be broken down into three broad steps: (*i*) obtaining an initial, preliminary translation of the document, (*ii*) locating certain kinds of mistakes in the translation and identifying possible new translation options for them, and (*iii*) re-translating taking into account the new translation options identified in the previous step.

Step (*i*) is standard: we use a Moses system to obtain the translation, sentence by sentence. Step (*ii*) depends on the goal, i.e., on the kind of phenomenon that is tackled. In particular, we focus on improving the lexical coherence for those source words that appear translated into more than one different form and, additionally, on fixing gender or number incoherences using coreference information. Each of these goals is tackled by a specific approach; the former is described in Section 3.2.1 and the latter in Section 3.2.2. Note that the work performed in this step can take advantage of already having a translation of the whole document. Finally, step (*iii*) is a re-translation with Moses performed in either a *restrictive* or a *probabilistic* way. The restrictive re-translation forces[3] as the only possible translation the option provided by step (*ii*). On the other hand, the probabilistic way suggests[4] several possible translations, the most suitable ones identified in step (*ii*), and lets the decoder choose among them and the phrase table options. Notice that the latter approach introduces more noise because the system is managing more possible translations than in the restrictive one, sometimes as many as in the initial translation.

In Section 3.2.3 we evaluate the post-processing of lexical inconsistencies, of disagreements, and their combination.

[3]Forcing a translation is a feature of the Moses decoder that involves an XML markup of the source sentence with the information of available translation options. For the restrictive approach, we use the *exclusive* option of the decoder, which forces Moses to only use the translation option provided in the XML tag when translating the tagged source word.

[4]Suggesting a translation is another feature of Moses handled through XML markup. In this case, for the probabilistic approach we use the *inclusive* option of the decoder, which makes Moses take into account the translation options provided in the XML tag when translating the tagged source word. Each of the provided options has an associated probability and, if the aggregated probability of all of them is less than 1, then the decoder can also use the phrase table in the remaining cases.

3.2.1 Lexical Consistency

The strategy that deals with lexical consistency works on source words that appear translated into more than one different form when translating a document. We focus on tackling this phenomenon for nouns, adjectives, and main verbs in English, filtering out determiners, prepositions, auxiliary verbs, and others, that is, we focus on content words. In particular, we use the part-of-speech tags obtained with FreeLing (Padró et al., 2010) to do this filtering. For each of the resulting relevant source words, we use the alignments provided by the Moses' preliminary translation to link them to their corresponding target words. Next, for each of the distinct source words in consideration, we count the number of different translations it has throughout the document. This counting of target words is done at the lemma level, i.e., we conflate different translations when their lemmas coincide. For instance, the Spanish target words "amigo" and its feminine "amiga" would be considered as the same translation form for "friend" since they share the same lemma. On the other hand, "compañero" would be considered a different translation than "amigo" since they do not share the lemma, even if they can be synonyms in certain contexts. Lemmas are also obtained with FreeLing.

At this point, the process has enough information to identify the words that have multiple different translations in the document. Finally, we try to identify the most suitable translations for them. We do that in two different ways, depending on whether the re-translation step follows the restrictive or the probabilistic approach. For the former, we only have to obtain one single translation option in each case: we provide as the translation for each ambiguous word the option with most occurrences in the current document, unless there is a tie, in which case we do not suggest anything to avoid biasing the result in a wrong direction. Note that, although the most frequent translation option needs not be the correct one, we expect the intra-sentential context to be enough for the decoder to pick the proper translation in the majority of cases. Another option would be to pick a random translation as the good one. Doing this we control the introduced noise but we also lose information given by the decoder in the available preliminary translation. For the probabilistic approach, we proceed slightly differently: as before, for each ambiguous word we provide as a translation the option with most occurrences in the current document, but, in the case of a tie, in this occasion, we suggest all the tying options (all with the same associated probability). In this case, we are giving Moses freedom to choose guided by its language information, but we are also introducing noise because we are managing more possible translations than in the previous situation.

Looking at a concrete example, if the English source word "desk" appears four times in the text and it is translated two times as "mesa", one as "mostrador", and another one as "ventanilla", then the four occurrences of "desk" in the source text will be marked for re-translation. In this situation, the word "mesa" will be the single option given in the restrictive and probabilistic approaches, as it is the most frequent translation. However, if "desk" had another occurrence translated as "mostrador", then the restrictive approach would not suggest any re-translation due to the tie between the two "mesa" and the two "mostrador", whereas the probabilistic approach would give both "mesa" and "mostrador" as re-translation options, with 1/2 proba-

bility each.

3.2.2 Coreference and Agreement

It is easy to find words that corefer in a text. A word corefers with another if both refer to the same entity. These words must in principle agree in gender and number since they are representing the same concept (person, object, etc.). For instance, if "the engineer" appears referring to a woman, the correct translation in Spanish would be "la ingeniera" and not "el ingeniero".

An easy manner to measure the coherence of a translation is to look at the projections of the source coreference chains at the target language text and study the quality of its translations, for instance, analyzing the gender and number agreement. We implemented a post-process system that gets the coreference analysis of the source text and projects its coreference chains to the translation using the alignments generated by the decoder at translation time. Before starting to design some heuristics to fix the disagreements inside the words of a coreference chain, we look at the results of applying our strategy to the texts in the newsCommentary corpus of 2011, where we can only identify 2 examples where to apply our techniques. Then we moved to study some Wikipedia articles but we did not succeed either. After our study, we conclude that we do not have seen enough fixable examples in our texts to start developing any algorithm or heuristic only in that direction.

However, we find interesting to follow the idea of fixing incoherences in gender or number but in an intra-sentential scope. This can be seen as a simpler problem because it is not affected by possible errors given by the coreference resolutor, but as a wider problem since it can be located within any sentence of the document. Inside a sentence, since dependencies among words are shorter, the expressions tend to be translated correctly by standard SMT engines. However, there are larger distance relationships that are not handled properly by the MT systems, for instance, the agreement among subject and verb is a larger distance dependency that sometimes is lost in translation.

We designed a post-process strategy to fix gender and number disagreements. In order to simplify the problem and to filter out possible noisy situations, we focus on agreement among words in the same noun phrase (nouns, determiners, and adjectives). Furthermore, in a second step, we go beyond the study of these types of content words and also handle the agreement case for subject and verb in a sentence. As a first step, the post-process analyzes a source document and annotates it with PoS, coreference chains, and dependency trees using FreeLing. At this point, for the process, the main data structure is the parse tree since it is the linguistic structure that allows it to link the elements that need to agree to maintain the text coherence. In particular, a tree traversal is performed in order to detect nouns in the source. When a noun is found and its children are determiners and/or adjectives, the matching subtree is projected into the target via the word alignments. Then, from the target side, the process checks the agreement among tokens by using the PoS tags. If there is a disagreement, the correct tag for the adjective or determiner is built using FreeLing, which allows getting the correct form in the target language for the translation. We assume that the nouns are mostly translated correctly since they are the part of

the noun phrase with a higher semantic load. So, our post-process makes adjectives or determiners agree with a given noun. Finally, the system implements a similar strategy to check the agreement among the subject and the verb of a sentence. A tree traversal allows detecting the node that represents the verb of a sentence and the child corresponding to the subject. The structure is projected into the target via the alignments and the agreement is verified using the PoS information. If the subject is a noun, we assume that the verb must be conjugated in the third person plural or singular depending on the number of the noun; if it is a pronoun, then gender, person, and number must agree. As before, if there is a disagreement, the correct form is generated using FreeLing.

In both cases (i.e., determiner–adjective(s)–noun(s) and subject–verb disagreements) the output of the process is a proposed new translation that agrees in gender and number. As in Section 3.2.1, the actual output depends on whether the re-translation step follows the restrictive or the probabilistic approach. For the former, we just provide the new translation. For the latter, we also associate to this new translation a probability less than 1[5] that allows the decoder to also take into account the remaining translation options of the phrase table.

3.2.3 Experiments

Settings

Our baseline English-to-Spanish translation system is a Moses decoder (Koehn et al., 2007) trained on the Europarl corpus (Koehn, 2005) in its version 7, and using GIZA++ (Och and Ney, 2003) to obtain the word alignments. It uses a 5-gram Spanish language model obtained by using SRILM (Stolcke, 2002) with interpolated Kneser-Ney discounting on the target side of the Europarl-v7 corpus. The feature weight optimization is done with MERT (Och, 2003) against the BLEU metric (Papineni et al., 2002) on the newsCommentary2009 development corpus. We use newsCommentary2011 as test set.

We carry out an automatic and manual evaluation of our post-processes. For the automatic evaluation we use the Asiya toolkit (Giménez and Màrquez, 2010; Gonzàlez et al., 2012) with several lexical metrics (TER, BLEU, NIST, $\text{METEOR}_{\text{ex}}$, and ROUGE_{L}), a syntactic metric based on the overlap of part-of-speech elements ($\text{SP-O}_{\text{p}}(\star)$), and an average of a set of 27 lexical and syntactic metrics[6] (ULC). Nevertheless, these measures are not informative enough considering that we only perform small modifications on the preliminary baseline translations. For this reason, we confer more relevance to the manual evaluation of the outputs.

[5] In practice, we set this probability to 0.8 to allow the decoder to consider the rest of translation options in the phrase table but controlling the introduced noise.

[6] The full list of metrics averaged in ULC is: WER, PER, TER, TER_{base}, BLEU, NIST, GTM_1, GTM_2, GTM_3, $\text{METEOR}_{\text{ex}}$, $\text{METEOR}_{\text{st}}$, $\text{METEOR}_{\text{sy}}$, $\text{METEOR}_{\text{pa}}$, ROUGE_{L}, ROUGE_{W}, $\text{ROUGE}_{\text{S}*}$, $\text{ROUGE}_{\text{SU}*}$, O_{l}, $\text{SP-O}_{\text{p}}(\star)$, $\text{SP-O}_{\text{c}}(\star)$, $\text{SP-NIST}_{\text{l}}$, $\text{SP-NIST}_{\text{p}}$, $\text{SP-NIST}_{\text{c}}$, $\text{SP-NIST}_{\text{iob}}$, $\text{CP-O}_{\text{p}}(\star)$, $\text{CP-O}_{\text{c}}(\star)$, CP-STM.

System			TER↓	BLEU↑	NIST↑	METEOR$_{ex}$↑	ROUGE$_L$↑	SP-O$_p$(⋆)↑	ULC↑
baseline			55.45	26.73	7.34	27.78	29.36	31.53	85.01
lexical	restrictive		55.39	26.76	7.34	27.80	29.39	31.60	83.26
	probabilistic		55.41	26.73	7.34	27.77	29.38	31.58	85.07
agreement	restrictive		55.46	26.66	7.33	27.75	29.41	31.69	85.10
	probabilistic		55.45	26.73	7.33	27.75	29.41	31.64	85.05
lex.	rest.	+ agr. rest.	55.46	26.65	7.32	27.74	29.40	31.68	85.08
	rest.	+ agr. prob.	55.45	26.73	7.33	27.75	29.40	31.63	85.05
	prob.	+ agr. rest.	55.48	26.64	7.32	27.74	29.38	31.67	79.28
	prob.	+ agr. prob.	55.46	26.72	7.32	27.74	29.40	31.63	85.04

Table 3.1: Automatic evaluation of the systems, compared to the MOSES baseline system. The rows for agreement check the agreement among nouns, determiners, and adjectives as well as the agreement among subject and verbs.

Lexical Consistency

Global automatic evaluations of the whole test set are shown in Table 3.1 (*lexical* rows), and scores for some individual documents in Table 3.2. Global results present a very small variability with respect to the baseline. These small variations are expected: overall, the full test set has 74,753 words in total, and we only introduce 476 changes in the restrictive strategy and 1,064 in the probabilistic. Moreover, recall that standard evaluation metrics, in general, are not designed to capture phenomena at document level. The same happens when evaluating individual documents, as shown in Table 3.2. There we can see score improvements in some of the documents but not in others; for instance, BLEU for the 1st document improves a 0.2% on the probabilistic approach with respect to the baseline, but it remains the same for the 2nd document. The scores do not show any systematic preference for a system and it is necessary a manual evaluation of the outputs.

Table 3.3 presents the results of manually evaluating the output of the post-processes for the five documents with the most changes proposed by the re-translation. Recall that all the words tagged for re-translation in the restrictive approach are also tagged in the probabilistic approach, since the latter handles all the cases of the restrictive and, additionally, also introduces tags in the tying situations. For this reason, as shown in the table, the number of tags for the restrictive approach is always a lower bound for the probabilistic. Clearly, the same happens for the number of different words involved in the tags and the number of tagged lines, which are also shown in the table. In order to see the scope of the introduced changes, the table also presents the total number of changed lines, the total number of actual changed words, and how many of those are correct. For reference, the respective BLEU scores are also listed in Table 3.3. As expected, these scores are very close to the baseline due to the small number of changes introduced by the post-process. For instance, the 20th document is the one with the most changes and, yet, only 9 words are modified with the restrictive approach and 11 with the probabilistic one. In this specific document, the accuracy of the changes in both approaches is above 50%, and it is higher in the remaining four documents of Table 3.3, with the restrictive approach reaching 100% in the four of them. For the whole test set, the probabilistic approach obtains

news item	System	TER↓	BLEU↑	NIST↑	METEOR$_{ex}$↑	ROUGE$_L$↑
1	baseline	68.87	10.79	3.8257	19.66	42.75
	restrictive	68.87	10.80	3.8318	19.64	42.52
	probabilistic	69.09	10.98	3.8173	19.53	42.44
2	baseline	63.69	20.73	4.2684	24.57	47.21
	restrictive	63.69	20.73	4.2684	24.57	47.21
	probabilistic	63.69	20.73	4.2684	24.57	47.21
3	baseline	66.79	14.15	4.1891	21.53	42.34
	restrictive	66.72	14.16	4.1929	21.54	42.39
	probabilistic	65.97	14.23	4.2232	21.69	42.58
4	baseline	67.55	18.69	4.0303	21.81	43.65
	restrictive	67.40	18.93	4.0638	22.07	44.11
	probabilistic	67.11	18.94	4.0731	22.14	44.31
5	baseline	69.15	13.74	3.7287	20.06	39.99
	restrictive	69.15	13.74	3.7287	20.06	39.99
	probabilistic	69.06	13.75	3.7319	20.04	40.09

Table 3.2: Automatic evaluation of the lexical consistency experiment on 5 individual news items, using either the restrictive or the probabilistic approaches, and compared to the MOSES baseline system.

news item	System	BLEU↑	tags	words	OK/ch	lineTags	lineDif
20	baseline	13.40					
	restrictive	13.56	26	8	5/9	13	6
	probabilistic	13.22	45	15	7/11	19	8
25	baseline	14.42					
	restrictive	14.45	18	4	4/4	16	3
	probabilistic	14.52	38	10	5/5	28	7
39	baseline	28.49					
	restrictive	28.20	16	5	5/5	15	4
	probabilistic	28.56	34	11	6/8	25	7
48	baseline	30.05					
	restrictive	30.06	42	3	3/3	23	10
	probabilistic	29.83	53	7	4/5	24	15
49	baseline	25.54					
	restrictive	25.87	24	5	5/5	17	8
	probabilistic	25.83	42	12	7/8	23	10

Table 3.3: Manual evaluation of the systems for lexical coherence, using either the restrictive or the probabilistic approaches, for 5 individual news items of the test set. The *tags* column shows the number of introduced tags, *words* shows the number of different words involved in the tags, *OK/ch* shows the number of changes made with respect to the baseline translation and how many are correct (according to our criterion of having one-sense-per-discourse and the word appearing in the reference), *lineTags* shows the number of tagged lines in the source text, and *lineDif* shows the number of different lines between the final and the baseline translations.

accuracies around 80%, and the restrictive about 1% higher.

As an illustrative example of how our system works, we consider a particular document with a news item about a trial. It contains the phrase "the trial coverage" translated in first place as "la cobertura de prueba" where the baseline system is translating wrongly the word "trial" as "prueba" (meaning evidence or proof). Our post-process has access to the other occurrences of the word "trial" throughout the document, which is more frequently translated as "juicio". Thus, "trial" is identified as an ambiguous word and it gets tagged with the good translation form "juicio". Unfortunately, on some occasions, the changes are not as positive. For example, in another document the word "building" appears five times, being translated three times as "construcción" and two times as "edificio". For our system, the first option is better as long as it appears more times in the translation than the second one. So, it suggests the decoder to always use "construcción" when re-translating "building". Doing that, we produce two changes in the final translation that generate two errors with respect to the reference translation, although both translation options are synonyms. In this case, our system moves the translation away from the reference although both translations should be correct.

Regarding the errors introduced by the systems, we find that they are caused mainly by bad alignments (which lead to an erroneous projection of the annotated structures on the source), errors in the part-of-speech tagging, the presence of untranslated words, or are a consequence of the fact that sometimes the most frequent translation for a given word in the initial state is wrong.

In general, we observed that the re-translation step performs very local changes, affecting mostly the tagged words without modifying their immediate context nor the general sentence structure. However, these few local changes are noticeable to a final user given the positive feedback from the manual evaluation.

Coreference and Agreement

Global automatic evaluations of the whole test set are shown in Table 3.1 (*agreement* rows), where, as in the previous experiment, we cannot observe significant improvements in the usual metrics. Table 3.4 presents scores for some individual documents, in this case tackling only agreement between nouns and their determiners and adjectives, but not between subject and verb. In this occasion, document by document we can see encouraging results, e.g., we gain 0.2% in the BLEU score for the 5th document by just introducing really simple changes.

Table 3.5 presents the manual evaluation of the post-processes for the five documents with the most changes proposed by the re-translation We observe that these changes have an impact on the BLEU score of the final translation because, in this case, the number of changes is higher. For instance, in the 22nd document, there is a drop of almost 2 points in BLEU after applying the post-process although many of the changes made after the re-translation are correct. We observe the same behaviour in the 27th document, although the rest of the news items show an opposite trend. According to the manual evaluation, the restrictive system is better than the probabilistic one and reaches accuracies above 80% in the analyzed documents.

A positive example of the performance of the system is the re-translation of the

news item	System	TER↓	BLEU↑	NIST↑	METEOR$_{ex}$↑	ROUGE$_L$↑
1	baseline	68.87	10.79	3.8257	19.66	42.75
	nn+dets	68.21	11.87	3.8778	19.97	43.02
	nn+dets+adj	67.99	11.88	3.8900	20.02	43.10
2	baseline	63.69	20.73	4.2684	24.57	47.21
	nn+dets	63.50	20.79	4.2859	24.69	47.59
	nn+dets+adj	63.50	20.72	4.2597	24.49	47.52
3	baseline	66.79	14.15	4.1891	21.53	42.34
	nn+dets	66.87	14.16	4.1833	21.50	42.25
	nn+dets+adj	66.79	14.16	4.1791	21.48	42.36
4	baseline	67.55	18.69	4.0303	21.81	43.65
	nn+dets	67.40	18.73	4.0474	21.89	43.77
	nn+dets+adj	67.40	18.83	4.0510	21.91	44.02
5	baseline	69.15	13.74	3.7287	20.06	39.99
	nn+dets	68.79	13.85	3.7687	20.32	40.29
	nn+dets+adj	68.61	13.92	3.7956	20.43	40.14

Table 3.4: Automatic evaluation of the agreement experiment on 5 individual news items of the test set, comparing the MOSES baseline system to the restrictive post-process correcting just gender and number disagreements between nouns and their determiners or between nouns, their determiners, and their adjectives.

news item	System	BLEU↑	OK/ch	dets	adjs	verbs
5	baseline	13.74				
	restrictive	14.06	23/26	17/19	6/7	0/0
	probabilistic	13.79	15/26	12/19	3/7	0/0
6	baseline	11.06				
	restrictive	11.22	19/23	8/11	11/11	0/1
	probabilistic	11.10	10/23	4/11	6/11	0/1
22	baseline	16.23				
	restrictive	14.74	17/25	4/8	13/17	0/0
	probabilistic	14.89	10/25	2/8	8/17	0/0
27	baseline	13.15				
	restrictive	12.35	22/28	14/19	7/8	1/1
	probabilistic	12.76	21/28	14/19	7/8	0/1
33	baseline	15.09				
	restrictive	16.05	18/22	14/16	3/3	1/3
	probabilistic	15.97	11/22	7/16	2/3	2/3

Table 3.5: Manual evaluation of the systems for agreement, using either the restrictive or the probabilistic approaches, for 5 individual news items of the test set. The *OK/ch* column shows the number of changes made with respect to the baseline translation and how many are correct, and these amounts are broken down into three categories: *dets* show the same information but only for changes done over determiners, *adjs* over adjectives, and *verbs* over verb forms.

source phrase “the amicable meetings”. This phrase is translated by the baseline as “el amistosa reuniones”, where one can find disagreements of gender and number among the determiner, the adjective, and the noun. The system detects these disagreements and after tagging the source with the correct forms and re-translating, one obtains the correct final translation “las reuniones amistosas”, where we observe also that the decoder has reordered the sentence.

Regarding the errors introduced by the system, we observe again that many of them are caused by wrong analysis. For instance, in the sentence “all (the) war cries” which should be translated as “todos los gritos de guerra”, the dependence tree shows that the determiner depends on the noun “war” and not on “cries”, so, according to this relation, our method identifies that the determiner and the translation do not agree and produces the wrong translation “todos (la) guerra gritos”.

These results also show that for our approach it is easier to detect and fix disagreements among determiners or adjectives and nouns than among subjects and their related verbs. In general, this is because our current system does not take into account subordinated sentences, agent subjects, or other complex grammatical structures, and therefore the number of detected cases is smaller than for the determiner–adjective–noun cases.

Chaining both Post-Processes

In order to complete this set of experiments, we run sequentially both systems. Global automatic evaluations of the whole test set are shown in Table 3.1 (*lex.+agr.* rows), where again it is only possible to observe a very small variability with respect to the baseline. Table 3.6 shows the results of a manual evaluation for 5 documents with the most suggested changes, following the same format as in the previous experiment. Once again, we observe small variations in BLEU scores, but we see that when the systems introduce changes, they are able to fix more translations than the ones they damage. Also as before, it is easier to detect and fix disagreements among determiners, adjectives, and nouns than those regarding verbs because it is more difficult to detect disagreements with verbs and also these kinds of errors are less frequent than gender-number disagreement among elements in a noun phrase.

3.3 Conclusions

Most of the current MT systems are designed to translate documents sentence by sentence ignoring the contextual information, generating then translation outputs with different kinds of errors that hinder the coherence and cohesion levels.

Section 3.1 analyzes and presents a categorization of the most noticeable translation errors related to document-level information in the sense that they may be easy to fix using context information. In particular, lexical consistency and gender and number agreement at intra- and inter-sentence scope are described.

Once we have identified and described the phenomena we are interested to handle, in Section 3.2 we explored a methodology to include document-level information within a translation system without changing the decoding mechanism. The method

news item	System	BLEU↑	OK/ch	dets	adjs	verbs
20	baseline	13.40				
	restrictive + restrictive	13.38	17/19	14/15	3/3	0/1
	restrictive + probabilistic	13.44	14/19	11/15	2/3	1/1
	probabilistic + restrictive	13.21	16/17	13/14	3/3	0/0
	probabilistic + probabilistic	13.44	12/17	10/14	2/3	0/0
25	baseline	14.42				
	restrictive + restrictive	14.68	12/19	9/13	3/6	0/0
	restrictive + probabilistic	15.09	15/19	10/13	5/6	0/0
	probabilistic + restrictive	14.39	10/17	6/11	4/6	0/0
	probabilistic + probabilistic	14.82	13/17	8/11	5/6	0/0
39	baseline	28.49				
	restrictive + restrictive	30.02	20/22	14/16	6/6	0/0
	restrictive + probabilistic	29.59	18/22	13/16	5/6	0/0
	probabilistic + restrictive	29.94	19/21	14/16	5/5	0/0
	probabilistic + probabilistic	29.59	17/21	13/16	4/5	0/0
48	baseline	30.05				
	restrictive + restrictive	29.57	6/6	5/5	1/1	0/0
	restrictive + probabilistic	29.60	4/6	4/5	0/1	0/0
	probabilistic + restrictive	29.57	6/6	5/5	1/1	0/0
	probabilistic + probabilistic	29.60	4/6	4/5	0/1	0/0
49	baseline	25.54				
	restrictive + restrictive	25.82	9/11	3/4	6/7	0/0
	restrictive + probabilistic	26.02	9/11	3/4	6/7	0/0
	probabilistic + restrictive	25.63	8/11	3/4	5/6	0/1
	probabilistic + probabilistic	26.02	9/11	3/4	5/6	1/1

Table 3.6: Manual evaluation of chaining both post-processes (first applying the disambiguation post-process and, afterwards, checking for the agreement), using either the restrictive or the probabilistic approaches, and compared to the Moses baseline system. The *OK/ch* column shows the number of changes made with respect to the baseline translation and how many are correct, and these amounts are broken down into three categories: *dets* show the same information but only for changes done over determiners, *adjs* over adjectives, and *verbs* over verb forms.

performs a two-pass translation. First, a translation is generated by means of a baseline SMT system. Afterwards, incorrect translations according to predefined criteria are detected and new translations are suggested. The re-translation step uses this information to promote the correct translations in the final output.

A common post-process is applied to deal with lexical coherence at document level and intra- and inter-sentence agreement. The source documents are annotated with linguistic processors and the interesting structures are projected on the translation where inconsistencies can be uncovered. In order to handle lexical coherence, we developed a post-process that identifies words translated with different meanings through the same document, described in Section 3.2.1. For treating disagreements, we developed a post-process that looks for inconsistencies in gender, number, and person

within the structures determiner–adjective(s)–noun(s) and subject–verb, described in Section 3.2.2.

Because we are treating sparse phenomena, an automatic evaluation of our systems does not give us enough information to assess the performance of the systems. A detailed manual evaluation of both systems shows that we only introduce local changes. The lexical-coherence-oriented post-process induces mostly correct translation changes when using our restrictive system, improving the final coherence of the translation. Furthermore, for the post-process focused on the analysis of the number and gender agreement, it achieves more than 80% of accuracy over the introduced changes in the manually-evaluated news documents. We also observed that some of the negative changes are consequence of bad word alignments which introduce noise when proposing new translations.

Chapter 4

Word Embeddings in Machine Translation

Recently, distributed representation models have been used successfully in many natural language processing tasks. These models have proved to be robust and powerful to predict semantic relations between words. However, they are unable to handle lexical ambiguity as they conflate word senses of polysemous words into one common representation[1]. This limitation is already discussed by Wolf et al. (2014), who also propose an extension of the learning architecture that jointly trains two monolingual models for a language pair. In this way, they are able to capture meanings across the two languages. This is a refinement of the observation of Mikolov et al. (2013b), who noticed that even when training the two monolingual models independently they can be related through a linear mapping: this allows the authors to use the language of the pair with the best linguistic resources to improve dictionaries or phrase tables for the language with the poorer resources. We also exploit language pairs, but in contrast to these approaches, our goal is to use word alignments between the two corpora to create bilingual tokens and, in this way, disambiguate the lexicon and obtain a bilingual vector model with more precise semantics.

We use the WORD2VEC package presented by Mikolov et al. (2013a) to build monolingual and bilingual word embeddings and then evaluate the acquired representations on three different tasks: (*i*) predicting semantically related words, (*ii*) performing a cross-lingual lexical substitution task, and (*iii*) guiding the decoding in a translation task. The latter evaluation is the most relevant for our ultimate ends, as it assesses the appropriateness of the models for the purpose of translation.

[1]More recent embedding approaches like BERT Devlin et al. (2019) are able to produce contextualized word embeddings, although they still consider only sentence level context.

4.1 Semantic Models Using word2vec

We use the implementation of the Continuous Bag-of-Words (CBOW) algorithm in the WORD2VEC package to build our models. This algorithm uses a neural network to predict a word given a set of its surrounding words, where the order of the words in the history does not influence the projection. This is in contrast to the other algorithm available in WORD2VEC for building the models, Skipgram, which trains a neural network to predict the context of a given word. We choose to use CBOW because we want to apply these models in the translation task and the CBOW training mechanism is closer to that, i.e., predict an adequate (translated) word taking into account a set of (translated) context words.

In order to enrich the semantic information encoded in the models, we transform the training data to hold information from both the source and target languages together. In particular, we use an aligned parallel corpus to extract a new training corpus of word pairs: $(w_{i,S}|w_{i,T})$. For instance, if the words *house* and *casa* are aligned in the parallel corpus, we consider the new form *house*|*casa*. In this way, we hope to better capture the semantic information that is implicitly contained by a text and its translation. For example, we expect to be able to distinguish among the different meanings of the word *desk* by considering its corresponding forms in Spanish, i.e., differentiate between *desk*|*mesa*, *desk*|*mostrador*, *desk*|*escritorio*, and any others.

The training has been performed under the following settings. We build the bilingual training data set from parallel corpora in the English-Spanish language pair available in OPUS[2] (Tiedemann, 2009, 2012). In particular, we select the EUROPARL-v7, UNITED NATIONS, MULTILINGUAL UNITED NATIONS, and SUBTITLES-2012 corpora, which total 584 million words for English and 759 millions for Spanish. These corpora have been automatically aligned to obtain the word alignment information necessary for our bilingual models. We choose the one-to-one alignments to avoid noise and repetitions in the final data. Monolingual models are built with the Spanish or English side of these same corpora. Finally, regarding the WORD2VEC training parameters, we consider several configurations for the dimensionality of the vectors and the context window size.

4.2 Accuracy of the Semantic Model

We first evaluate the quality of the models based on the task of predicting related words. A bilingual test set is manually built from the SEMANTIC-SYNTACTIC WORD RELATIONSHIP test set of Mikolov et al. (2013a) by attaching to each English word its Spanish translation, according to a native Spanish speaker. This test set contains 19,544 questions, divided into 8,869 semantic and 10,675 syntactic questions. For the monolingual models, we use the same test set projected to either the English or the Spanish side.

The evaluation task consists in predicting a word given a pair of related words and a question word. Intuitively, the task can be understood as solving analogies, such as:

[2] `http://opus.lingfil.uu.se/`

Model	Vector dimensions			Context window size		
	300	**600**	**1,000**	**2**	**5**	**10**
English	31.24%	33.53%	33.34%	31.97%	33.53%	32.68%
Spanish	11.42%	12.30%	12.16%	12.54%	12.30%	14.15%
English-Spanish	21.96%	23.68%	23.41%	19.50%	23.68%	25.54%

Table 4.1: Accuracy of the vector models filtered to the 30,000 most frequent vocabulary entries, when varying the vector dimensions (with the size of the context window fixed to 5) and when varying the context window size (with the dimension of the vectors fixed to 600). The 600 and 5 columns coincide by definition.

$$\textit{Athens} \text{ is to } \textit{Greece} \text{ as } \textit{Paris} \text{ is to } \textbf{?}$$

which in the word vector setting becomes the equation:

$$\textit{Paris} - \textit{Athens} + \textit{Greece} = \textbf{?}$$

with its expected solution being *France* or, rather, a vector that is closest to the word embedding of *France*. In that example, the words provided by the English test set are the pair *Athens Greece* and the question is *Paris*. In our English-Spanish bilingual scenario, the same example is represented by the pair *Athens|Atenas Greece|Grecia* and the question *Paris|París*, and in this case the task is to predict *France|Francia*.

The previous analogy is an example of a semantic question. The syntactic questions of the test set follow the same scheme, but relating words through syntactic transformations. For instance:

$$\textit{good} \text{ is to } \textit{better} \text{ as } \textit{bad} \text{ is to } \textbf{?}$$

with the expected solution being the comparative *worse*.

4.2.1 Results

First of all, we evaluate how the training parameters affect the quality of our models. Table 4.1 shows the effect of varying the vector dimensionality in terms of accuracy,[3] with the vocabulary of each model filtered to its 30,000 most frequent entries. We observe the benefit of using more than 300 dimensions, but there is no clear gain in using more than 600 dimensions in any of the models. Regarding the context window size, the same Table 4.1 shows that we obtain the best results for the English model when using a window of 5, whereas a window of 10 is optimal for the Spanish model. In the case of the bilingual model, we also observe an improvement of the accuracy when increasing the size of the context window, showing how a larger context helps the model disambiguate word senses.

[3]All the accuracy values are computed over the questions of the test set whose words are known to the model in consideration. The percentage of questions of the test set with known words for the English model is 64.67%, for the Spanish model 44.96%, and for the English-Spanish model 13.74%.

Model	Overall	Semantic	Syntactic
English	32.47%	19.17%	38.57%
Spanish	10.24%	15.15%	8.70%
English-Spanish	23.68%	25.63%	22.60%

Table 4.2: Accuracy of the vector models per question category.

Table 4.2 shows the accuracy results for our models of 600-dimensional vectors and trained with a context window size of 5, both overall for the whole test set and broken down into the subcategories of semantic and syntactic questions. Notice that, compared to Table 4.1, without filtering vocabularies the models achieve a slightly worse accuracy. This is because, although using the whole vocabulary improves the coverage, it also decreases the precision: each subspace is populated by more vectors (e.g., because more synonyms are present), making it harder for the correct word embedding to be the closest one. An analogous problem will manifest later in the results of the cross-lingual lexical substitution task in Section 4.3.2, where considering the 5 closest embeddings will prove to be more accurate than focusing exclusively on the single closest one.

We observed that the low accuracies achieved by the Spanish models are due to the noise in the training data, such as multiple spellings or synonyms for some words, e.g., "Kazakhstan" appearing as "Kazajistán", "Kazajstán", or "Kazakstán". Accuracy decrease may also be caused by the compromises made during the translation of the test set, e.g., "faster" becoming "rápido" instead of "más rápido" since we are not considering multi word expressions. In the bilingual case, the accuracy is lower than for English due to the coverage problem inherited from the Spanish data and the noise from word alignments.

4.3 Cross-Lingual Lexical Substitution Task

We now evaluate the semantic models through the effect they can have in a scenario resembling translation. In particular, we implement a cross-lingual lexical substitution task inspired by the one carried out in SemEval-2010.[4] To this end, first, we identify the content words of an English test set which are translated into Spanish in more than one different way by a Moses system. We call these words ambiguous. Then, the task consists in choosing the adequate Spanish translation for each of the ambiguous words. In our case, the correct choice is given by the reference translation of the test set.

To give an example, the word "desk" appears several times in a news item about a service to attend grievances against exaggerated rents. This word in such context has the meaning of *a service counter or table in a public building, such as a hotel.*[5] The correct translation to that meaning in Spanish would be the word "mostrador" or "ventanilla". But in the output of the SMT system, besides the correct translations,

[4] `http://semeval2.fbk.eu/semeval2.php?location=tasks#T24`

[5] Definition taken from Collins Concise English Dictionary.

Model		Top 1	Top 5
Spanish	CBOW	47.71%	65.44%
	Skipgram	47.71%	59.19%
English-Spanish	CBOW	62.39%	85.49%
	Skipgram	62.39%	78.36%

Table 4.3: Accuracy of the vector models in the cross-lingual lexical substitution task, when suggesting either the top 1 (i.e., only the best) or the top 5 (i.e., the five best) Spanish translation options.

we can also find "desk" translated as "mesa" or even as "escritorio" in the same document. Since the reference translation contains "ventanilla", only this word will be considered correct in the evaluation.

We tackle the problem of choosing the adequate translation for each occurrence of an ambiguous word as follows. Given one of such occurrences, we first compute a vector representing its context in the target side: we take the 2 previous and 2 following words in the target, look for their word embedding in our models, and combine them with a vector addition. Note that, clearly, when accessing a bilingual model it is necessary to retrieve the source word aligned to the target word in question in order to build the pairs $(w_S|w_T)$ needed for querying the model. Next, we score each translation option seen within the document. This score is the cosine similarity between the computed context vector and the word embedding of the translation option in consideration. Finally, we choose the translation option with highest score as the best option for that particular occurrence of the ambiguous word.

4.3.1 Settings

Our English-to-Spanish translation system is a Moses decoder (Koehn et al., 2007) trained on the Europarl-v7 corpus (Koehn, 2005) and using GIZA++ (Och and Ney, 2003) to obtain the word alignments. The Spanish language model is an interpolation of several 5-gram language models obtained using SRILM (Stolcke, 2002) with interpolated Kneser-Ney discounting on the target side of the Europarl-v7, United Nations, newsCommentary2007-2010, AFP, APW, and Xinhua corpora as given by QuEst (Specia et al., 2013).[6] The feature weight optimization is done with MERT (Och, 2003) against the BLEU metric (Papineni et al., 2002) on the newsCommentary2009 development corpus. We use newsCommentary2011 as test set.

4.3.2 Results

Table 4.3 shows the results of the evaluation of our bilingual model in comparison to our monolingual model trained in Spanish. In this case, we consider models trained with both the CBOW and the Skipgram algorithms of word2vec, using 600 dimen-

[6]Resources are available at: `https://www.quest.dcs.shef.ac.uk/wmt13_qe.html`

sions for the vectors and a context window size of 5. We use a stop-word list to skip some adverbs, common verbs, the prepositions and conjunctions as ambiguous words to avoid noise in the results. For our test set, 8.12% of the words are ambiguous and, in average, there are 3.26 translation options per ambiguous word. The monolingual model has, in average, a coverage of 90.97% per document and the bilingual one 87.53%, which over the ambiguous words become 87.37% for the monolingual and 83.97% for the bilingual.

The two WORD2VEC algorithms have the same performance for this task when they suggest only the best option: an accuracy of 47.71% for the monolingual model and 62.39% for the bilingual one. So, bilingual models are encoding considerably more semantic information than monolingual models in the Top 1 setting. Furthermore, the bilingual model is able to outperform a frequentist approach, since always using the most frequent translation option for this task leads to 59.76% of accuracy.

Accuracies are significantly improved when more options are taken into account. This is expected since, for instance, synonym translations are considered mistakes when they turn up in the Top 1 setting, leading to underestimating those accuracy results (Mikolov et al., 2013b). More precisely, when looking at the accuracy at Top 5, CBOW achieves 65.44% with the monolingual model and 85.49% with the bilingual one, whereas the Skipgram models have, approximately, 6 less points in the monolingual case and 7 in the bilingual one. These results indicate that CBOW bilingual models are capturing better the semantics and that considering more than one option can be important in the full translation task. Furthermore, the models improve over the 59.76% accuracy of the frequentist approach, except for the Skipgram monolingual model, which nevertheless achieves a similar accuracy.

4.4 Translation Task with Semantic Space Language Models

We now focus on the usage of word embeddings within the LEHRER document-oriented decoder. Intuitively, the decoder uses these models in analogy with the standard language models, but working on vectors representing words and their contexts instead of n-grams of words. Thus, when using monolingual word embeddings, the approach mimics a language model computed over semantic information from the target document, whereas in the case of the bilingual vector models, the approach mimics a language model over semantic information from both the target and the source sides. In any case, the expected effect is to promote translation choices that are semantically similar to the target context.

More precisely, we follow the use of word vector models as a Semantic Space Language Model (SSLM) by Hardmeier et al. (2012). In that work, the authors use latent semantic analysis[7] (Bellegarda, 2000; Foltz et al., 1998) to build their word vector models and proceed as follows to reward word choices that are semantically close to their context. For each word w in a document translation candidate, a score

[7]This technique uses the analysis of relationships among a set of documents and their terms to create a simplified matrix that represents the word counts per paragraph. This approach assumes that two close words occur in similar pieces of text.

is computed based on the cosine similarity between the vector representation of w and the sum of the vector representations of the n target words that precede w in the document translation. The similarity is then converted into a probability by a histogram lookup, as proposed by Bellegarda (2000). The non-content words and the words unknown to the model are handled specially, both when computing their associated score and when considering them as part of the context of any later word. Formally, the score for w is:

$$score(w|\vec{h}) = \begin{cases} p_{unigram}(w) & \text{if } w \text{ is a stop word} \\ \alpha \cdot p_{similarity}(cossim(\mu(w), \vec{h})) & \text{if } w \in \text{dom}(\mu) \text{ is not a stop word} \\ \varepsilon & \text{otherwise} \end{cases} \tag{4.1}$$

where $\vec{h}$ is the vector representing the preceding context of w (i.e., the sum of the vector representations of the n previous non-stop known words in the document translation), $p_{unigram}$ maps each stop word to its relative frequency in the training corpus, α is the proportion of content words in the training corpus, $p_{similarity}$ takes the cosine similarity computed by *cossim* and maps it from the range $[-1, 1]$ into a probability in $[0, 1]$ according to a given histogram, μ represents the word vector model mapping words to their associated vector representations, with $\text{dom}(\mu)$ being its domain, and ε is a small fixed probability. The final score for the document translation candidate is the sum of the natural logarithm of the score of each of its words. The logarithm is required to properly fit in the log-linear scoring model from Koehn et al. (2003) used by the decoder.

The value chosen by Hardmeier et al. (2012) for the parameter n is 30 to make it possible that the context used in the computations crosses sentence boundaries. Other parameters are offered by the SSLM implementation to alter its basic scheme described above. For instance, it is possible to ignore casing differences when accessing either the stop-word list or the vector model, to perform bilingual queries into the vector model by taking the source words aligned with the target word in consideration, or to use a transformation different from a histogram when mapping the cosine similarities to probabilities. For the latter, we use the mapping where $p_{similarity}(x)$ is x when $x > 0$ and 1 otherwise. In this way, the cosine similarity of words that are semantically distant from their context (in particular, when the vectors are $\pi/2$ radians or more apart) does not contribute to the final score, since its natural logarithm after such $p_{similarity}$ mapping is 0. This idea is similar to how the translation model of the decoder scores out-of-vocabulary words.

4.4.1 Settings

We use as SMT baseline for English-to-Spanish translation the Moses system described in Section 4.3.1. Regarding the document-level decoder, we use a Lehrer baseline system with the following configuration. First, the enabled features coincide with the ones of the Moses baseline system, except for lexical reordering, which is not implemented. Second, its feature weights are tuned with MERT (Och, 2003) against the BLEU metric (Papineni et al., 2002) on the newsCommentary2009 develop-

ment corpus.[8] This tuning is run on a Moses decoder with the exact same feature configuration as the Lehrer baseline. Third, the decoding is set to use the output of the Moses baseline system as initial translation. Finally, the remaining parameters are left to their default values recommended by Hardmeier (2014).

We consider three variants of the Lehrer system having the document-level SSLM as an additional feature function. The word vector models used for SSLM are the ones built with the CBOW algorithm, using a context window size of 5 and 600-dimensional vectors for the monolingual Spanish model and, due to memory constraints, 200-dimensional vectors for the bilingual English-Spanish model. We denote these system variants as Lehrer+SSLMmo, Lehrer+SSLMbi, or Lehrer+SSLMbi&mo depending on whether the monolingual, the bilingual, or both models are in use, respectively. For tuning the weights of the SSLM features of these system variants, we resort to performing manual grid searches with the newsCommentary2009 development set.[9] This is because, even though Stymne et al. (2013) reported some initial success on automatic feature weight optimization on Docent, weight optimization for the document-level features still persists as a hard problem (Smith, 2015). For instance, Figures 4.1, 4.2, and 4.3 show the evolution of the feature weights during three of the experiments[10] on MERT tuning that we conducted. The first two correspond to tuning only sentence-level features on Lehrer and Moses, respectively, and both end successfully and with similar results. The latter figure corresponds to tuning also the document-level SSLM features on Lehrer and, in this case, ends unsuccessfully: the process exhausts its quota of iterations without having converged.

Finally, the experiments are conducted over the newsCommentary2010 test set.

[8] For completeness, the used weight are as follows: $\langle 0.142478, 10^{30} \rangle$ for *geometric-distortion-model*, -0.305865 for *word-penalty*, 100 for *oov-penalty*, 0.0928037 for *phrase-penalty*, 0.157589 for *ngram-model*, and $\langle 0.092929, 0.071066, 0.0911327, 0.0461362 \rangle$ for *phrase-table*. Notice that they do not add up to 1, since the 10^{30} and 100 values are not obtained through the tuning process but are left to their default configuration (Hardmeier, 2014).

[9] The grid search is performed by trying weights for SSLM at regularly-spaced values, leaving the remaining feature functions with their MERT-tuned values of the Lehrer baseline. After an initial exploration, the most promising region is further analyzed with finer-spaced values. When combining the monolingual and the bilingual models together, the cartesian product of their possible values is considered for the grid search. The resulting non-normalized weights for the additional feature functions are as follows: 0.03 for the +SSLMbi variant, 0.015 for the +SSLMmo, and $\langle 0.03, 0.015 \rangle$ for their combination +SSLMbi&mo.

[10] For these experiments, we use newsCommentary2009 as development set, and trim the phrase table such that each source phrase only contains, at most, its 30 entries with highest $p(tgt|src)$ probability. Also, Lehrer is configured to start the decoding with a random initial translation and to perform 10^6 hill-climbing steps with a maximum quota of 10^5 consecutive unsuccessful steps. When introducing the document-level SSLM features, the number of steps is lowered to $2 \cdot 10^5$ due to time constraints. Finally, since we are interested in ruling out the small amount of documents of the development set as the cause for non-convergence, we also conduct tunings on Moses with a limited development set: we use only the first 136 sentences of the set, to match the 136 documents available when tuning on Lehrer. This limited set is the one in use for Figure 4.2. In the three presented figures, the initial weights of the features are uniformly distributed. We also conducted unsuccessful MERT tuning experiments for SSLM where the initial weight distribution corresponded to the grid-optimized values.

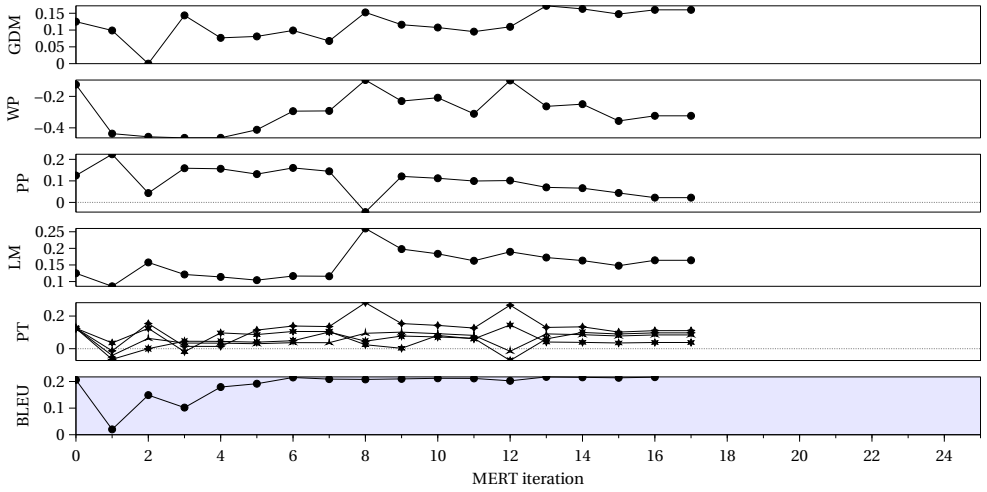

Figure 4.1: MERT on Lehrer on sentence-level features. The plots depict the evolution of the weights for the tuned feature functions (from top to bottom: geometric-distortion-model, word-penalty, phrase-penalty, ngram-model, and the 4-score phrase-table) and also the evolution of the respective obtained BLEU score (bottom). The process has already converged at its 17th iteration (of a maximum of 25).

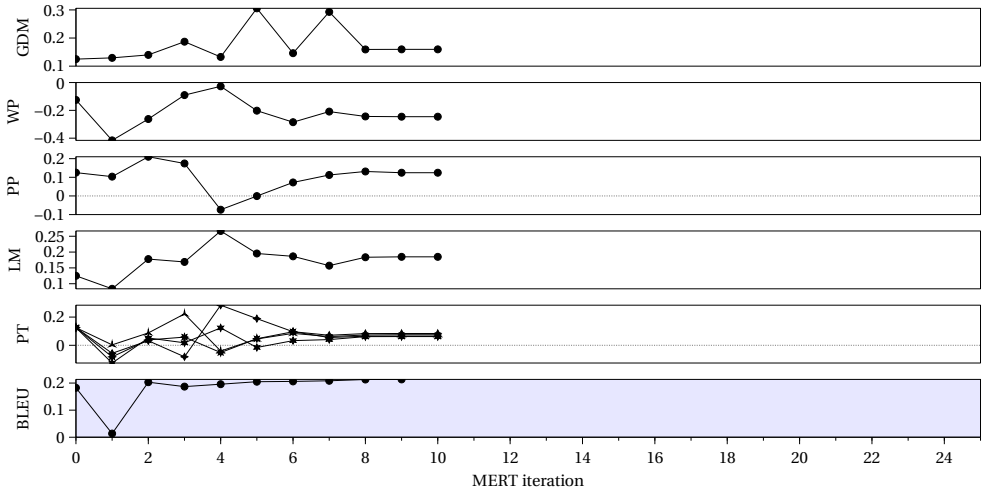

Figure 4.2: MERT on Moses, with analogous interpretation as Figure 4.1. The process has already converged at its 10th iteration.

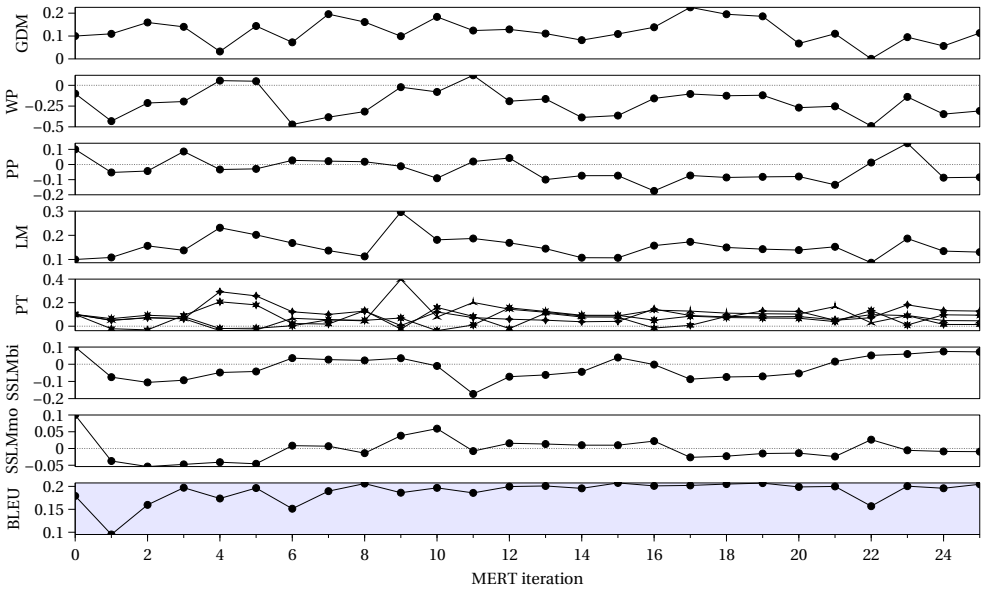

Figure 4.3: MERT on Lehrer, with analogous interpretation as Figure 4.1, but adding the document-level features SSLMbi and SSLMmo to be tuned. The process reaches its final 25th iteration without having converged.

4.4.2 Results

Table 4.4 shows the automatic evaluation obtained with the Asiya toolkit (Gonzàlez et al., 2012) for several lexical metrics and their ULC average. The results show only small differences between the systems. However, these differences roughly reflect the impact of our word embeddings in the translation process and are fairly consistent across metrics. Nevertheless, the differences are only statistically significant[11] between the Lehrer and Lehrer+SSLMbi systems. We observe that Lehrer systems have in general a positive trend in their performance as long as we introduce models with more information (from only monolingual to bilingual).

Looking a little bit closer at each system, first note that switching from the sentence-based Moses decoder to the document-based Lehrer baseline does not lead to a clear change: the scores of the PER, BLEU, and $\text{METEOR}_{\text{pa}}$ metrics improve, whereas WER, TER, and NIST worsen, with the ULC global ranking showing a slight preference of 0.07 points for the Lehrer system. A similar phenomenon takes place when comparing the Lehrer baseline with the Lehrer+SSLMmo variant: the latter only improves in WER, TER, NIST, and ULC, but gets worse scores in the others. On the other hand, using bilingual models seems to cause a clearer improvement: Lehrer+SSLMbi obtains the best scores in all the metrics except for WER and PER, where it is surpassed by other Lehrer variants. In particular, it improves

[11] According to bootstrap resampling (Koehn, 2004) over BLEU and NIST metrics with a p-value of 0.1.

System	WER↓	PER↓	TER↓	BLEU↑	NIST↑	METEOR$_{pa}$↑	ULC↑
MOSES	59.54	39.95	53.70	27.52	7.3229	50.02	49.92
LEHRER	59.67	**39.72**	53.78	27.58	7.3127	50.08	49.99
+SSLMbi	59.38	39.84	**53.49**	**27.60**	**7.3491**	**50.13**	**50.22**
+SSLMmo	59.58	39.83	53.70	27.57	7.3194	50.07	50.00
+SSLMbi&mo	**59.37**	39.97	**53.49**	27.48	7.3436	50.10	50.07

Table 4.4: Automatic evaluation of the systems. The ULC is computed over the other metrics of the table.

System	news item					
	12	**37**	**35**	**107**	**41**	**97**
MOSES	11.94	37.46	30.73	30.77	37.55	26.96
LEHRER	12.56	38.28	31.23	30.81	37.33	26.80
+SSLMbi	12.02	38.21	32.19	33.03	37.61	27.48
+SSLMmo	11.99	38.18	31.87	31.38	37.79	27.65
+SSLMbi&mo	12.27	37.85	32.15	31.38	37.21	27.57

Table 4.5: Scores of the automatic evaluation of the different systems using the BLEU metric on some individual documents from the test set.

0.30 points in ULC with respect to the MOSES baseline and 0.13 with respect to the LEHRER baseline. This seems in contradiction with the statistical significance test, which only detected a difference of LEHRER+SSLMbi against the LEHRER baseline, but not against the MOSES one. The cause for this apparent discrepancy is that the NIST value for the former baseline is worse than the value for the latter. Finally, the variant using both monolingual and bilingual embeddings together obtains mixed results: according to the ULC ranking, the scores of LEHRER+SSLMbi&mo seem to be at a midpoint between the LEHRER+SSLMmo and LEHRER+SSLMbi systems.

In summary, we conclude from these results that the semantic information captured by our vector models helps the document-level translation decoder. This behaviour is coherent with the previous evaluation of the models shown in Sections 4.2 and 4.3, where bilingual models outperformed their monolingual Spanish counterpart. Also, this is an expected behaviour since the systems including the bilingual SSLM models are the ones that include source context information which is more reliable than the target context information which is being generated from the same system that the SSLM want to improve.

Table 4.5 shows the BLEU scores for some particular documents with some interesting cases. These results reflect the behaviour of our systems. We found some documents where the LEHRER systems cannot improve the MOSES translation. For instance, the phrase "*the portrait*" appears in a document about a famous photographer. Its correct translation would be "*el retrato*" according to the reference, although "*el cuadro*" would be also a correct translation depending on the context. MOSES translates it as "*el cuadro*" and LEHRER systems suggest the reference translation but also introduce a new incorrect option "*el marco*". On the other hand, we find many ex-

amples where word vector models are helping. For instance, in the example of *desk* that we mentioned in Section 4.3, it is translated as *mostrador*, *mesa*, and *escritorio* by MOSES. Using the LEHRER baseline, it appears translated as *escritorio* and *mesa*. That shows how LEHRER is controling the coherence level of the translation. Using the LEHRER extended with the monolingual model, it appears as *escritorio*, *mesa*, and *taquilla*. The word vector language model helps the system change one translation option for a more correct one. Finally, using the bilingual vector model, we observe the word translated as *mostrador*, *mesa*, and *taquilla*, obtaining here 2 good translations instead of only one. This shows how the bilingual information helps to obtain better translations. We observe how monolingual vector models improve the LEHRER base translation and, at the same time, how the bilingual information helps to improve the translation and even obtain better results than the ones with the MOSES baseline.

4.5 Conclusions

We have presented an evaluation of word vector models trained with neural networks. First, we build monolingual and bilingual models using the WORD2VEC package. Then, we evaluate the models carrying out two different evaluation tasks. One to assess the quality of the semantic relationships enclosed in the models and, in the second one, we test the models to see their capability to select a good translation option for a word that appears translated in more than one sense in a first translation of a document. The results of these evaluations show that the CBOW bilingual model performs better than the Skipgram one in our test set, achieving 85.49% and 78.36%, respectively, for the accuracy at Top 5. Also, the bilingual model achieves better results than the monolingual one, with a 65.44% of accuracy for the best monolingual model trained with CBOW against the 85.49% for the bilingual model under the same conditions. These results indicate that word embeddings can be useful for translation tasks.

We also evaluated our word vector models inside a machine translation system. In particular, we chose the LEHRER decoder since it works at document-level and allows a fast integration of word embeddings as semantic space language models. This option allows us to assess the vector models quality in a specific translation environment. The experiments we carried out showed that word vector models can help the decoder improve the final translation. Although we only observe a slight improvement in the results in terms of automatic evaluation metrics, the improvement is consistent among metrics and is larger as we introduce more semantic information into the system. That is, we get the best results when using the models with bilingual information.

Summing up, the evaluation has shown the utility of word vector models for translation-related tasks. However, the results also indicate that these systems can be improved.

Chapter 5

Lexical Consistency in Statistical Machine Translation

Here we focus on granting the "one sense per discourse" assumption. Chapters 3 and 4 have presented our first attempts to that end, building on standard SMT systems but without modifying their internal architecture. We now focus on a specific, significant document-level phenomenon: lexical consistency. We tackle it by designing and integrating new strategies within the decoder itself.

As pointed out in Section 3.1, lexical consistency in a translation can be increased by translating a word always in the same way. However, even though term repetition may improve the coherence of the text, it can also mar the translation by making it more monotonous and tedious. Instead of that, we are going to allow certain variability as long as the translated words are consistent with their context. In this scenario, using word vector models helps maintain such consistency as they can give a measure of semantic distance between a word and its context.

5.1 Approach

We strive to obtain lexically consistent translations, only allowing term variations as long as they are semantically similar to their surrounding context (see Figure 5.1). To this end, we need a document-oriented decoder, since identifying translation inconsistencies requires access to the entire document translation. The approach by Hardmeier et al. (2012) detailed in Section 2.2.2 is a suitable framework for our purposes. Inspired by the SSLMs and with these aims, we introduce a new feature function that uses a Semantic Space to measure the Lexical Consistency (SSLC) of a document translation. Following the idea of a not strict lexical consistency, the SSLC uses word embeddings to measure how suitable the translation of a word is taking into account

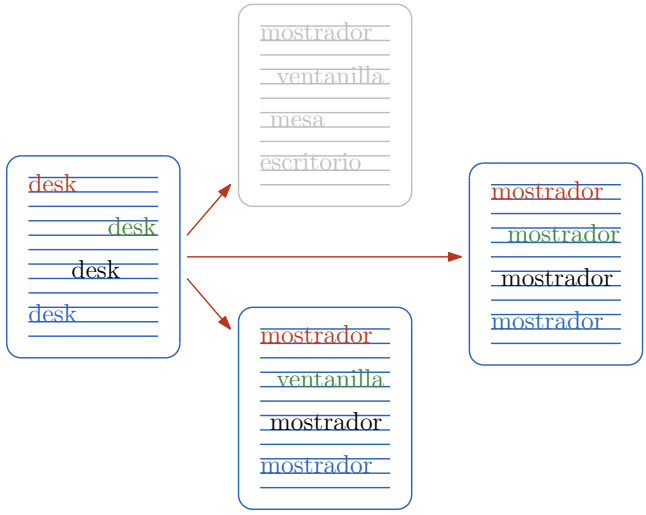

Figure 5.1: Example where the source document (left) is translated with differing options for the word "desk". The greyed-out target document (top) is undesirable for us since the translation options used for "desk" are inconsistent. Moreover, the specific options "mesa" and "escritorio" are not adequate in Spanish assuming that the context of the document is related to a service attending rent grievances: they are not likely to occur in such setting. The other two target documents (right and bottom) are valid for us. Notice that the bottom one also uses multiple translation options, but both "mostrador" and "ventanilla" are adequate for the context.

its context and the other translation options seen through the document.

However, the SSLC may not be enough to help the decoder obtain better translations in terms of consistency. Recall from Section 2.2.2 that the decoding performs a hill climbing in a translation search space. At each step, the decoder explores the neighborhood of the current translation by randomly applying to it one of the available change operations. The default operations perform simple modifications such as changing the translation of a phrase, swapping phrase-pairs, or re-segmenting the data. Unfortunately, these simple operations do not aid directly in our goal of reaching more lexically consistent translations. The reason for this fact is twofold. On the one hand, to increase the consistency it is in general necessary to perform multiple changes within the document and, since the default change operations only perform one change at a time, it would take several steps to fix one of the lexical choice inconsistencies. On the other hand, since hill climbing only performs a step when it strictly increases the score, each of the intermediate steps that try to fix an inconsistency would need to increase the score. Here arises the necessity of introducing a new change operation. We implement the Lexical Consistency Change Operation (LCCO) that shortcuts the process by, at a single step, performing multiple simultaneous changes that fix inconsistent translations of the same source word.

5.2 Semantic Space Lexical Consistency Feature

Intuitively, SSLC scores each occurrence of an inconsistently translated source word with a value in $[-\infty, 0]$. This value is intended to measure how worse (in terms of adequacy) the current translation option is when compared to the other translation options seen in the document. We consider a translation option to have better adequacy the more semantically similar it is to the context surrounding the occurrence being scored. We compute this semantic similarity as a cosine similarity between two vectors: (*i*) the word embedding of the translation option and (*ii*) the vector representation of its context within the target document. A vector representation of the context is obtained in our case as the sum of all its word vectors. Recall that the aim of SSLC is not to enforce a strict lexically consistent translation, since we allow lexical inconsistencies when they are semantically similar to their surrounding context.

As a first step, we need to define one basic property: when two words must be considered to be the same word. To this end, we use a criterion looser than the strict identity: we allow to conflate words having distinct casing and differing on certain kinds of inflection. This is necessary to properly identify ambiguous words; for instance: if "desk" is being translated into Spanish as "mostrador" whereas "desks" is being translated as "mesas", we want to identify that the common stem of "desk" and "desks" is obtaining two distinct translated stems. More precisely, we define a normalization process for words, and consider two words to be the same one if, and only if, their normalized forms coincide. Formally, we introduce the functions $norm_{src}$ and $norm_{tgt}$ which take as input a source or target word, respectively, and return a normalized version of it. In our settings, $norm_{src}$ and $norm_{tgt}$ are implemented by, first, lower-casing the word and, then, by stemming it with the Snowball library[1] for the appropriate language. For example, with English source we get $norm_{src}(\text{Penguins}) = \text{penguin}$, where "P" has been lower-cased to "p" and the plural suffix "s" has been erased by the stemming.

Second, to formalize SSLC we still need some preliminary artillery. Let the source and target documents be the sequences of words $s_1, s_2, \ldots, s_N$ and $t_1, t_2, \ldots, t_M$, respectively, for some $N, M > 0$. Recall that source and target words are aligned with an arbitrary relation, say $\mathcal{T}$. In particular, one source word may be aligned to zero, one, or more target words, and conversely, one target word may be aligned to zero, one, or more source words. We focus only on the one-to-one aligned words, i.e., source words that are aligned to just one target word and, in turn, such target word is only aligned with that source word. We denote by τ this one-to-one sub-relation of $\mathcal{T}$, hence $\tau : \{1, \ldots, N\} \to \{1, \ldots, M\}$ is a partial, injective mapping that associates to a source word index its corresponding target word index. τ is partial since it is not defined for those source word indexes aligned to zero or more than one target words, and it is injective since we only consider target words aligned to exactly one source word.

In order to identify inconsistently translated words, we first need to identify all occurrences of the same source word. To this end, let $occ : \{1, \ldots, N\} \to 2^{\{1,\ldots,N\}}$ be the function that associates to each source word index i the set of indexes of the

[1] `http://snowballstem.org/`

source words that have the same normalized form as s_i, i.e.:

$$occ(i) = \{j \in \{1, \ldots, N\} \mid norm_{src}(s_j) = norm_{src}(s_i)\}$$

Observe that $i \in occ(i)$ always holds. Additionally, we need to obtain the target word indexes aligned to the source word indexes in $occ(i)$. To this end, we use $\tau occ(i)$ to denote the set of target word indexes resulting of applying τ to each source word index in $occ(i)$, skipping those for which τ is undefined. Finally, we say that the ith source word is *inconsistent* in the current translation, denoted $incons(i)$, if the source words s_j that have the same normalized form as s_i have been translated into more than two distinct normalized targets. Formally:

$$incons(i) = \left(\left|\{norm_{tgt}(t_j) : j \in \tau occ(i)\}\right| > 2\right)$$

We are now in the position to define the associated score for inconsistent words. First, let μ be the mapping defined by the word vector model in use by the decoder. Recall that these models are a projection that maps words to vectors in a certain space $\mathbb{R}^n$, for some $n > 0$. Second, let $C > 0$ be the size of the context to either side of the target word, possibly crossing sentence boundaries. The vector representation for the context of the jth target word is defined as the sum of the word embeddings around j:

$$ctxt(j) = \sum\nolimits_{k \in \{\max(1, j-C), \ldots, j-1, j+1, \ldots, \min(j+C, M)\}} \mu(t_k)$$

And third, the *score* associated to the ith source word, denoted $score(i)$, computes the difference between two similarities: the similarity of the current translation option $t_{\tau(i)}$ and its context $\vec{c} := ctxt(\tau(i))$ minus the similarity of $\vec{c}$ and the translation option which is closest to $\vec{c}$, i.e., the t_k whose similarity to $\vec{c}$ is maximal, with $k \in \tau occ(i)$. More precisely:

$$score(i) = \begin{cases} 0 & \text{if } i \notin \mathrm{dom}(\tau) \vee \neg incons(i) \\ sim(\vec{c}, \mu(t_{\tau(i)})) - \max\limits_{k \in \tau occ(i)} sim(\vec{c}, \mu(t_k)) & \text{otherwise} \end{cases}$$

where sim of two vectors is the natural logarithm of their cosine similarity linearly scaled to the range $[0, 1]$, i.e.:

$$sim(\vec{a}, \vec{b}) = \ln\left(\frac{1}{2}\frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\|\|\vec{b}\|} + \frac{1}{2}\right) \tag{5.1}$$

Note that sim ranges in $[-\infty, 0]$, with $-\infty$ corresponding to the case where the vectors are diametrically opposed (semantically distant) and 0 to the case where they have the same orientation (semantically close).

The final SSLC score for the whole document simply adds together the individual scores: $\sum_{i=1}^{N} score(i)$.

As a final remark, notice that for ease of presentation we have assumed that the word vector model is monolingual. If it were bilingual, the expressions like $\mu(t_j)$ would be $\mu(t_j, s_{\tau^{-1}(j)})$ instead. Also, unknown words for the vector model, i.e., words w such that $\mu(w)$ is undefined, are ignored when computing the scores, and not taken into account when considering the C-sized context of the target words.

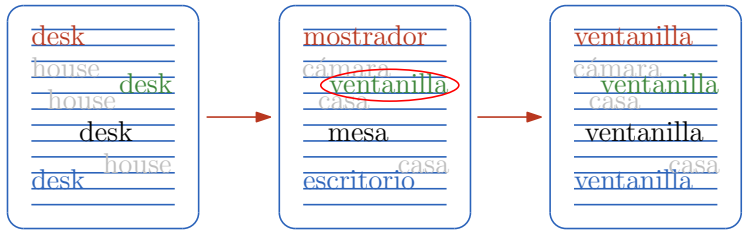

Figure 5.2: Sketch of the behaviour of LCCO. The source document (left) has both "desk" and "house" translated into different forms (middle). LCCO selects "desk" as the inconsistent source word to fix, and chooses its second occurrence as the translation (the encircled "ventanilla") to use in its remaining occurrences. The target document resulting from the LCCO modifications (right) has a consistent translation for "desk".

5.3 Lexical Consistency Change Operation

Intuitively, LCCO first randomly selects an inconsistently translated source word, then, randomly chooses one of its translation options used in the document, and finally, re-translates its occurrences throughout the document to match the chosen translation option (see Figure 5.2). This random behaviour is important to allow the hill climbing performed by the decoder to properly explore the neighborhood.

In order to formalize LCCO we need a more refined view of the source and target documents than in Section 5.2. Nevertheless, we will reuse some of the previous definitions where possible. Since the decoder works with phrases as its minimum translation units, the documents are processed as sequences of phrases. Hence, we now consider that all the s_i and t_j are phrases instead of words. The definition of τ is still the same as before, although we can now guarantee that it is a total bijection since the decoder works with phrase-pairs. The functions $norm_{src}$ and $norm_{tgt}$ are similar to before but have phrases as input and output instead of single words. Also, we consider that they normalize each word of the input phrase individually and, in particular, that they preserve the number of words in the phrase.

The goal of LCCO is to change the translation of inconsistently translated words but, since the decoder works with phrases, we focus on only changing those inconsistent words appearing in 1-word phrases. This does not hinder our goals, as the other change operations of the decoder can resegment the data and, in this way, isolate for LCCO any inconsistent words appearing in multi-word phrases. For this reason, let us now consider a more restricted definition of *occ* that only deals with indexes of source phrases having a single word. That is, for any $i \in \{1, \ldots, N\}$ we have:

$$occ(i) = \{j \in \{1, \ldots, N\} \mid norm_{src}(s_j) = norm_{src}(s_i) \ \wedge \ |s_j| = 1\}$$

where $|s_j|$ is the number of words in the source phrase s_j. Note that $i \notin occ(i) = \emptyset$ if the source phrase s_i has more than one word. Using this redefined *occ*, we can keep the same definition for τocc and *incons* as before.

LCCO works as follows. First, it selects a source phrase index $i \in \{1, \ldots, N\}$ such that $incons(i)$ is true. This is done by uniformly drawing that i from the following

set:

$$\{k \in \{1,\ldots,N\} \mid incons(k) \ \wedge \ \forall k' \in occ(k) : (k \leq k')\}$$

where the universally-quantified condition is simply used to pick one single representative from each set $occ(k)$, in particular, the one with the least index (although any other would work too). Using such representatives is important to guarantee that the selection is uniform on the distinct inconsistent source phrases, without biasing the selection towards the ones with most occurrences in the source document. Second, it selects a specific occurrence $j \in occ(i)$ of that source phrase and considers $t_{\tau(j)}$ as the translation to use in the other occurrences. This is done by uniformly drawing that j from the following set:

$$\{k \in occ(i) \mid \forall k' \in occ(i) : (k \leq k' \ \vee \ norm_{tgt}(t_{\tau(k)}) \neq norm_{tgt}(t_{\tau(k')}))\}$$

where the universally-quantified condition is, again, simply used to pick a single representative (the one with the least index, although any other would work too) from each subgroup of $occ(i)$ whose corresponding target phrases have an identical normalized form. As before, the use of representatives guarantees that the selection is not biased towards the translation options with most occurrences in the document. Finally, the new document translation $t'_1, t'_2, \ldots, t'_M$ is obtained by setting for each $k \in \{1,\ldots,M\}$:

$$t'_k := \begin{cases} t_k & \text{if } k \notin \tau occ(i) \\ t_k & \text{if } k \in \tau occ(i) \wedge norm_{tgt}(t_k) = norm_{tgt}(t_{\tau(j)}) \\ t_k & \text{if } k \in \tau occ(i) \wedge \nexists t \in \rho(s_{\tau^{-1}(k)}) : norm_{tgt}(t) = norm_{tgt}(t_{\tau(j)}) \\ t & \text{else, with random } t \in \rho(s_{\tau^{-1}(k)}) \text{ such that } norm_{tgt}(t) = norm_{tgt}(t_{\tau(j)}) \end{cases}$$

where ρ maps a source phrase to the set of target phrases that are its possible translations according to the phrase table in use by the decoder. Note that t'_k and t_k coincide in the three first cases of the definition. In the first one, this is simply because the target phrase index k is not aligned through τ to any of the source phrase indexes in $occ(i)$ that are being affected. The second and third cases do involve an affected index k, but in the second one we already have a target phrase with the same normal form as the desired $t_{\tau(j)}$ and in the third case the phrase table has no translation option for the corresponding source phrase $s_{\tau^{-1}(k)}$ with the same normal form as $t_{\tau(j)}$. In other terms, in the second case it is unnecessary to perform any change since we already have the desired translation, whereas in the third one it is not possible to perform the change. The third case would never arise if $norm_{src}$ had been defined as the identity. The fourth and final case is the only one that alters the kth target phrase: it involves an affected index k, containing a translation t_k with different normal form than the desired $t_{\tau(j)}$, and the phrase table contains some translation options for $s_{\tau^{-1}(k)}$ with the same normal forms as $t_{\tau(j)}$. Thus, in this fourth case it suffices to set t'_k to a t uniformly drawn from the available options in the phrase table.

5.4 Experiments

We conduct English-to-Spanish translation experiments building on the settings detailed in Chapter 4. In particular, we reuse the baseline Moses and Lehrer systems

from Sections 4.3.1 and 4.4.1, respectively, and for a more complete comparison we consider again the three LEHRER system variants from Section 4.4.1 that implement the SSLM with monolingual and bilingual word embeddings. Furthermore, here we also use the same development and test sets as the ones specified there.

Besides the systems inherited from the previous chapter, we introduce several new variants of LEHRER implementing the SSLC feature function and further ones with the LCCO change operation. Overall, we analyze the performance of 17 systems, comprising the standard baseline MOSES from Section 4.3.1, the following 8 variants of LEHRER:

- the baseline LEHRER system from Section 4.4.1,
- the three systems from Section 4.4.1 that implement the SSLMs within LEHRER using either the bilingual (+SSLMbi), the monolingual (+SSLMmo), or both (+SSLMbi&mo) embeddings,
- two new systems implementing our SSLC feature within LEHRER using the same bilingual embeddings as the SSLMbi in Section 4.4.1 (+SSLCbi) and its combination with both SSLM features (+SSLMbi&mo+SSLCbi), and
- two new systems implementing our SSLC feature using the same monolingual embeddings as the SSLMmo in Section 4.4.1 (+SSLCmo) and its combination with both SSLMs (+SSLMbi&mo+SSLCmo),

and another 8 new analogous variants of LEHRER+LCCO (which we denote with equivalent names).

We tried several values for the context size parameter C and decided to fix $C = 15$ for the experiments as a good trade off between performance and results and to assure that the context is beyond sentence scope. To avoid extra noise in the process, we use a list of stop-words that are filtered out from the scoring of the SSLC and not considered for changing by the LCCO. Thus, the scoring and changes are only applied to content words.

For tuning the weights of the document-level features in the +SSLC system variants, we again resort to performing manual grid searches due to the difficulty commented in Section 4.4.1 of applying automatic methods (see Figure 5.3).[2] For the +LCCO variants, we additionally perform a manual grid search to optimize the weights for the change operations in use. In particular, we adjust the weights for the default change operations (change-phrase-translation, swap-phrases, and resegment) and for LCCO in the LEHRER+LCCO system, and use the resulting weights in all its variants.[3]

[2]The grid search is performed along the same lines as the one in Section 4.4.1. The resulting non-normalized weights for the document-level features in the extensions of the LEHRER system variants of Section 4.4.1 are as follows: 0.01 for the new +SSLCbi variant, 0.006 for +SSLCmo, $\langle 0.03, 0.015, 0.01 \rangle$ for +SSLMbi&mo+SSLCbi, and $\langle 0.03, 0.015, 0.03 \rangle$ for +SSLMbi&mo+SSLCbi. When adding LCCO, we re-tune again the document-level features, obtaining: 0.035 for +SSLMbi, 0.011 for +SSLMmo, $\langle 0.035, 0.001 \rangle$ for +SSLMbi&mo, 0.25 for +SSLCbi, 0.06 for +SSLCmo, $\langle 0.035, 0.001, 0.2 \rangle$ for +SSLMbi&mo+SSLCbi, and $\langle 0.035, 0.001, 0.08 \rangle$ for +SSLMbi&mo+SSLCbi.

[3]The tuned weights for the change operations are 0.45 for change-phrase-translation, 0.1 for swap-phrases, 0.4 for resegment, and 0.05 for LCCO. For comparison, the weights for the default change operations alone, as reported by Hardmeier (2014), are 0.8 for change-phrase-translation, 0.1 for swap-phrases, and 0.1 for resegment.

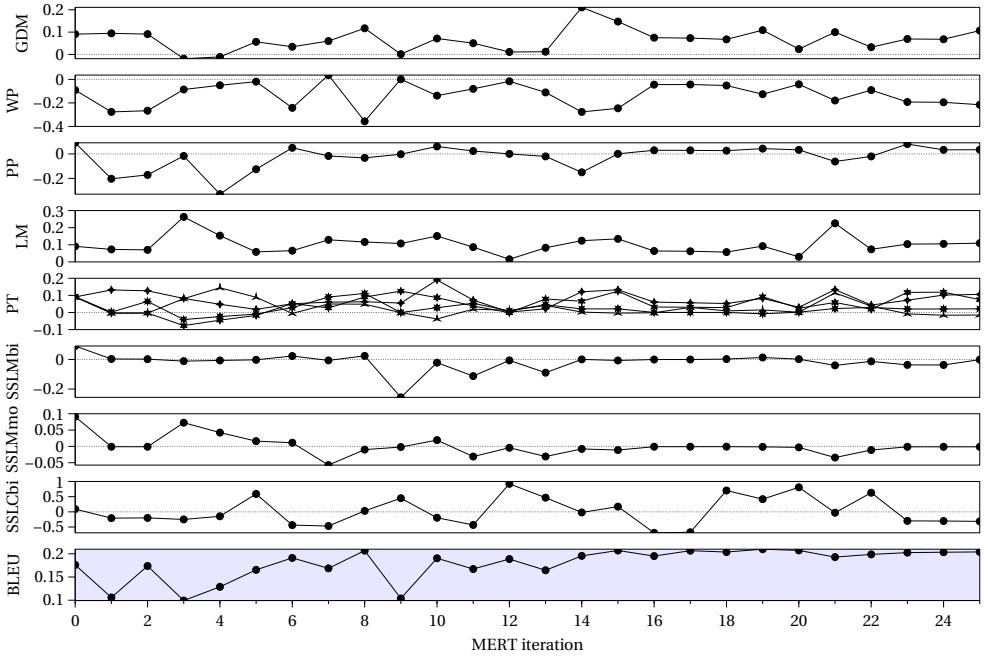

Figure 5.3: MERT on Lehrer, with analogous interpretation as Figure 4.3, but adding the document-level feature SSLCbi to be tuned. The process reaches its final 25th iteration without having converged.

5.4.1 Automatic Evaluation

We carry out an automatic evaluation using again the Asiya toolkit (Gonzàlez et al., 2012) and the same metrics as in Section 4.4.2.

In Tables 5.1 and 5.2 we show the performance of the systems. On the development set, results without LCCO show that bilingual information in SSLM appears to be more helpful than monolingual, but also seems that both kinds of models can work together to improve the final system output, as already seen in Section 4.4.2. Looking at the results for both SSLC systems, there are almost no noticeable differences with respect to baseline Lehrer. The best results have been obtained combining all the information: bilingual and monolingual SSLMs with either of the SSLCs. When introducing LCCO, we observe more or less the same trends as before, except that combining SSLC and SSLM does not seem to provide the same benefit. On the test set we observe a similar behaviour, although differences among system scores are smaller. In this occasion both SSLC appear to improve the baseline Lehrer. Note that, as in the development set, both SSLC seem to work better in combination with SSLM, even though now the trend is reversed in some of the metrics, like BLEU.

As a general remark, the differences between most of the systems are not statistically significant.[4] Several causes contribute to this effect. On the one hand, a pairwise

[4]According to bootstrap resampling (Koehn, 2004) over BLEU and NIST metrics with a p-value

System	WER↓	PER↓	TER↓	BLEU↑	NIST↑	METEOR$_{pa}$↑	ULC↑
Moses	64.17	43.10	58.28	24.27	6.8264	46.84	49.96
Lehrer	64.30	43.34	58.34	24.28	6.8199	46.92	49.84
+SSLMbi	64.05	42.90	58.08	24.35	6.8451	46.93	50.26
+SSLMmo	64.21	43.18	58.28	24.27	6.8272	46.89	49.95
+SSLMbi&mo	63.96	42.83	58.01	24.36	6.8535	46.91	50.35
+SSLCbi	64.33	43.36	58.38	24.26	6.8165	46.90	49.79
+SSLCmo	64.34	43.34	58.37	24.24	6.8182	46.91	49.79
+SSLMbi&mo+SSLCbi	**63.91**	**42.79**	**57.99**	**24.39**	6.8607	**46.95**	**50.43**
+SSLMbi&mo+SSLCmo	63.93	**42.79**	**57.99**	24.37	**6.8629**	**46.95**	50.42
Lehrer+LCCO	64.30	43.33	58.36	24.27	6.8194	46.92	49.83
+SSLMbi	63.99	42.85	58.04	**24.38**	6.8489	**46.94**	50.34
+SSLMmo	64.23	43.21	58.29	24.27	6.8247	46.91	49.93
+SSLMbi&mo	**63.98**	42.84	58.04	24.35	6.8480	46.92	50.32
+SSLCbi	64.30	43.31	58.36	24.25	6.8189	46.89	49.81
+SSLCmo	64.29	43.32	58.35	24.27	6.8194	46.91	49.84
+SSLMbi&mo+SSLCbi	64.00	42.85	58.06	24.34	6.8460	46.93	50.30
+SSLMbi&mo+SSLCmo	63.99	**42.80**	**58.03**	24.36	**6.8510**	46.92	**50.35**

Table 5.1: Automatic evaluation of the systems on the development set. The ULC is computed over the other metrics of the table.

System	WER↓	PER↓	TER↓	BLEU↑	NIST↑	METEOR$_{pa}$↑	ULC↑
Moses	59.54	39.95	53.70	27.52	7.3229	50.02	49.91
Lehrer	59.67	**39.72**	53.78	27.58	7.3127	50.08	49.98
+SSLMbi	59.38	39.84	**53.49**	27.60	**7.3491**	**50.13**	**50.21**
+SSLMmo	59.58	39.83	53.70	27.57	7.3194	50.07	49.99
+SSLMbi&mo	59.37	39.97	**53.49**	27.48	7.3436	50.10	50.06
+SSLCbi	59.63	39.75	53.77	**27.61**	7.3152	50.07	50.00
+SSLCmo	59.66	39.74	53.78	27.59	7.3125	50.07	49.98
+SSLMbi&mo+SSLCbi	**59.36**	39.96	53.50	27.50	7.3436	50.07	50.07
+SSLMbi&mo+SSLCmo	**59.36**	39.96	53.51	27.51	7.3470	50.08	50.08
Lehrer+LCCO	59.67	39.76	53.77	27.57	7.3081	50.07	49.94
+SSLMbi	59.32	39.88	53.45	**27.61**	7.3518	50.14	50.24
+SSLMmo	59.60	39.79	53.71	27.58	7.3195	50.09	50.01
+SSLMbi&mo	**59.29**	39.86	**53.43**	27.60	**7.3554**	**50.15**	**50.27**
+SSLCbi	59.70	**39.75**	53.81	27.59	7.3097	50.07	49.94
+SSLCmo	59.63	39.76	53.77	27.59	7.3114	50.07	49.97
+SSLMbi&mo+SSLCbi	59.32	39.89	53.46	27.57	7.3508	50.12	50.20
+SSLMbi&mo+SSLCmo	59.35	39.90	53.47	27.57	7.3481	50.12	50.18

Table 5.2: Automatic evaluation of the systems on the test set. The ULC is computed over the other metrics of the table.

comparison of all the system outputs shows that the amount of different sentences is only between 8% and 42%. On the other hand, SSLC and LCCO deal with very sparse phenomena, and thus, they cannot have a huge impact on the automatic metrics. For instance, in average, LCCO is applied on 8% of the documents[5] on the development and test sets, and in those cases it comprises between 4% and 9% of the total amount of change operation applications.[6] Nevertheless, this does not necessarily hinder our goals, as consistent lexical selection improvements can also be introduced by the default change operations (although taking more search steps in decoding than LCCO, as the latter performs several modifications at once), which are promoted by SSLC.

These results make necessary a human evaluation of the translations, since we expect that the few changes induced by SSLC and LCCO will be appreciated by humans.

5.4.2 Human Evaluation

We carry out two distinct evaluation tasks. The first one tries to assess the quality of the different systems, working with and without LCCO. The second one is a small document-level evaluation task that compares the adequacy of the lexical choices between pairs of system variants that differ on whether they use LCCO or not.

For the first evaluation task, we select a common subset of sentences from the test set translated by the Moses system and by the 8 variants of the Lehrer system. More precisely, we randomly choose 100 sentences with at least 5 and at most 30 words, and with at least 3 different translations among all the considered system outputs. We set up an evaluation environment where 3 native Spanish annotators with a high English level have been asked to rank the output of all the systems for each of the 100 selected sentences, from best to worst general translation quality and with possible ties. System outputs were presented in random order to avoid system identification. The same evaluation procedure is also carried out with the 8 variants of Lehrer+LCCO. Table 5.3 shows the results obtained, where each entry of the table contains the mean number of times that the row system is better/worse than the column system according to the annotators, the remainder being ties. For the ranking with Lehrer variants, the annotators agreed 70% of the time when ranking two distinct outputs, and for Lehrer+LCCO, they agreed 72% of the time,[7] respectively reaching $\kappa = 0.4362$ and $\kappa = 0.4623$ (Fleiss, 1971) showing in both cases a "moderate" inter-annotator agreement (Landis and Koch, 1977).

of 0.05. In particular, the only statistical differences found are between Lehrer+LCCO and its variants +SSLMbi and +SSLMbi&mo and, additionally, between its variant +SSLMbi&mo and the variants +SSLCbi and +SSLCmo.

[5]The LCCO is, in fact, applied on all the documents multiple times, but most applications are unsuccessful (i.e., unable to improve the score, and thus, rejected by the hill climbing). The reported amount corresponds to the percentage of documents where there has been, at least, one successful application of LCCO during the translation process.

[6]Similarly to the previous footnote, the two reported percentages are computed over the successful applications of the change operations only, disregarding the vast majority of the unsuccessful ones.

[7]These agreements are computed as follows. For each pair of annotators, for each pair of systems, and for each sentence in the sample where the two systems have produced distinct output, we consider that the annotators agree if they have given the same relative ranking to both outputs, otherwise they disagree. The reported amounts are simply the percentage of agreements among the total.

ID	System	1	2	3	4	5	6	7	8	9
1	MOSES	-	39 / 39	**44 / 43**	35 / 45	38 / 48	37 / 41	**43 / 39**	36 / 47	40 / 46
2	LEHRER	39 / 39	-	28 / 32	24 / 28	37 / 40	11 / 14	**14 / 11**	35 / 45	34 / 44
3	+SSLMbi	43 / 44	**32 / 28**	-	**36 / 33**	34 / 34	33 / 34	**37 / 29**	23 / 34	23 / 34
4	+SSLMmo	**45 / 35**	**28 / 24**	33 / 36	-	31 / 35	**31 / 30**	**32 / 26**	27 / 38	26 / 39
5	+SSLMbi&mo	**48 / 38**	**40 / 37**	34 / 34	**35 / 31**	-	**42 / 36**	**44 / 36**	18 / 27	20 / 25
6	+SSLCbi	**41 / 37**	**14 / 11**	**34 / 33**	30 / 31	36 / 42	-	**13 / 8**	34 / 43	36 / 45
7	+SSLCmo	39 / 43	11 / 14	29 / 37	26 / 32	36 / 44	8 / 13	-	31 / 47	33 / 47
8	+SSLMbi&mo+SSLCbi	**47 / 36**	**45 / 35**	**34 / 23**	**38 / 27**	**27 / 18**	**43 / 34**	**47 / 31**	-	**21 / 18**
9	+SSLMbi&mo+SSLCmo	**46 / 40**	**44 / 34**	**34 / 23**	**39 / 26**	**25 / 20**	**45 / 36**	**47 / 33**	18 / 21	-

ID	System	1	2	3	4	5	6	7	8	9
1	MOSES	-	**40 / 38**	44 / 45	39 / 43	41 / 49	36 / 40	39 / 40	40 / 46	**44 / 42**
2	LEHRER+LCCO	38 / 40	-	32 / 40	23 / 32	28 / 38	14 / 19	13 / 19	31 / 41	35 / 38
3	+SSLMbi	**45 / 44**	**40 / 32**	-	38 / 39	21 / 26	**40 / 36**	36 / 36	21 / 28	24 / 26
4	+SSLMmo	**43 / 39**	**32 / 23**	**39 / 38**	-	36 / 37	**31 / 27**	**32 / 26**	34 / 36	**37 / 36**
5	+SSLMbi&mo	**49 / 41**	**38 / 28**	**26 / 21**	**37 / 36**	-	**39 / 34**	**40 / 35**	18 / 24	22 / 23
6	+SSLCbi	**40 / 36**	**19 / 14**	36 / 40	27 / 31	34 / 39	-	**16 / 13**	35 / 40	**36 / 35**
7	+SSLCmo	**40 / 39**	**19 / 13**	36 / 36	26 / 32	35 / 40	13 / 16	-	37 / 44	37 / 37
8	+SSLMbi&mo+SSLCbi	**46 / 40**	**41 / 31**	**28 / 21**	**36 / 34**	**24 / 18**	**40 / 35**	**44 / 37**	-	**21 / 19**
9	+SSLMbi&mo+SSLCmo	42 / 44	**38 / 35**	**26 / 24**	36 / 37	**23 / 22**	35 / 36	37 / 37	19 / 21	-

Table 5.3: The two pairwise system comparisons done in the human evaluation. Each entry is the mean % of times a row system is evaluated better/worse than the column system (in bold if the better times are more than the worse ones).

From the results in Table 5.3, we can say that LEHRER and LEHRER+LCCO are equivalent to MOSES: they have a few ties, and either system is considered better than the other in roughly the same amount of cases. On the other hand, most non-baseline variants of LEHRER and LEHRER+LCCO surpass MOSES on wins. Translations from the systems including the combination of several features appear to be preferred in general; for instance, annotators prefer the combination SSLMbi&mo over SSLMbi or SSLMmo alone. Another interesting detail is that the SSLC systems seem analogous to the corresponding LEHRER and LEHRER+LCCO baselines, as they have many ties (although the SSLC systems have a slight advantage on wins). Also, SSLCbi and SSLCmo seem analogous, with SSLCbi having a slight win advantage over SSLCmo. This fact shows that bilingual information has helped SSLC more than monolingual information. Both combinations of SSLMbi&mo with either of the SSLCs also seem analogous. As final remarks, the SSLMbi&mo+SSLCbi variants of LEHRER and LEHRER+LCCO systematically beat the other systems, and the non-baseline LEHRER and LEHRER+LCCO variants beat their respective baseline variant (except for LEHRER+SSLCmo).

The second, small evaluation task is a comparison between three system pairs with and without LCCO: the baseline, +SSLCbi, and +SSLMbi&mo+SSLCbi variants of LEHRER against the analogous variants of LEHRER+LCCO. We selected 10 documents with lexical changes introduced by LCCO, and asked an annotator to choose the translation with best lexical consistency and adequacy, given the source and two translated documents obtained by a system pair. The annotator preferred the translations of the variants with LCCO 60% of the time, and 20% of the time considered the translations of either system to have the same quality. So, systems with LCCO provided better translations according to the annotator regarding lexical consistency

source:	[...] Due to the choice of the camera and the equipment, these **portraits** remember the classic photos. [...] The passion for the *portrait* led Bauer to repeat the idea [...]
reference:	[...] Son **retratos** que, debido a la selección de la cámara y del material recuerdan la fotografía clásica. [...] La pasión por los *retratos* de Bauer le llevó a repetir la idea [...]
MOSES:	[...] Debido a la elección de la cámara y el equipo, estos **retratos** recordar el clásico fotos. [...] la pasión por el *cuadro* conducido Bauer a repetir la idea [...]
LEHRER+LCCO:	[...] Debido a la elección de la cámara y el equipo, estos **retratos** recordar el clásico fotos. [...] la pasión por el *retrato* conducido Bauer a repetir la idea [...]

Figure 5.4: Systems translation example with (in)consistent lexical choices.

and adequacy.

To conclude, we provide in Figure 5.4 a translation example from a news item about a photographer and his portraits work. MOSES has not translated consistently an occurrence of the word "portrait" (the one in italics) which wrongly becomes "cuadro" (painting) instead of the correct choice "retrato". Without LCCO, only the baseline, +SSLMbi, and both SSLC variants of LEHRER correctly produce "retrato" instead of "cuadro". With LCCO, on the contrary, all the system variants are able to produce the consistent translation.

5.5 Conclusions

Through this chapter we have presented two new document-level strategies that aid MT systems in producing more coherent translations by improving the lexical consistency of the translations during the decoding process. In particular, we have developed a new document-level feature function and a new change operation for a document-level decoder. The SSLC feature function scores the lexical selection consistency of a translation document. To this end, it uses word embeddings to measure the adequacy of word translations given their context, computed on words that have been translated in several different forms within a document. The change operation helps the decoder explore the translation search space by performing simultaneous lexical changes in a single translation step. Since it is able to modify several words at a time, even across sentences, it boosts the process of correcting the lexical inconsistencies.

Results show that, although differences among systems are not statistically significant for the automatic evaluation metrics, they are noticeable for human evaluators that prefer the outputs from the enhanced systems.

Chapter 6

Document-Aware Neural Machine Translation Decoding

NMT systems represent the current state-of-the-art for MT technologies. With regard to document-aware MT, there are several approaches that successfully enhance NMT systems to take into account document-level information, as already reviewed in Section 2.2.4. These systems usually propose modifications to the neural architecture and require the training data to be annotated with document-level information, such as the document boundaries. The main benefit of these approaches is that the neural translation models they obtain are better tuned and able to handle document-level information. However, their design makes it necessary to train the entire system every time a new type of document is to be translated, and also, the training data with the document-level annotations that they require is still scarce.

Through this chapter, we explore an alternative to introducing inter-sentence information in an NMT system without changing the neural translation model architecture. Furthermore, our approach neither needs a costly training process with scarce document-level tagged data. Roughly, we modify the beam search algorithm to allow the introduction of a Semantic Space Language Model (SSLM, recall Section 4.4) working in *shallow fusion* with a pre-trained NMT model. We analyze the impact of the associated parameters on the final translation quality. We obtain consistent and statistically significant improvements in terms of BLEU and METEOR and observe how the fused systems are able to handle synonyms to propose more adequate translations as well as help the system to disambiguate among several translation candidates for a word.

6.1 Fusion of an NMT System and an SSLM

In order to better explain how to fuse an NMT and an SSLM, firstly, we revisit the most used fusion techniques and, afterwards, we detail our particular approach for shallow fusion.

6.1.1 Deep, Shallow, Cold, and Simple Fusion

Fusion techniques have shown to be successful in several natural language tasks to merge information from two different neural models. In general, they combine information from two different models before producing the final output.

There are four main fusion techniques: deep, shallow, cold, and simple fusion. All of them extend the conditional probability learned by one model introducing the information from a second one, where the specific method that is used to combine both models is the main differentiator between the approaches. These techniques are motivated by how SMT integrates the information from different feature functions that represent different probabilistic models. In particular, recall from Section 2.1.1 that the posterior probability $p(y|x)$ maximized by an SMT system is decomposed using a log-linear model as the weighted sum of the different feature functions:

$$\log p(y|x) = \sum_i w_i f_i(x, y) + C$$

and that $p(y|x)$ can be expressed as follows after applying the Bayes' rule:

$$p(y|x) \propto p(x|y)p(y)$$

thus decomposing the translation probability as the combination of an inverse translation model, represented by $p(x|y)$, and a target language model, $p(y)$.

Deep fusion (Gülçehre et al., 2015, 2017) proposes a method to merge a translation model and a language model by introducing a gating mechanism that learns to balance the weight of the additional language model. In particular, Gülçehre et al. (2017) explain how to combine a pre-trained RNN language model with a pre-trained NMT system. They introduce a controller network with gating units that dynamically adjusts the weight for the RNN LM at each time step. They concatenate the hidden state from each neural model and introduce the weight outputted by the controller network to produce the final system output, estimating the next word in a sequence by computing (cf. Equation 2.1):

$$y_t = \text{softmax}\left(\text{DNN}\left(\left[\vec{H}_t^{TM}; g_t \vec{H}_t^{LM}\right], y_{t-1}, \vec{C}_t\right)\right) \tag{6.1}$$

where DNN represents any deep neural network, $[\vec{H}_t^{TM}; g_t \vec{H}_t^{LM}]$ is the concatenation of the hidden state $\vec{H}_t^{TM}$ from the translation model and the hidden state $\vec{H}_t^{LM}$ from the neural language model scaled by the weight g_t outputted from the gating mechanism at time t. As usual, the output is a softmax layer over the target vocabulary depending also on the previously generated target word y_{t-1} and the context vector

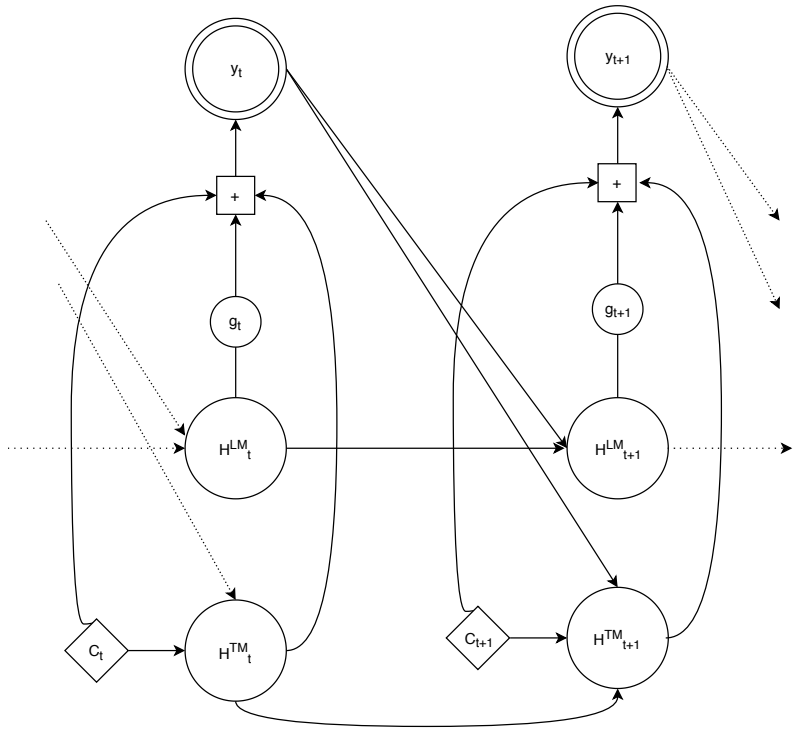

Figure 6.1: Sketch of the deep fusion approach. It merges the hidden representation of the NMT decoder and the neural language model before predicting the next translated word at each time step, with g_t controlling the contribution from the language model.

$\vec{C}_t$ from the NMT encoder. Finally, g_t takes as input the hidden state from the LM and is defined as follows:

$$g_t = \sigma\left(\vec{v}^\top \vec{H}_t^{LM} + b\right)$$

where σ is a logistic sigmoid function and $\vec{v}$ and b are learned parameters. The controller network is trained over a development set by only freezing the weights of the neural LM, allowing the decoder to use the NMT full signal and the signal from the NLM with an adjusted magnitude. The controller mechanism will learn the importance of each model to produce the next translated word. In summary, deep fusion allows for guiding the NMT model to produce more suitable translations regarding the development data and the language model domain information. Notice that in this approach the language model and the NMT model share the target vocabulary and also both models are trained independently. Figure 6.1 illustrates how deep fusion works.

Shallow fusion (Gülçehre et al., 2017) is a simpler approach that follows the same

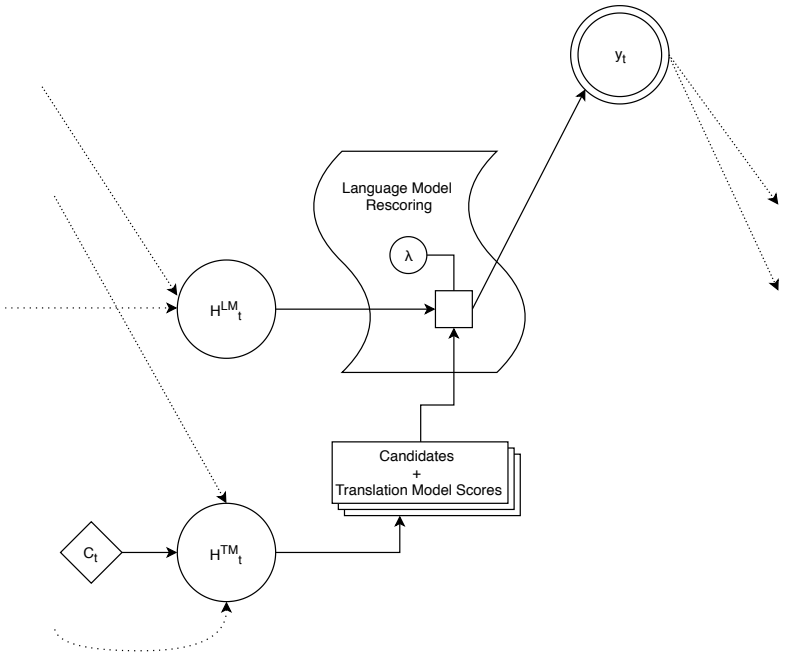

Figure 6.2: Sketch of the shallow fusion approach. It combines the scores from the NMT decoder and the neural language model probabilities before predicting the next translated word at each time step, with the parameter λ controlling the contribution from the language model.

idea as deep fusion but, in contrast, proposes the combination of the probabilities from the two models at inference time (see Figure 6.2). To this end, it changes the decoding objective function to integrate an LM prediction. Recall from Section 2.1.1 that the usual decoding objective function for an MT system can be written as:

$$\hat{y} = \arg\max_{y} \log p(y|x)$$

whereas the shallow fusion variation introduces the LM in a manner inspired by the SMT log-linear model:

$$\hat{y} = \arg\max_{y} \big(\log p(y|x) + \lambda \log p_{LM}(y) \big) \tag{6.2}$$

where p_{LM} is a language model trained on monolingual target data and λ is its weight. This formulation is a simpler version of the deep fusion since, instead of integrating a trainable neural controller mechanism, it combines the output probability distributions of the two models by using a parameter λ that is tuned by a grid search on a development set. The LM used by Gülçehre et al. (2017) is an LSTM-based RNN language model, but could be any model that generates as output a probability distribution on the discrete space of the target vocabulary shared with the translation

model. This fusion technique is the starting point we use to develop our combination of an SSLM within an NMT model.

Both deep and shallow fusion mechanisms use pre-trained LM and NMT models that were trained independently. This fact can hinder the system performance, but can also be seen as an advantage due to the flexibility it confers.

Cold fusion (Sriram et al., 2018) goes a step beyond the previous fusion techniques. It proposes to implement a deep fusion where the NMT model is trained from scratch including the LM as a fixed part of the network. This allows the NMT to better model the conditioning on the source sequence while the target language modeling is covered by the LM. These changes to deep fusion are reflected in the final formulation of the approach (cf. Equations 2.1 and 6.1):

$$y_t = \text{softmax}\left(\text{DNN}\left(\left[\vec{H}_t^{TM}; \vec{g}_t \circ \vec{h}_t^{LM}\right], y_{t-1}, \vec{C}_t\right)\right)$$

where $\vec{h}_t^{LM}$ is the result of processing the logit output of the language model by a deep neural network. Note that this parameter stands in place of the hidden state $\vec{H}_t^{LM}$ of the LM that is used in deep fusion. Also, the controller $\vec{g}_t$ is, on this occasion, a fine-grained gating mechanism (Yang et al., 2017) defined as:

$$\vec{g}_t = \sigma\left(W\left[\vec{H}_t^{TM}; \vec{h}_t^{LM}\right] + \vec{b}\right)$$

Note that, in contrast to deep fusion, the controller mechanism also depends on the hidden state $\vec{H}_t^{TM}$ from the translation model, and that σ is, in this case, applied element-wise. This formulation allows having a different gating value for each hidden node of the language model's state, resulting in greater flexibility for the fusion model to consider different aspects from the language model. Sriram et al. (2018) demonstrated the superior performance of cold fusion on a speech recognition task, but did not apply it to a translation task. The main advantage of cold fusion is to allow for building an adapted NMT model taking into account the target language modeling of a neural language model. However, it forces the training of an entirely new system when moving into a different domain modeled by a new different language model.

Simple fusion (Stahlberg et al., 2018) is the latest approach. It arises as an alternative simple method to use monolingual data for NMT training. Roughly, it integrates the shallow fusion technique in training time. This approach trains a translation model to predict the probability added to an LM prediction. Similarly to cold fusion, it trains an NMT model from scratch while combining the scores from the translation model and a pre-trained fixed LM. However, simple fusion does not integrate any controller mechanism as deep or cold fusion do. Formally, two variants of simple fusion are defined, with the PostNorm variant being:

$$y_t = \text{softmax}\left(\vec{S}_t^{TM}\right) \cdot p_{LM}(y_t)$$

and the PreNorm one being:

$$y_t = \text{softmax}\left(\vec{S}_t^{TM} + \log p_{LM}(y_t)\right)$$

where $\vec{s}_t^{TM}$ is the output of the translation model projection layer before softmax. Thus, PostNorm combines the $\vec{s}_t^{TM}$ transformed into a probability distribution via a softmax with the LM probabilities, whereas PreNorm applies normalization after combining $\vec{s}_t^{TM}$ with the logarithmic probabilities of the LM.

In contrast to deep or cold fusion, simple fusion benefits from not needing a gating network to balance the translation and language models. However, it proposes a more sophisticated model than the one of shallow fusion, since a translation model in simple fusion has to be trained with a fixed LM. The shallow fusion approach allows to use different LMs depending on the domain of the document to translate, without the need to change the base translation model or conduct a new training process.

6.1.2 Shallow Fusion of an NMT System and an SSLM

The extension of the NMT decoding process at document level we propose through this section benefits from the shallow fusion technique. In particular, it exploits the flexibility of being able to combine a general NMT model with a more domain specific language model to guide the NMT system towards a more adequate translation. In our approach, this other model is an SSLM used to introduce inter-sentence context information into the NMT decoding process. An additional advantage of shallow fusion is that it is one of the less time consuming fusion techniques in terms of training time, since it only needs to adjust the λ weight for the language model by a grid-search on development data, avoiding a long training on a large amount of data. Furthermore, this technique can be easily applied to any NMT model, either RNN-based or purely attention-based neural models.

Formally, we substitute the language model probability $p_{LM}(y)$ in the decoding function of shallow fusion by the SSLM associated probability (cf. Equation 6.2):

$$\hat{y} = \arg\max_{y} \big(\log p(y|x) + \lambda \log p_{SSLM}(y) \big)$$

where $p_{SSLM}(y)$ represents the probability that the SSLM model estimates for a generated sentence y. That probability is the product of the individual probabilities associated by SSLM to each of the words of y, which we compute as detailed in Equation 4.1 when $p_{similarity}$ is defined as a linear scale from the range $[-1, 1]$ to the range $[0, 1]$ (cf. Equation 5.1, where that scale has an additional ln).

Since SSLM requires the preceding context $\vec{c}_{y_t}$ of the next word y_t to be generated in order to estimate its probability, we need to modify the beam search of the NMT decoding that produces the translation of a sentence. We implement a cache mechanism to take into account the context information from the previously generated words, extending beyond sentence boundaries. In particular, the cache allows to add together the word embeddings from the previously generated words to obtain $\vec{c}_{y_t}$. However, the NMT model requires not only an estimate for a given target word, but a distribution probability over the entire target vocabulary space. Thus, it must be computed for each word y_i in the target vocabulary. Unfortunately, such an approach would have a high computational cost. Following the ranking/filtering approaches of Jean et al. (2015) and Wang et al. (2017b), we speed up this computation by filtering the words to score by the SSLM. In particular, it is only computed on the N

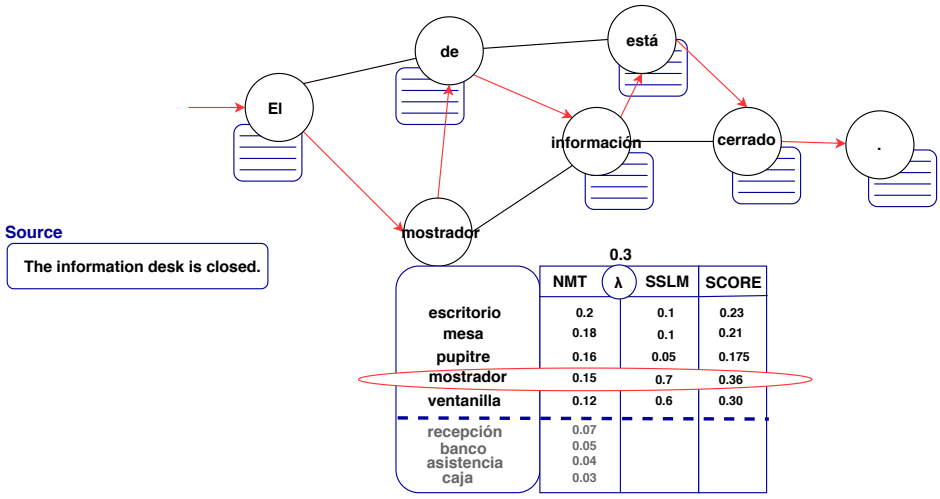

Figure 6.3: Sketch of the shallow fusion of an SSLM and an NMT inside the beam search algorithm. In this example, the process re-scores the $N = 5$ best candidates from the NMT model using the scores from the SSLM.

target words with the highest probabilities from the NMT model, that is, only the N best candidates from the NMT model are considered by the SSLM. Figure 6.3 depicts how the filtering process works in combination with the shallow fusion of the NMT and the SSLM models during the beam search. As a final remark, notice that although our system does not need any document-level annotation, it will understand any set of sentences in its input as a document.

6.2 Experiments

6.2.1 Settings

Our baseline NMT model follows the encoder-decoder architecture with attention of Bahdanau et al. (2015) and it is built using the OpenNMT-lua toolkit (Klein et al., 2017). We use a 4-layered bidirectional RNN encoder and a 4-layered RNN-based decoder with 800-dimensional hidden layers. Word embeddings are set to 500 dimensions for both source and target vocabularies. Stochastic gradient descent is used as optimizer algorithm for training, setting an initial learning rate of 1 and a learning decay of 0.7 after epoch 10 or if there is no loss improvement over the validation set. Training data is distributed on batches of 64 sentences and we use a 0.3 dropout probability between recurrent layers. Finally, a maximum sentence length of 50 tokens is used for both source and target sides and the vocabulary size is 50,000 for both target and source languages. The system is trained on the Europarl-v7 parallel corpus, analogously to the SMT baseline system from Section 4.3.1, using the newsCommentary2009 corpus as validation set. The system at epoch 20 is to be shallow fused with the SSLM.

We implement the shallow fusion of the SSLM and an NMT as an extension of the attentional encoder-decoder NMT baseline. The SSLM is the one we described and used in Section 4.4.1 with monolingual Spanish data. We use newsCommentary2011 as test set.

6.2.2 Analysis with Oracles

We implement three oracles to assess the potential impact of our techniques. The oracles behave as our fused approach, but leverage the reference translation to bias the decoding towards the word choices that are present in the reference. The goal of Oracle1 and Oracle2 is to assess the utility of the information enclosed in the Word Vector Model (WVM) used by the SSLM, i.e., to check whether the semantic information of SSLM can help in producing better translations. Oracle3 mimics our fused decoding approach and its goal is to evaluate the potential gain of using an SSLM in combination with an NMT. In other words, with Oracle3 we check how much the SSLM can help the NMT disambiguate between its best translation candidates, thus obtaining an upper bound for the improvements that can be achieved by shallow fusing an SSLM and an NMT system.

Oracle1 proceeds offline as follows: once a sentence has been translated, for each target word t it (i) uses the attention information to map that t to its corresponding source word s and, in turn, maps that s to its corresponding target word r found in the reference, and (ii) it replaces the target word t by r whenever $t \neq r$ and, furthermore, r is among the M words that are closest to t (with respect to cosine similarity) according to our WVM. Note that the use of attention in step (i) to map between target and source words is not as straightforward as the alignment information in an SMT system. In particular, we consider that a target word t and a source word s are one-to-one mapped, denoted $t \overset{1}{\leftrightarrow} s$, when the following holds: the attention from t to s is maximal among the attentions from that t to any source word s' and also among the attentions from any target word t' to that s, i.e., $t \overset{1}{\leftrightarrow} s$ if and only if $att(t, s) = \max\{att(t', s') : t' = t \vee s' = s\}$, where $att(.,.)$ denotes the attention value between two words. We use an analogous definition for the one-to-one mapping $s \overset{1}{\leftrightarrow} r$ between the source and reference words. Thus, for the target word t in consideration, step (i) tries to find the word r of the reference satisfying $t \overset{1}{\leftrightarrow} s \overset{1}{\leftrightarrow} r$, for some source word s. Table 6.1 and Figure 6.4 show the results for Oracle1. We observe that the WVM encodes semantically-valid candidates close together, as there is a noticeable improvement in the BLEU score even when considering just the $M = 5$ closest candidates. Also, the accuracy of the oracle's translations increases with the number M of considered closest words. This is expected since augmenting the number M also increases the coverage of the target vocabulary. In the limit, when M allows to encompass the whole $50K$-word vocabulary, Oracle1 simply rewrites the translation into the reference as far as the attention information allows, reaching an increase of +8.02 in BLEU score.

Oracle2 works as Oracle1 but proceeds online with the beam search. That is, when a hypothesis of the beam is to be extended with a new target word t, the oracle (i) analyzes the attention information to identify the actual word r used in the

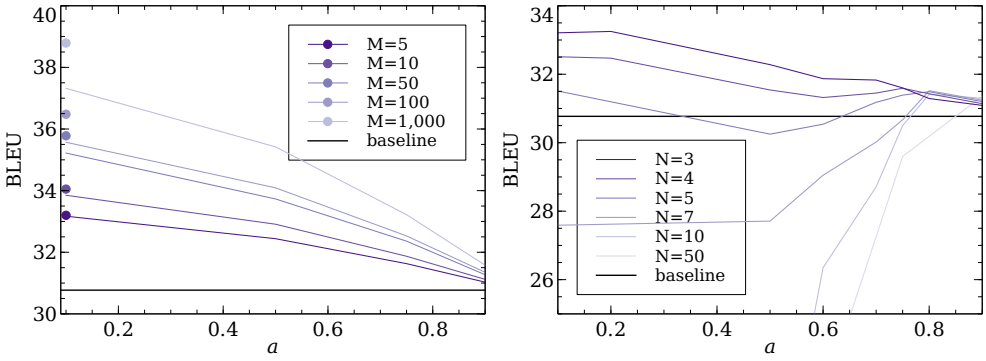

Figure 6.4: BLEU score of ORACLE1 (left, bullets), ORACLE2 (left, line plots), and ORACLE3 (right, line plots), as a function of the threshold a (ORACLE2 and ORACLE3) and for several values of the parameters M (ORACLE1 and ORACLE2) and N (ORACLE3). For ORACLE1 and ORACLE2, increasing the value of M beyond 1,000 does not affect the obtained scores noticeably.

System	**BLEU↑**	**METEOR↑**	N	M	a
baseline	30.77	49.86	-	-	-
ORACLE1	38.79	57.85	-	1,000	-
ORACLE2	37.32	54.35	-	1,000	0.1
ORACLE3	33.25	51.74	3	-	0.2

Table 6.1: Automatic evaluation of the oracle systems, together with the value used for their respective parameters.

reference to translate the source word s that t corresponds to and (*ii*) replaces t with r under the same circumstances as before (i.e., when $t \neq r$ and r appears in the list of M words closest to t according to our WVM). In this occasion, however, the attention information needed in step (*i*) to deduce the one-to-one mappings between the target and source is not fully available, as the target sentence is still being generated. For this reason, we need to add a minimal threshold a for the attention and refine our criterion as $t \overset{1,a}{\longleftrightarrow} s$ if and only if $t \overset{1}{\longleftrightarrow} s \wedge att(t,s) \geq a$. Thus, for the target word t in consideration, step (*i*) tries to find the word r of the reference satisfying $t \overset{1,a}{\longleftrightarrow} s \overset{1}{\longleftrightarrow} r$, for some source word s. Table 6.1 and Figure 6.4 present also the results for ORACLE2. The results are analogous to those of ORACLE1, but with lower scores. This difference of score between both oracles is almost negligible for the smallest values of M and a, but the distance widens as either M or a increases. This shows that our definition of $\overset{1,a}{\longleftrightarrow}$ is a proper approximation to obtain the mappings when not having the full attention information, as the permissive value $a = 0.1$ does not seem to be affected by noisy alignments for low values of M. This is because the oracle only replaces words by other semantically-close words (e.g., by synonyms), and thus, each of the substitutions preserves the meaning of the replaced word even if in some occasions the

computed alignment is not adequate. Conversely, by increasing M the oracle handles lists of candidates that are more semantically distant, and thus, in combination with the uncertainty of the alignments, the system introduces more errors.

ORACLE3 proceeds online with the beam search like ORACLE2, just differing on the criterion used to replace the target word t by the corresponding reference word r: the replacement is done when $t \neq r$ and, moreover, r appears among the N best candidates proposed by the NMT model. Note that this oracle does not use in any way the WVM underlying the SSLM: it simply assumes that such model will properly promote the correct word (i.e., the reference word) whenever it is present among the N top candidates of the NMT. Table 6.1 and Figure 6.4 present also the results for ORACLE3, which show that there is some margin for improvement for the fused system with respect to the NMT working in isolation. In contrast with ORACLE2, ORACLE3 produces more errors the more candidates that it considers, i.e., the greater the value of N is. Also, considering alignments with lower probabilities only helps when the value of N is small. In particular, considering more candidates by increasing N needs a stronger (i.e., higher) attention threshold a in order to filter out noisy substitutions. Nevertheless, in that more restrictive configuration of a, the results for the various values of N tend to converge.

In summary, ORACLE1 shows that the WVM of the SSLM properly clusters semantically-valid candidates close together, ORACLE2 that incomplete attention information does not hinder the oracle's ability to approximate the alignments, and ORACLE3 that there is a wide enough margin for improvement when fusing the systems.

6.2.3 Results

Our system has two main hyperparameters: the number N of NMT translation options that are used in the fusion, and the weight of the semantic language model λ. Table 6.2 and Figure 6.5 show the results of the automatic evaluation of the different variations of the presented fused system. The figure shows how the maximum quality is achieved around $\lambda = 0.15$, independently of the number N of re-scored candidates. All of our systems are able to improve the baseline for every value of N that we explored, achieving a statistically significant improvement of +0.23 in BLEU score and +0.31 in METEOR. Nevertheless, there is still room for further gains since, as seen in Table 6.1, ORACLE3 is able to increase +2.48 BLEU and +1.88 METEOR points.

We observe in Table 6.2 that the scores improve as long as we increase the value of N until it seems to stabilize for $N \geq 4$. Furthermore, comparing the outputs for $\lambda = 0.15$, the translations that the system produces with $N = 4$ only differ in 95 sentences with respect to those for $N = 5$ and in 107 for $N = 7$, while having 1,407 sentences out of 3,003 that differ with respect to the baseline. Also, the translations for $N = 5$ are almost exactly the same as with $N = 7$, differing only in 30 sentences, whereas the translations for $N = 7$ and $N = 10$ coincide. These facts support that the systems with $N \geq 4$ are converging towards an equivalent output. Looking into these differences, we realize that they manage different synonyms that may or not be in the reference. Like translating "*I have to*" as "*Tengo*" or "*Voy a tener*" which can be equivalent depending on the context.

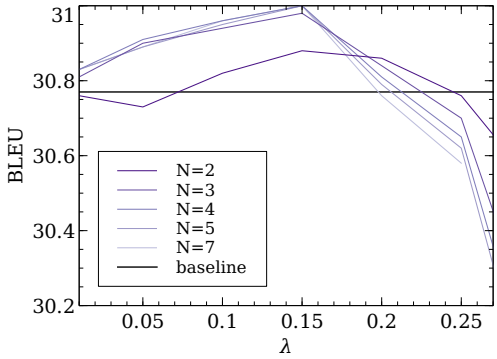

Figure 6.5: BLEU score of the fused system as a function of the weight λ, for several values of the parameter N.

N	**BLEU↑**	**METEOR↑**	**#unknown**
-	30.77	49.86	5901
2	30.88	50.17	4632
3	† 30.98	50.14	4501
4	† 31.00	50.15	4475
5	† 31.00	50.14	4459
7	† 31.00	50.14	4463
10	† 31.00	50.14	4463

Table 6.2: Automatic evaluation of the fused systems for varying values of the parameter N and with $\lambda = 0.15$, together with the amount of unknown words in their output. The first row corresponds to the baseline. † marks systems that are significantly different to the baseline with a p-value of 0.05, according to bootstrap resampling (Koehn, 2004).

We also observe that with larger values of N, the translations tend to be noisier or less adequate with respect to the source. For instance, "*Offices need a kindergarten nearby, architects have understood.*" is translated as:

"*las Oficinas necesitan una guardería cercana, los arquitectos han comprendido*" (N=4)

"*las oficinas de las oficinas de asistencia necesitan una guardería cercana.*" (N=7)

Notice in the second one the useless repetition of the translation for "*Offices*" and the appearance of the extra concept of assistance ("*asistencia*") that does not appear in the source sentence. Also, the information regarding the architects is missing in the second translation. Two important error types in NMT systems, word omission and new word creation, are exacerbated with large values for N.

Another example of more accurate translation occurs when translating "*According to Meteo France*". The best system using $N \geq 5$ translates this as "*Según Francia*" losing the reference to the meteorological company. In contrast, using $N = 4$, the

system is able to generate a more accurate translation "*Según Meteo Francia*". This analysis reflects the noise introduced by increasing the number of re-scored translation candidates by the system. In other words, it is important to have enough candidates to see more adequate translations, but there is a trade-off that the system needs to maintain between the number of new options and the noise introduced by these re-scored options.

Finally, we observe that the increase in the translation quality is also related to the decrease in the number of unknown words generated by the system. Since we use complete tokens without BPE (Sennrich et al., 2016) or SentencePiece (Kudo and Richardson, 2018) as translation units, several tokens are unknowns to the system. In general, the number of generated unknown words with the shallow fusion approach drops almost a 25% with respect to the unknown words generated by the baseline. For instance, the worst case-scenario sentence "*I'm rather a novice in Prague politics responded Lukas Kaucky.*" is translated by the baseline as:

"*Más bien soy un ⟨unk⟩ en la política de Praga, ⟨unk⟩ a Lucas ⟨unk⟩.*"

whereas our fused system is able to produce:

"*Más bien soy un* ***novato*** *en la política de Praga,* ***respondió*** *a Lucas ⟨unk⟩.*"

generating good translations for "*novice*" and "*responded*". These examples illustrate how fusing the SSLM with the NMT model helps the latter to disambiguate between the considered translation candidates for a word.

Finally, we pursue a little manual evaluation with 3 native-Spanish speakers with fluent English. We select a common subset of sentences from the test set translated by the baseline NMT and by the fused system with $N = 4$ and $\lambda = 0.15$. We randomly choose 100 sentences with at least 5 and at most 30 words with different translations. The annotators were asked for each of the 100 selected sentences to rank the output of both systems according to their general translation quality, allowing to rank them as tying. System outputs were presented in random order to avoid system identification. The annotators find 49% of the time that the translation from the fused system is better than the baseline, and they consider the quality of both translations to tie 19% of the time. They agreed 67% of the time, reaching a $\kappa = 0.4733$ (Fleiss, 1971) showing a "moderate" inter-annotator agreement (Landis and Koch, 1977). These results support that fused systems are able to improve the translations' quality.

6.3 Conclusions

We have presented a new approach that extends NMT decoding by introducing information from the preceding context on the target side. It fuses an attentional RNN with an SSLM by modifying the computation of the final score for an element of the target vocabulary inside the beam search algorithm. It is a flexible approach since it is compatible with any NMT architecture, and it allows to combine pre-trained models.

A preliminary, positive assessment of the potential improvement in the translation quality when introducing target context semantics has been conducted with oracles. The final validation of the implemented systems has resulted in improvements in the

BLEU and METEOR scores of up to +0.23 and +0.31, respectively, for English-to-Spanish translations. We have analyzed the impact of the different parameters of the system on these scores, observing that it is important to maintain a trade-off between the number of re-scored candidates, the SSLM weight, and the noise that will be introduced in the final translations. It is remarkable that our systems are able to propose valid translations where the baseline fails to choose one, making the number of unknown words drop while the translation quality increases. Also, a small manual evaluation has shown that humans tend to prefer the fused system outputs.

Chapter 7

Conclusions

Through this thesis, we explored several techniques to enhance MT systems by introducing global context information from a document.

We began by analyzing the most notable translation errors related to document-level phenomena and developed the set of post-processing strategies described in Chapter 3. In particular, these post-processes try to correct disagreements of gender and number and to improve lexical consistency over a preliminary full-document translation generated by an SMT system. These simple approaches showed the benefits of using document context information to promote correct translations according to the final users' criterion, even if not to the evaluation of the automatic metrics. A drawback of using a two-pass decoding strategy is that it limits the attainable performance. Furthermore, the post-processes require a set of additional NLP tools, like the ones included in FreeLing or the coreference resolutor RelaxCor, that are language-dependant and too resource-hungry to adapt to online decoding.

Our next step was to explore a set of methods to tackle this kind of document-level phenomena at translation time. We looked for a technique that would allow us to introduce the semantics from the context into the decoding process. Word embeddings emerged in the field as a solution to manage document semantics. Our analysis of these models, discussed in Chapter 4, proved that they can be helpful in the task of maintaining lexical choice consistency through the translation of a document by modeling the context of the words. This was especially so with the models having bilingual tokens. Additionally, we introduced the studied word embedding models inside the decoding process of the Lehrer document-oriented SMT decoder as the basis for Semantic Space Language Models. SSLMs capture the previous context for each word in a document and measure the deviation between the current word and its preceding context, promoting translations with consistent semantics. However, the integration of this information at decoding time did not show significant improvements according to the automatic evaluation metrics. We further analyzed these systems in Chapter 5, together with two new document-level strategies designed to aid MT systems to produce more coherent translations by improving their lexical consistency.

In particular, these two strategies consisted in a new document-level feature function and a new change operation integrated into Lehrer, working in tandem at decoding time to bias the exploration of the search space towards more consistent translations. To this end, they use word embeddings to measure the adequacy of word translations given their context. Once again, automatic evaluation metrics did not help us assess the impact of the implemented techniques. Nevertheless, a human evaluation showed a preference for the systems enhanced with the SSLM or the additional features over the baselines.

Our work concluded by applying the SSLM technique with word embeddings to the NMT framework as explained in Chapter 6. We followed a known fusion approach to perform this integration because fusing NMT models with neural LMs has been shown to be useful for several NLP tasks. The shallow fusion technique, in particular, represented a suitable method to combine the information from an NMT model and an SSLM. The combination required us to modify the usual beam search algorithm followed by NMT systems in order to keep track of the previously generated words, i.e., the context that has to be fed into an SSLM when re-scoring the best translation candidates proposed by the NMT model. Our approach presented a twofold novelty. On the one hand, it introduced context information from the target side, whereas most of the neural document-oriented approaches only take into account the source side context. On the other hand, it presented a modification for the NMT framework that was compatible with any architecture for the NMT system core. Since we only modified how the system combines probabilities when exploring the translation search space, we allowed using any neural architecture without the necessity of designing new models with a higher number of parameters to learn and adjust. We implemented three oracles that showed a promising potential improvement in the translation quality when introducing target context semantics into the NMT decoding process. Our approach achieves small improvements in the automatic metrics, but is nonetheless the preferred system according to the conducted human evaluation. When analyzing the impact of the different parameters of the system, we observed that it is important to maintain a trade-off between the number of re-scored candidates, the SSLM weight, and the noise that will be introduced in the final translations. Finally, it is remarkable that our systems are able to propose valid translations where the baseline fails to choose one, making the number of unknown words drop while the translation quality increases.

7.1 Future Work

There is still margin of improvement to reach document-aware MT systems that perform well, both in terms of computing performance and the quality of their outputs. Several new research paths arise from the work we explored through this thesis.

For instance, we are interested in exploring other decoding strategies that can take as input full documents and that try to optimize an objective function evaluated on whole documents. The ACO-based proposal we presented only obtains average results at sentence level, but further variants of the ACO metaheuristic are explored in the literature and their applicability to document-level MT might be more suitable and

should be studied. Furthermore, other standard metaheuristics beyond the already tried simulated annealing (Hardmeier, 2014), genetic algorithms (Douib et al., 2016), and ACO could also be considered for guiding the document-aware decoding process.

A recurrent issue we observed throughout all the conducted experiments is the limited capability of the automatic metrics to capture and measure the translation improvements at document level. Although we are aware of some existing evaluation metrics that take into account some discourse information, like the one by Giménez et al. (2010) or the MEANT family (Lo, 2017; Lo and Wu, 2011; Lo et al., 2014), we think it would be desirable to study how to design an effective automatic metric able to assess the translation quality by taking into account inter-sentence context information. One idea to explore in this line would be to design an evaluation metric that takes into account the semantics of the inter-sentence context captured by document embeddings, learnt directly from document-annotated data or computed from pre-trained word embeddings. Then, such semantic information could be combined with lexical information like n-gram matching. In order to weight the contribution of each part, it might be possible to train a regressor or a small neural network on the annotations from manual evaluation tasks, and in this way learn the relevance that humans assign to each part when assessing the translation quality. The benefit of such a metric would be twofold. On the one hand, it would aid the development of document-level MT systems by facilitating the assessment of their outputs. On the other hand, an additional benefit of tackling the scoring of translations at document-level is that automatic parameter optimization for document-aware decoders would be improved. Thus far, there are only limited results for tuning DOCENT through MERT or PRO (Smith, 2015; Stymne et al., 2013) and, in those cases, the optimization is done against BLEU. Such metric is based on lexical similarity against a set of references and does not directly capture any document-level phenomenon. Therefore, the outputted weights optimized in this way need not properly correlate with such phenomena. It would also be interesting to study whether alternative automatic optimization approaches such as SPSA (Simultaneous Perturbation Stochastic Approximation) (Lambert and Banchs, 2006; Spall, 1992) can successfully tune DOCENT, and how sensitive they are to the presence of the different discourse-level features.

Furthermore, we left as future work a thorougher evaluation of the used word embeddings. We look forward to explore the effect of using other kinds of word embeddings like the ones presented by Madhyastha et al. (2014) or the newer BERT (Devlin et al., 2019) and ELMO (Peters et al., 2018) embeddings that use the notion of intra-sentence context to learn the distributed representations of words, and compare their performance with the vector models presented in this dissertation.

Regarding document-level NMT, future work will include but not be limited to the following research suggestions:

- In order to better attain the improvements reachable by our oracles, we want to analyze the validity of the cosine similarity as a measure and use other alternatives such as CSLS (Cross-domain Similarity Local Scaling) (Lample et al., 2018), or other margin-based scores instead (Artetxe and Schwenk, 2019).

- Study the benefits of introducing context information into different NMT architectures. We find necessary to study the enhancement of recurrent- or

Transformer-based NMT systems to observe how document-level information affects the performance of each type of architecture. This experimentation will shed light on the best approach to enhance an NMT model with context information. In particular, we want to study how the inter-sentence information can affect the quality of attention-based translation systems and also to use BPEed input to compare the positive effect on unknown words that we observed. These two studies will improve the quality of the systems as a whole (both baseline and fused).

- Assess the benefits of using document-level information to perform a better domain adaptation of the translation models. In this line, we propose to combine general-purpose NMT systems with SSLMs trained on data from different domains.

- Design new neural architectures able to model document context beyond the k previously seen sentences. Following the works of Wang et al. (2017b) and Voita et al. (2018), we propose as first step to study how to extend their work to model a wider context.

- Study the different effects of introducing wider context information in the source side (encoder), target side (decoder), or both, and state the benefits of each one in order to understand the effect of each kind of context into the final translation quality. In our approach we explore the use of inter-sentence information only on the target side. We would like to carry out a comparative among systems that handle context at different points of the model and from source and target sides in isolation or simultaneously.

- Extend the decoding algorithm used by NMT systems to change the exploration of the translation search space from sentence to document level. The approach we exposed in Chapter 6 proposes a particular extension of the beam search algorithm used to build the translation output. However, we are interested in studying new search algorithms able to explore a document translation search space compatible with the NMT decoding framework.

These research ideas appear as a natural continuation for the work we developed in this thesis, both on SMT and NMT systems. This work gave a mature starting point to design new MT approaches able to handle document-level information regardless of the system core technology.

As final remark, it will be interesting to explore the applicability of the presented techniques into other NLP tasks by redefining the concept of context, for instance, inside a chat bot or in an automatic generator of reviews.

Bibliography

Mikel Artetxe and Holger Schwenk. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Conference of the Association for Computational Linguistics (ACL) – Volume 1*, pages 3197–3203, 2019.
Cited on page 85.

Abhishek Arun, Barry Haddow, Philipp Koehn, Adam Lopez, Chris Dyer, and Phil Blunsom. Monte Carlo techniques for phrase-based translation. *Machine Translation*, 24(2):103–121, 2010.
Cited on page 21.

Duygu Ataman, Matteo Negri, Marco Turchi, and Marcello Federico. Linguistically motivated vocabulary reduction for neural machine translation from Turkish to English. *CoRR*, abs/1707.09879, 2017. URL `http://arxiv.org/abs/1707.09879`.
Cited on page 20.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2015.
Cited on pages 8, 9, 17, 18, and 75.

Rafael E. Banchs, Luis Fernando D'Haro, and Haizhou Li. Adequacy-fluency metrics: Evaluating MT in the continuous space model framework. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 23(3):472–482, 2015.
Cited on page 28.

Satanjeev Banerjee and Alon Lavie. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization (IEEvaluation@ACL)*, pages 65–72, 2005.
Cited on page 29.

Rachel Bawden, Rico Sennrich, Alexandra Birch, and Barry Haddow. Evaluating discourse phenomena in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) – Volume 1*, pages 1304–1313,

2018.
Cited on page 26.

Jerome R. Bellegarda. Exploiting latent semantic information in statistical language modeling. *Proceedings of the IEEE*, 88(8):1279–1296, 2000.
Cited on pages 50 and 51.

Yoshua Bengio, Nicolas Boulanger-Lewandowski, and Razvan Pascanu. Advances in optimizing recurrent networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8624–8628, 2013.
Cited on page 18.

Winfield S. Bennett and Jonathan Slocum. The LRC machine translation system. *Computational Linguistics*, 11(2-3):111–121, 1985.
Cited on page 8.

Christian Blum and Marco Dorigo. The hyper-cube framework for ant colony optimization. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 34(2): 1161–1172, 2004.
Cited on page 105.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, 1990.
Cited on pages 8 and 14.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. Re-evaluating the role of BLEU in machine translation research. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 249–256, 2006.
Cited on page 27.

Borja Calvo and Guzmán Santafé. scmamp: Statistical comparison of multiple algorithms in multiple problems. *The R Journal*, 8(1):248–256, 2016.
Cited on page 117.

Marine Carpuat. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW)*, pages 19–27, 2009.
Cited on page 21.

María Asunción Castaño and Francisco Casacuberta. A connectionist approach to machine translation. In *Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 160–167, 1997.
Cited on page 17.

David Chiang, Yuval Marton, and Philip Resnik. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 224–233, 2008.
Cited on page 15.

Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. In *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST@EMNLP)*, pages 103–111, 2014a.
Cited on pages 17 and 18.

Kyunghyun Cho, Bart van Merrienboer, Çağlar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, 2014b.
Cited on page 17.

Alberto Colorni, Marco Dorigo, and Vittorio Maniezzo. Distributed optimization by ant colonies. In *Towards a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life (ECAL)*, pages 134–142, 1992.
Cited on page 24.

Ryan Cotterell, Thomas Müller, Alexander M. Fraser, and Hinrich Schütze. Labeled morphological segmentation with semi-markov models. In *Proceedings of the 19th Conference on Computational Natural Language Learning (CoNLL)*, pages 164–174, 2015.
Cited on page 20.

Deborah Coughlin. Correlating automated and human assessments of machine translation quality. In *Proceedings of the Machine Translation Summit IX*, pages 63–70, 2003.
Cited on page 27.

Koby Crammer, Ofer Dekel, Joseph Keshet, Shai Shalev-Shwartz, and Yoram Singer. Online passive-aggressive algorithms. *Journal of Machine Learning Research*, 7: 551–585, 2006.
Cited on page 15.

Christopher Culy and Susanne Z. Riehemann. The limits of n-gram translation evaluation metrics. In *Proceedings of the Machine Translation Summit IX*, pages 71–78, 2003.
Cited on page 27.

Michael Denkowski and Alon Lavie. Extending the METEOR machine translation evaluation metric to the phrase level. In *Proceedings of Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 250–253, 2010.
Cited on page 29.

Michael Denkowski and Alon Lavie. METEOR-NEXT and the METEOR paraphrase tables: Improved evaluation support for five target languages. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR (WMT@ACL)*, pages 339–342, 2012a.
Cited on page 28.

Michael Denkowski and Alon Lavie. Challenges in predicting machine translation utility for human post-editors. In *Proceedings of the 10th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, 2012b.
Cited on page 27.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL) – Volume 1*, pages 1370–1380, 2014.
Cited on page 17.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4171–4186, 2019.
Cited on pages 45 and 85.

Luis Fernando D'Haro, Rafael E. Banchs, Chiori Hori, and Haizhou Li. Automatic evaluation of end-to-end dialog systems with adequacy-fluency metrics. *Computer Speech & Language*, 55:200–215, 2019.
Cited on page 28.

Teun Adrianus van Dijk. *Text and context: Explorations in the semantics and pragmatics of discourse*. Number 21 in Longman Linguistics Library. Longman, 1977.
Cited on page 8.

Shuoyang Ding, Kevin Duh, Huda Khayrallah, Philipp Koehn, and Matt Post. The JHU machine translation systems for WMT 2016. In *Proceedings of the First Conference on Machine Translation (WMT)*, pages 272–280, 2016.
Cited on page 20.

George Doddington. Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In *Proceedings of the 2nd International Conference on Human Language Technology Research (HLT)*, pages 138–145, 2002.
Cited on pages 27 and 29.

Marco Dorigo and Christian Blum. Ant colony optimization theory: A survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005.
Cited on page 25.

Marco Dorigo, Vittorio Maniezzo, and Alberto Colorni. Ant system: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics – Part B*, 26(1):29–41, 1996.
Cited on page 24.

Ameur Douib, David Langlois, and Kamel Smaïli. Genetic-based decoder for statistical machine translation. In *Proceedings of the 17th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing) – Part II*,

pages 101–114, 2016.
Cited on pages 22 and 85.

Cristina España-Bonet, Dana Ruiter, and Josef van Genabith. UdS-DFKI participation at WMT 2019: Low-resource (*en–gu*) and coreference-aware (*en–de*) systems. In *Proceedings of the Fourth Conference on Machine Translation (WMT) – Volume 2: Shared Task Papers*, pages 382–389, 2019.
Cited on page 27.

Thierry Etchegoyhen, Eva Martínez Garcia, Andoni Azpeitia, Gorka Labaka, Iñaki Alegria, Itziar Cortes Etxabe, Amaia Jauregi Carrera, Igor Ellakuria Santos, Maite Martin, and Eusebi Calonge. Neural machine translation of Basque. In *Proceedings of the 21st Annual Conference of the European Association for Machine Translation (EAMT)*, pages 139–148, 2018.
Cited on page 20.

Joseph L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.
Cited on pages 66 and 80.

Peter W. Foltz, Walter Kintsch, and Thomas K. Landauer. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25(2–3):285–307, 1998.
Cited on page 50.

Mikel L. Forcada and Ramón P. Ñeco. Recursive hetero-associative memories for translation. In *Biological and Artificial Computation: From Neuroscience to Technology, Proceedings of the 4th International Work-Conference on Artificial and Natural Neural Networks (IWANN)*, pages 453–462, 1997.
Cited on page 17.

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.
Cited on page 117.

Jesús Giménez and Lluís Màrquez. Heterogeneous automatic MT evaluation through non-parametric metric combinations. In *Proceedings of the Third International Joint Conference on Natural Language Processing (IJCNLP)*, pages 319–326, 2008.
Cited on page 30.

Jesús Giménez and Lluís Màrquez. Asiya: An open toolkit for automatic machine translation (meta-)evaluation. *The Prague Bulletin of Mathematical Linguistics*, 94:77–86, 2010.
Cited on pages 28 and 37.

Jesús Giménez, Lluís Màrquez, Elisabet Comelles, Irene Castellón, and Victoria Arranz. Document-level automatic MT evaluation based on discourse representations. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and*

MetricsMATR (WMT@ACL), pages 333–338, 2010.
Cited on page 85.

Zhengxian Gong, Min Zhang, and Guodong Zhou. Cache-based document-level statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 909–919, 2011.
Cited on page 21.

Meritxell Gonzàlez, Jesús Giménez, and Lluís Màrquez. A graphical interface for MT evaluation and error analysis. In *Proceedings of the 50th Association for Computational Linguistics (ACL), System Demonstrations*, pages 139–144, 2012.
Cited on pages 28, 37, 54, and 64.

Liane Guillou. Improving pronoun translation for statistical machine translation. In *Proceedings of the Student Research Workshop at the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 1–10, 2012.
Cited on page 21.

Çağlar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535, 2015. URL `http://arxiv.org/abs/1503.03535`.
Cited on page 70.

Çağlar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. On integrating a language model into neural machine translation. *Computer Speech & Language*, 45:137–148, 2017.
Cited on pages 70, 71, and 72.

Michael Alexander Kirkwood Halliday and Ruqaiya Hasan. *Cohesion in English*. Number 9 in English Language Series. Longman, 1976.
Cited on page 8.

Christian Hardmeier. *Discourse in Statistical Machine Translation*. PhD thesis, Uppsala Universitet, 2014.
Cited on pages 24, 52, 63, and 85.

Christian Hardmeier and Marcello Federico. Modelling pronominal anaphora in statistical machine translation. In *Proceedings of the 7th International Workshop on Spoken Language Translation (IWSLT)*, pages 283–289, 2010.
Cited on page 21.

Christian Hardmeier, Joakim Nivre, and Jörg Tiedemann. Document-wide decoding for phrase-based statistical machine translation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 1179–1190, 2012.
Cited on pages 9, 22, 24, 25, 50, 51, 57, 108, and 112.

Christian Hardmeier, Sara Stymne, Jörg Tiedemann, and Joakim Nivre. Docent: A document-level decoder for phrase-based statistical machine translation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*, pages 193–198, 2013.
Cited on pages 22 and 114.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. Achieving human parity on automatic Chinese to English news translation. *CoRR*, abs/1803.05567, 2018. URL `http://arxiv.org/abs/1803.05567`.
Cited on pages 7 and 9.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.
Cited on page 18.

John H. Holland. Genetic algorithms and the optimal allocation of trials. *Society for Industrial and Applied Mathematics Journal on Computing (SICOMP)*, 2(2): 88–105, 1973.
Cited on page 22.

Mark Hopkins and Jonathan May. Tuning as ranking. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1352–1362, 2011.
Cited on page 15.

Paul Jaccard. The distribution of the flora in the alpine zone. *New Phytologist*, 11 (2):37–50, 1912.
Cited on page 29.

Sébastien Jean and Kyunghyun Cho. Context-aware learning for neural machine translation. *CoRR*, abs/1903.04715, 2019. URL `http://arxiv.org/abs/1903.04715`.
Cited on page 26.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation (WMT)*, pages 134–140, 2015.
Cited on page 74.

Sébastien Jean, Stanislas Lauly, Orhan Firat, and Kyunghyun Cho. Does neural machine translation benefit from larger context? *CoRR*, abs/1704.05135, 2017. URL `http://arxiv.org/abs/1704.05135`.
Cited on page 26.

Marcin Junczys-Dowmunt. Microsoft translator at WMT 2019: Towards large-scale document-level neural machine translation. In *Proceedings of the Fourth Conference on Machine Translation (WMT) – Volume 2: Shared Task Papers*, pages 424–432, 2019.
Cited on page 27.

David Kauchak and Regina Barzilay. Paraphrasing for automatic evaluation. In *Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NAACL)*, pages 455–462, 2006.
Cited on page 28.

Huda Khayrallah and Philipp Koehn. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation (NMT@ACL)*, pages 74–83, 2018.
Cited on page 20.

Scott Kirkpatrick, Charles Daniel Gelatt Jr., and Mario P. Vecchi. Optimization by simmulated annealing. *Science*, 220(4598):671–680, 1983.
Cited on page 24.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics: System Demonstrations (ACL)*, pages 67–72, 2017.
Cited on page 75.

Kevin Knight. Decoding complexity in word-replacement translation models. *Computational Linguistics*, 25(4):607–615, 1999.
Cited on page 24.

Philipp Koehn. Statistical significance tests for machine translation evaluation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 388–395, 2004.
Cited on pages 54, 64, 79, and 115.

Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of the Machine Translation Summit X*, pages 79–86, 2005.
Cited on pages 37 and 49.

Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation (NMT@ACL)*, pages 28–39, 2017.
Cited on page 20.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology (HLT-NAACL) – Volume 1*, pages 48–54, 2003.
Cited on pages 8, 15, 21, 22, 24, 32, 51, 105, and 111.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics on Interactive Poster and Demonstration Sessions (ACL)*, pages 177–180, 2007.
Cited on pages 9, 21, 37, 49, and 105.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 66–71, 2018.
Cited on page 80.

Patrik Lambert and Rafael E. Banchs. Tuning machine translation parameters with SPSA. In *Proceedings of the 2006 International Workshop on Spoken Language Translation (IWSLT)*, pages 190–196, 2006.
Cited on page 85.

Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou. Word translation without parallel data. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2018.
Cited on page 85.

John Richard Landis and Gary Grove Koch. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174, 1977.
Cited on pages 66 and 80.

Philippe Langlais, Alexandre Patry, and Fabrizio Gotti. A greedy decoder for phrase-based statistical machine translation. In *Proceedings of the 11th International Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 104–113, 2007.
Cited on page 22.

Samuel Läubli, Rico Sennrich, and Martin Volk. Has machine translation achieved human parity? A case for document-level evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4791–4796, 2018.
Cited on page 26.

Alon Lavie and Abhaya Agarwal. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In *Proceedings of the Second Workshop on Statistical Machine Translation (StatMT@ACL)*, pages 228–231, 2007.
Cited on page 27.

Ronan Le Nagard and Philipp Koehn. Aiding pronoun translation with co-reference resolution. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR (WMT@ACL)*, pages 252–261, 2010.
Cited on page 21.

Vladimir Iosifovich Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics-Doklady*, 10(8):707–710, 1966.
Cited on page 28.

Chin-Yew Lin and Franz Josef Och. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, 2004a.
Cited on page 29.

Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (ACL)*, 2004b.
Cited on page 29.

Ding Liu and Daniel Gildea. Syntactic features for evaluation of machine translation. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization (IEEvaluation@ACL)*, pages 25–32, 2005.
Cited on page 30.

Chi-kiu Lo. MEANT 2.0: Accurate semantic MT evaluation for any output language. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 589–597, 2017.
Cited on pages 28 and 85.

Chi-kiu Lo and Dekai Wu. MEANT: An inexpensive, high-accuracy, semi-automatic metric for evaluating translation utility based on semantic roles. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (ACL-HLT)*, pages 220–229, 2011.
Cited on pages 28 and 85.

Chi-kiu Lo, Meriem Beloucif, Markus Saers, and Dekai Wu. XMEANT: Better semantic MT evaluation without reference translations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL) – Volume 2: Short Papers*, pages 765–771, 2014.
Cited on pages 28 and 85.

Annie Louis and Bonnie Webber. Structured and unstructured cache models for SMT domain adaptation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 155–163, 2014.
Cited on page 21.

Pranava Swaroop Madhyastha, Xavier Carreras, and Ariadna Quattoni. Learning task-specific bilexical embeddings. In *Proceedings of the 25th International Conference on Computational Linguistics (COLING)*, pages 161–171, 2014.
Cited on page 85.

Eva Martínez Garcia, Cristina España-Bonet, and Lluís Màrquez. Document-level machine translation as a re-translation process. *Procesamiento del Lenguaje Natural (SEPLN)*, 53:103–110, 2014a.
Cited on page 10.

Eva Martínez Garcia, Cristina España-Bonet, and Lluís Màrquez. Experiments on document level machine translation. Technical Report LSI-14-11-R, Universitat Politècnica de Catalunya, Spain, 2014b. URL `http://hdl.handle.net/2117/26965`.
Cited on page 10.

Eva Martínez Garcia, Cristina España-Bonet, Jörg Tiedemann, and Lluís Màrquez. Word's vector representations meet machine translation. In *Proceedings of the 8th Workshop on Syntax, Semantics and Structure in Statistical Translation (SSST@EMNLP)*, pages 132–134, 2014c.
Cited on page 10.

Eva Martínez Garcia, Cristina España-Bonet, and Lluís Màrquez. Document-level machine translation with word vector models. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation (EAMT)*, pages 59–66, 2015.
Cited on page 10.

Eva Martínez Garcia, Carles Creus, Cristina España-Bonet, and Lluís Màrquez. Using word embeddings to enforce document-level lexical consistency in machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108:85–96, 2017.
Cited on page 11.

Sameen Maruf and Gholamreza Haffari. Document context neural machine translation with memory networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) – Volume 1*, pages 1275–1284, 2018.
Cited on page 26.

Ilya Dan Melamed, Ryan Green, and Joseph P. Turian. Precision and recall of machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 61–63, 2003.
Cited on page 29.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013a. URL `http://arxiv.org/abs/1301.3781`.
Cited on pages 17, 45, and 46.

Tomas Mikolov, Quoc Viet Le, and Ilya Sutskever. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168, 2013b. URL `http://arxiv.org/abs/1309.4168`.
Cited on pages 45 and 50.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3111–3119, 2013c.
Cited on page 17.

Sonja Nießen, Franz Josef Och, Gregor Leusch, and Hermann Ney. An evaluation tool for machine translation: Fast evaluation for MT research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC)*, 2000.
Cited on page 28.

Franz Josef Och. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics (ACL) – Volume 1*, pages 160–167, 2003.
Cited on pages 15, 37, 49, 51, and 114.

Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51, 2003.
Cited on pages 37 and 49.

Franz Josef Och, Nicola Ueffing, and Hermann Ney. An efficient A* search algorithm for statistical machine translation. In *Proceedings of the Workshop on Data-driven Methods in Machine Translation (DMMT) – Volume 14*, pages 1–8, 2001.
Cited on page 16.

Karolina Owczarzak, Declan Groves, Josef van Genabith, and Andy Way. Contextual bitext-derived paraphrases in automatic MT evaluation. In *Proceedings of the Workshop on Statistical Machine Translation (StatMT@NAACL)*, pages 86–93, 2006.
Cited on page 28.

Lluís Padró, Samuel Reese, Eneko Agirre, and Aitor Soroa. Semantic services in FreeLing 2.1: WordNet and UKB. In *Principles, Construction, and Application of Multilingual Wordnets*, pages 99–105, 2010.
Cited on pages 32 and 35.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, pages 311–318, 2002.
Cited on pages 29, 37, 49, 51, and 115.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT) – Volume 1*, pages 2227–2237, 2018.
Cited on page 85.

John R. Pierce and John B. Carroll. *Language and Machines: Computers in Translation and Linguistics*. National Academy of Sciences/National Research Council, Washington, DC, USA, 1966.
Cited on page 13.

Martin Popel and Ondřej Bojar. Training tips for the Transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110:43–70, 2018.
Cited on page 19.

Martin Popel, Dominik Macháček, Michal Auersperger, Ondřej Bojar, and Pavel Pecina. English-Czech systems in WMT19: Document-level transformer. In *Proceedings of the Fourth Conference on Machine Translation (WMT) – Volume 2: Shared Task Papers*, pages 541–547, 2019.
Cited on page 27.

Andrei Popescu-Belis. Context in neural machine translation: A review of models and evaluations. *CoRR*, abs/1901.09115, 2019. URL `http://arxiv.org/abs/1901.09115`.
Cited on page 26.

Cornelis Joost van Rijsbergen. *Information Retrieval*. Butterworth-Heinemann, 2nd edition, 1979.
Cited on page 29.

Grazia Russo-Lassner, Jimmy Lin, and Philip Resnik. A paraphrase-based approach to machine translation evaluation. Technical Report LAMP-TR-125/CS-TR-4754/UMIACS-TR-2005-57, University of Maryland, College Park, 2005.
Cited on page 28.

Theodorus Johannes Maria Sanders and Henk L. W. Pander Maat. Cohesion and coherence: Linguistic approaches. In *Encyclopedia of Language & Linguistics*, pages 591–595. Elsevier, 2nd edition, 2006.
Cited on page 8.

Emili Sapena, Lluís Padró, and Jordi Turmo. A global relaxation labeling approach to coreference resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters (COLING)*, pages 1086–1094, 2010.
Cited on page 32.

Holger Schwenk, Daniel Déchelotte, and Jean-Luc Gauvain. Continuous space language models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL)*, pages 723–730, 2006.
Cited on page 17.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL) – Volume 1*, pages 1715–1725,

2016.
Cited on pages 19 and 80.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.
Cited on page 15.

Aaron Smith. BLEU decoding and feature weight tuning in Docent. Master's thesis, Uppsala Universitet, 2015.
Cited on pages 52 and 85.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Biennial Conference of the Association for Machine Translation in the Americas (AMTA)*, pages 223–231, 2006.
Cited on page 29.

Matthew Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. Fluency, adequacy, or HTER?: Exploring different human judgments with a tunable MT metric. In *Proceedings of the 4th Workshop on Statistical Machine Translation (WMT@EACL)*, pages 259–268, 2009a.
Cited on page 29.

Matthew G. Snover, Nitin Madnani, Bonnie J. Dorr, and Richard Schwartz. TER-Plus: paraphrase, semantic, and alignment enhancements to Translation Edit Rate. *Machine Translation*, 23(2):117–127, 2009b.
Cited on page 28.

James C. Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3): 332–341, 1992.
Cited on page 85.

Lucia Specia, Kashif Shah, José Guilherme Camargo de Souza, and Trevor Cohn. QuEst – A translation quality estimation framework. In *Proceedings of the 51th Conference of the Association for Computational Linguistics (ACL), Demo Session*, pages 79–84, 2013.
Cited on page 49.

Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates. Cold fusion: Training Seq2Seq models together with language models. In *Proceedings of the 19th Annual Conference of the International Speech Communication Association (Interspeech)*, pages 387–391, 2018.
Cited on page 73.

Felix Stahlberg, James Cross, and Veselin Stoyanov. Simple fusion: Return of the language model. In *Proceedings of the Third Conference on Machine Translation: Research Papers (WMT)*, pages 204–211, 2018.
Cited on page 73.

Felix Stahlberg, Danielle Saunders, Adrià de Gispert, and Bill Byrne. CUED@WMT19:EWC&LMs. In *Proceedings of the Fourth Conference on Machine Translation (WMT) – Volume 2: Shared Task Papers*, pages 563–572, 2019.
Cited on page 27.

Andreas Stolcke. SRILM – An extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP)*, pages 901–904, 2002.
Cited on pages 37 and 49.

Thomas Stützle and Holger H. Hoos. $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System. *Future Generation Computer Systems*, 16(8):889–914, 2000.
Cited on page 105.

Sara Stymne, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. Feature weight optimization for discourse-level SMT. In *Proceedings of the Workshop on Discourse in Machine Translation (DiscoMT@ACL)*, pages 60–69, 2013.
Cited on pages 52 and 85.

Mihai Surdeanu, Jordi Turmo, and Eli Comelles. Named entity recognition from spontaneous open-domain speech. In *Proceedings of the 9th European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 3433–3436, 2005.
Cited on page 32.

Ilya Sutskever, Oriol Vinyals, and Quoc Viet Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014 (NIPS)*, pages 3104–3112, 2014.
Cited on page 17.

Aarne Talman, Umut Sulubacak, Raúl Vázquez, Yves Scherrer, Sami Virpioja, Alessandro Raganato, Arvi Hurskainen, and Jörg Tiedemann. The University of Helsinki submissions to the WMT19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (WMT) – Volume 2: Shared Task Papers*, pages 611–622, 2019.
Cited on page 27.

Jörg Tiedemann. News from OPUS – A collection of multilingual parallel corpora with tools and interfaces. *Recent Advances in Natural Language Processing (RANLP)*, 5:237–248, 2009.
Cited on page 46.

Jörg Tiedemann. To cache or not to cache? Experiments with adaptive models in statistical machine translation. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR (WMT@ACL)*, pages 189–194, 2010.
Cited on page 21.

Jörg Tiedemann. Parallel data, tools and interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC)*, pages 2214–2218, 2012.
Cited on page 46.

Jörg Tiedemann and Yves Scherrer. Neural machine translation with extended context. In *Proceedings of the Third Workshop on Discourse in Machine Translation (DiscoMT@EMNLP)*, pages 82–92, 2017.
Cited on page 26.

Christoph Tillmann, Stephan Vogel, Hermann Ney, Alex Zubiaga, and Hassan Sawaf. Accelerated DP based search for statistical translation. In *Proceedings of the Fifth European Conference on Speech Communication and Technology (EUROSPEECH)*, pages 2667–2670, 1997.
Cited on page 28.

Antonio Toral, Sheila Castilho, Ke Hu, and Andy Way. Attaining the unattainable? Reassessing claims of human parity in neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers (WMT)*, pages 113–123, 2018.
Cited on page 9.

Zhaopeng Tu, Yang Liu, Shuming Shi, and Tong Zhang. Learning to remember translation history with a continuous cache. *Transactions of the Association for Computational Linguistics (TACL)*, 6:407–420, 2018.
Cited on page 26.

Ferhan Ture, Douglas W. Oard, and Philip Resnik. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*, pages 417–426, 2012.
Cited on page 21.

Joseph P. Turian, Luke Shen, and Ilya Dan Melamed. Evaluation of machine translation and its evaluation. In *Proceedings of the Machine Translation Summit IX*, pages 386–393, 2003.
Cited on page 29.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS)*, pages 6000–6010, 2017.
Cited on page 19.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical Report 25/2013 in Aalto University publication series SCIENCE + TECHNOLOGY, Aalto University, Finland, 2013. URL `http://urn.fi/URN:ISBN:978-952-60-5501-5`.
Cited on page 20.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL) – Volume 1*, pages 1264–1274, 2018.
Cited on pages 26 and 86.

Longyue Wang, Zhaopeng Tu, Andy Way, and Qun Liu. Exploiting cross-sentence context for neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2826–2831, 2017a.
Cited on page 26.

Tian Wang and Kyunghyun Cho. Larger-context language modelling with recurrent neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL) – Volume 1*, pages 1319–1329, 2016.
Cited on page 26.

Yuguang Wang, Shanbo Cheng, Liyang Jiang, Jiajun Yang, Wei Chen, Muze Li, Lin Shi, Yanfeng Wang, and Hongtao Yang. Sogou neural machine translation systems for WMT17. In *Proceedings of the Second Conference on Machine Translation (WMT)*, pages 410–415, 2017b.
Cited on pages 74 and 86.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, 2007.
Cited on page 15.

Lior Wolf, Yair Hanani, Kfir Bar, and Nachum Dershowitz. Joint word2vec networks for bilingual semantic representations. *International Journal of Computational Linguistics and Applications (IJCLA)*, 5(1):27–42, 2014.
Cited on page 45.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Viet Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016. URL `http://arxiv.org/abs/1609.08144`.
Cited on pages 7 and 9.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. Document-level consistency verification in machine translation. In *Proceedings of the Machine Translation Summit XIII*, pages 131–138, 2011.
Cited on page 21.

Zhilin Yang, Bhuwan Dhingra, Ye Yuan, Junjie Hu, William W. Cohen, and Ruslan Salakhutdinov. Words or characters? Fine-grained gating for reading comprehension. In *Proceedings of the 5th International Conference on Learning Representations (ICLR)*, 2017.
Cited on page 73.

Liang Zhou, Chin-Yew Lin, and Eduard Hovy. Re-evaluating machine translation results with paraphrase support. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 77–84, 2006.
Cited on page 28.

Appendix A

Decoding with Ants

Here we detail the ACO-based decoding algorithm briefly introduced in Section 2.2.3. We also provide an analysis of the asymptotic time complexity of the approach and describe the experiments conducted to assess its behaviour, together with the obtained results. To ease the presentation, we formalize the algorithm at sentence-level. Its generalization from sentence-level to document-level requires just a few changes: the input would be a sequence of sentences, and the sole additional restriction for the ants is that the translations they produce must not change the order of the sentences nor intersperse their phrases.

A.1 Decoding Method

The decoding method builds on the phrase-based SMT model of Koehn et al. (2003) and a particular sort of ACO: the $\mathcal{MAX}$–$\mathcal{MIN}$ Ant System implemented in the Hyper-Cube Framework (Blum and Dorigo, 2004; Stützle and Hoos, 2000).

From phrase-based SMT we take several underlying ingredients. First, the input of the problem is a *source sentence* S, that is, a sequence $s_1.s_2 \dots s_N$ of words, with $N > 0$. Second, translations are *scored* with a combination of feature models comparable to those implemented in Moses (Koehn et al., 2007), i.e., translation model, language model, distortion model, and phrase and word penalties. Third, the universe of possible translations of S is obtained from the *phrase table* associated to the translation model, that is, from a function pt mapping source phrases to the set of target phrases that are possible translations for them. And fourth, tackling the *problem of translating* consists in (*i*) segmenting S into a sequence of phrases occurring in pt, (*ii*) reordering that sequence of phrases, and (*iii*) choosing a translation option from pt for each phrase, all while maximizing the score. To ease the presentation, we make two assumptions on pt. On the one hand, we assume that out-of-vocabulary words in the source sentence have already been identified and inserted into pt with an untranslated target. That is, if a word s_j satisfies that no phrase $s_i \dots s_j \dots s_k$ of S, with $1 \leq i \leq j \leq k \leq N$, belongs to $\mathrm{dom}(\mathsf{pt})$, then s_j is considered an out-of-vocabulary

Algorithm 1 Decoding a sentence with ant colony optimization.

`Input`: a source sentence S.

(1) Generate graph G for the sentence S, initializing all pheromone values to the midpoint $\tau_{\mathsf{mid}} := (\tau_{\mathsf{min}} + \tau_{\mathsf{max}})/2$.

(2) Initialize $t_{\mathsf{rb}}, t_{\mathsf{bs}}$ as fictitious translations with score $-\infty$.

(3) Initialize $\mathsf{bsUpdate}$ to false and $\mathsf{convergenceFactor}$ to 0.

(4) For i from 1 to I do:

  (a) Generate translation t_a^i for each ant $a \in \{1, \ldots, A\}$.

  (b) Set t_{ib} to be the translation with highest score among $t_1^i, \ldots, t_A^i$.
  Set t_{rb} to be the translation with highest score among $t_1^i, \ldots, t_A^i, t_{\mathsf{rb}}$.
  Set t_{bs} to be the translation with highest score among $t_1^i, \ldots, t_A^i, t_{\mathsf{bs}}$.

  (c) Update pheromone according to the translations $t_{\mathsf{ib}}, t_{\mathsf{rb}}, t_{\mathsf{bs}}$ and the variables $\mathsf{bsUpdate}$ and $\mathsf{convergenceFactor}$.

  (d) Set $\mathsf{convergenceFactor}$ to $\frac{1}{|\mathcal{T}|} \sum_{\mathcal{T}_j \in \mathcal{T}} \frac{|\tau_j - \tau_{\mathsf{mid}}|}{(\tau_{\mathsf{max}} - \tau_{\mathsf{min}})/2}$.

  (e) If $\mathsf{convergenceFactor} > C$ do:

    (i) If $\mathsf{bsUpdate}$ is true, then perform a restart as follows:

      - Set all pheromone values to τ_{mid}.
      - Set t_{rb} to a fictitious translation with score $-\infty$.

    (ii) Set $\mathsf{bsUpdate}$ to $\neg\mathsf{bsUpdate}$.

`Output`: translation t_{bs}.

word and pt is altered such that $\mathsf{pt}(s_j) = \{s_j\}$. On the other hand, even though pt is a partial function, we assume that undefined entries are mapped to an empty set of translation options. So, for any phrase $s_i \ldots s_j$ of S, with $1 \leq i \leq j \leq N$, we assume $\mathsf{pt}(s_i \ldots s_j) = \emptyset$ when $s_i \ldots s_j \notin \mathrm{dom}(\mathsf{pt})$.

From ACO we take the general optimization strategy, which is summarized in Algorithm 1. In this setting, a translation of S is identified with a *path* that an *ant* has followed through the so-called *construction graph* (see Figure A.1). In our case, each node of the graph corresponds to a translation option from pt for a phrase $s_i \ldots s_j$ of S, and thus, a path through the graph defines a sequence of translations of phrases of S. Not all paths are valid translations: they must satisfy that each source word s_i is translated by exactly one node of the path. The ants walk guided by a probability distribution, which is based on certain *heuristic* information and the amount of *pheromone* that the ants leave on the construction graph. The place where pheromone is deposited is formalized as the *pheromone model* $\mathcal{T}$, which is a collection of *pheromone trail parameters* $\mathcal{T}_i$, each with a value τ_i. A usual definition of $\mathcal{T}$ is to have a parameter $\mathcal{T}_i$ associated to either each edge of the construction graph or to each node. The general overview of the process is straightforward: a swarm with a given amount $A > 0$ of ants is released onto the construction graph to obtain A translations, these translations are used to update the pheromone, and this process is repeated until reaching a termination criterion; the final output is the translation with highest score that has been constructed. Our implementation uses as termination

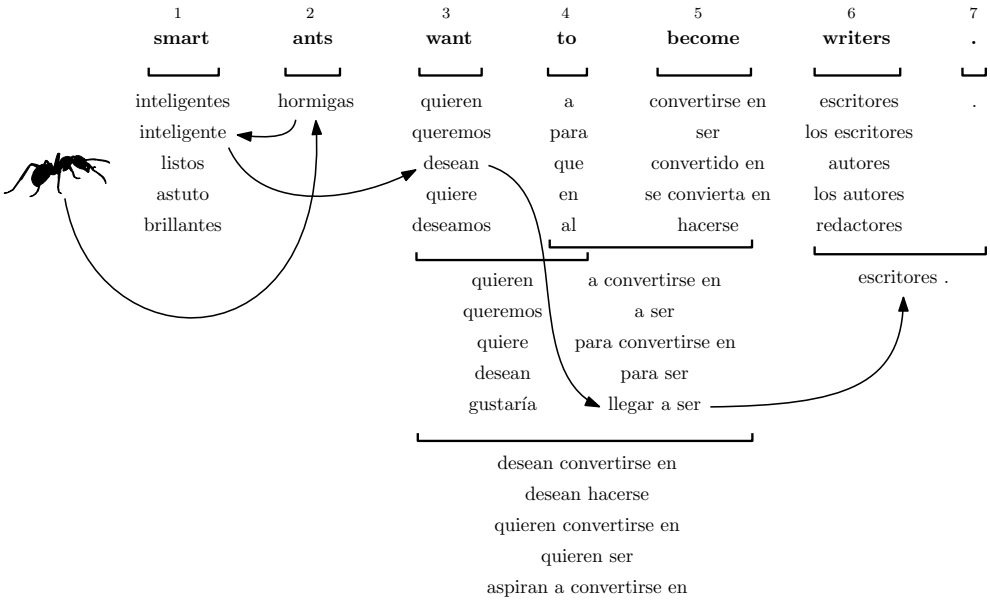

Figure A.1: Sketch of the construction graph for the source sentence "*smart ants want to become writers.*" (each of the Spanish phrases corresponds to a node of the graph; edges are omitted to avoid clutter) and path that an ant follows through it to form the translation "*hormigas inteligente desean llegar a ser escritores.*" with some gender and number disagreements. For each phrase of the source sentence we show up to 5 translations, listed by decreasing probability. Note that only 3 two-token phrases ("*want to*", "*to become*", and "*writers .*") and 1 three-token phrase ("*want to become*") have translations. Also note that the nodes visited in the path translate each of the 7 source tokens, and do it just once.

criterion a given amount $I > 0$ of iterations, although different criteria—such as, for example, a time budget—could be used.

One of the most important aspects of ACO is the handling of pheromone. In our case, the value of each pheromone trail parameter is kept within a given range $[\tau_{\mathsf{min}}, \tau_{\mathsf{max}}]$ throughout the process and it is initially set to the midpoint $\tau_{\mathsf{mid}} := (\tau_{\mathsf{min}} + \tau_{\mathsf{max}})/2$. One of the benefits of having the value of each $\mathcal{T}_i$ restricted to $[\tau_{\mathsf{min}}, \tau_{\mathsf{max}}]$ is that a convergence of the pheromone distribution can easily be identified. It suffices to compute a factor in $[0, 1]$ and compare it to a given threshold $C \in [0, 1)$, with 0 corresponding to the initial, uniform distribution of pheromone and 1 to the case where the value of each $\mathcal{T}_i$ has already reached either τ_{min} or τ_{max}. When convergence is detected, the algorithm escapes this situation by restarting itself, which consists in resetting the pheromone of each $\mathcal{T}_i$ back to its initial value τ_{mid}. Nevertheless, not all the obtained information is lost during the restart: the algorithm keeps track of the best translation found since the beginning of the process (called the *best-so-far* or bs) and uses it to bias the pheromone distribution during a few iterations

Variable name	**Value when** bsUpdate **is false and** convergenceFactor **is in:** [0, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1]	**Value when** bsUpdate **is true**
κ_{ib}	1	2/3	1/3	0	0
κ_{rb}	0	1/3	2/3	1	0
κ_{bs}	0	0	0	0	1

Table A.1: Value associated to the variables $\kappa_{\mathsf{ib}}, \kappa_{\mathsf{rb}}, \kappa_{\mathsf{bs}}$ of the pheromone update.

between the instant when convergence is detected and the instant when the restart is actually performed. In this way, both old and new information get combined when ants construct translations during convergence. In general, the pheromone update performed at the end of each iteration consists in, first, evaporating a given fraction ρ (called the *learning rate*) of the value of each pheromone trail parameter, and second, increasing the value of pheromone trail parameters involved in the construction of (at most) three translations: the best one found in the current iteration (called the *iteration-best* or ib), the best one found since the beginning of the process (i.e., the *best-so-far* or bs), and the best one found since the last time that the process was restarted (called the *restart-best* or rb). Resulting amounts of pheromone below τ_{min} are increased to τ_{min} and amounts above τ_{max} are decreased to τ_{max}. More precisely, for each pheromone trail parameter $\mathcal{T}_i \in \mathcal{T}$, with i being an edge or node of the construction graph depending on the chosen definition, its new pheromone value is:

$$\max\{\tau_{\mathsf{min}}, \min\{\tau_{\mathsf{max}}, (1-\rho)\cdot\tau_i + \rho\cdot(\kappa'_{\mathsf{ib}} + \kappa'_{\mathsf{rb}} + \kappa'_{\mathsf{bs}})\}\}$$

where the term κ'_{ib} is 0 when i is not part of the path followed to define the iteration-best translation and, otherwise, it is κ_{ib} as detailed in Table A.1, analogously for the term κ'_{rb} corresponding to the restart-best translation and the term κ'_{bs} corresponding to the best-so-far translation.

We have considered three distinct variants of the decoding method, which we denote as ACODec, $\text{ACODec}^{\text{mono}}_{\text{indep}}$, and $\text{ACODec}^{\text{mono}}_{\text{share}}$. In ACODec, the pheromone trail parameters are associated to each edge of the construction graph and ants are free to walk along the edges in any order, as long as the paths they follow correspond to valid translations. $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ follow the same general scheme as ACODec, but modify it on two fronts. On the one hand, instead of having ants define full translations, they work *incrementally*: ants take their translations from the previous iteration, erase a part of each of them, and then fill in the holes. On the other hand, we simplify the problem by forcing ants to always walk in *monotonic* order, that is, the translations they create must keep the order of phrases from the source sentence. To be able to obtain arbitrarily-reordered translations, the phrases of each translation are later permuted with a local search like the one presented by Hardmeier et al. (2012). The main benefit of having ants advance in monotonic order is that we can easily shift the pheromone trail parameters from the edges to the nodes (thus, pheromone only models decisions concerning segmentation and translation, but not concerning reordering) and reduce the size of the construction graph (edges can be left implicit). The single difference between $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ is that in the former each ant has an *independent* translation to work on, whereas in the latter all

ants *share* a common translations to work on. The following sections detail for each approach the generation of the construction graph (step 1 of Algorithm 1) and the definition of the ant paths and their associated translations (step 4.a of Algorithm 1).

As a final remark, note that we have assumed that ACODec, $\text{ACODec}^{\text{mono}}_{\text{indep}}$, and $\text{ACODec}^{\text{mono}}_{\text{share}}$ start uninformed. Nevertheless, it is also possible to precompute a translation t as starting point for the process. In such case, step 2 of the algorithm initializes t_{rb} and t_{bs} to t. Additionally, $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ also set $t^0_1, \ldots, t^0_A$ to t, such that in the very first iteration each ant already has a previous translation to work on.

A.1.1 Detailed Steps of ACODec

Step 1 – Graph Construction and Pheromone

The directed graph $G = \langle V, E \rangle$ is constructed as follows. Each node of V corresponds to a translation option of a single segment of the source sentence. To represent it, we use tuples of the form $\langle i, j, t \rangle$ as nodes, where i and j satisfy $1 \leq i \leq j \leq N$ and identify a segment of the source sentence and t is a target phrase that translates this segment. We use $\mathsf{cover}(\langle i, j, t \rangle)$ to denote the set $\{i, \ldots, j\}$ and $\mathsf{target}(\langle i, j, t \rangle)$ to denote t. Additionally, V contains one extra node used as the special starting point for the ants: $\mathsf{start} = \langle 0, 0, \varepsilon \rangle$, where ε is the empty sequence. Note that start identifies a fictitious segment of the source sentence, which is assumed to cover a single non-existing word at index 0. Overall, the set of nodes of G is:

$$V = \{\mathsf{start}\} \uplus \{\langle i, j, t \rangle \mid 1 \leq i \leq j \leq N \ \wedge \ t \in \mathsf{pt}(s_i \ldots s_j)\}$$

The set E of directed edges is a subset of $V \times V$, not containing the following undesirable and useless edges. First, we do not need an edge connecting any two nodes whose segments overlap, as no ant is allowed to walk along such an edge since those nodes cannot appear together in a translation. We say that two nodes n, n' overlap, denoted by $\mathsf{overlap}(n, n')$, when the source segments they specify overlap, i.e., when $\mathsf{cover}(n) \cap \mathsf{cover}(n')$ is non-empty. Second, we omit those edges that exceed a given maximum *distortion* $D \geq 0$. The distortion is a measure on how much the source segments are reordered in the respective translation. It is computed as the amount of source words separating the end of one segment from the beginning of the next segment; formally: $\mathsf{distortion}(n, n') = |\min(\mathsf{cover}(n')) - (\max(\mathsf{cover}(n)) + 1)|$. Note that, as expected, the distortion is 0 when the segment of the second node starts just after the segment of the first node. Third, we also discard edges directed towards start, since such node is only used as starting point for the ants and should not be visited mid-sentence. Overall, the set of directed edges of G is:

$$E = \{\langle n, n' \rangle \in V^2 \mid \neg\mathsf{overlap}(n, n') \ \wedge \ \mathsf{distortion}(n, n') \leq D \ \wedge \ n' \neq \mathsf{start}\}$$

Note that for edges of the form $\langle \mathsf{start}, n' \rangle$ we also impose $\mathsf{distortion}(\mathsf{start}, n') \leq D$. This is because we also want the beginning of the sentence to respect the maximum distortion.

We associate to each directed edge $e \in E$ a pheromone trail parameter $\mathcal{T}_e$, with its value τ_e initialized to the amount τ_{mid}.

Step 4.a – Ant Paths and Associated Translations

A *path* through the graph G is a sequence $p = n_0.n_1 \dots n_m$ of nodes of V, with $0 < m \leq N$, satisfying these conditions: (i) it begins with start, i.e., $n_0 = \mathsf{start}$, (ii) the nodes are pairwise non-overlapping, (iii) the directed edge $\langle n_i, n_{i+1} \rangle$ is in E for each $i \in \{0, \dots, m-1\}$, and (iv) the nodes cover the whole sentence, i.e., $\mathsf{cover}(p) := \mathsf{cover}(n_0) \uplus \mathsf{cover}(n_1) \uplus \dots \uplus \mathsf{cover}(n_m)$ coincides with $\{0, \dots, N\}$. To construct such a path p, an ant proceeds by starting at start and then walking along edges while guaranteeing that the path conditions (i)-(iii) are met. The ant proceeds until (iv) is also met, at which point the followed path p straightforwardly corresponds to a valid translation for the source sentence: $\mathsf{target}(n_1) \dots \mathsf{target}(n_m)$.

The main difficulty of constructing a path is how an ant performs a step, in particular, how it chooses an out-edge among all the out-edges of the current node. Consider a partially constructed path $p = n_0.n_1 \dots n_k$, with $0 \leq k < N$ and satisfying (i)-(iii) but not (iv). To proceed, the ant has to choose one of the out-edges of node n_k. Note that, possibly, not all out-edges in E are valid, since some might lead to nodes overlapping with the nodes already in p, thus creating a conflict concerning (ii). Let $E|_p$ be the out-edges of n_k that avoid such overlap, i.e.:

$$E|_p = \{\langle n_k, n \rangle \in E \mid \mathsf{cover}(p) \cap \mathsf{cover}(n) = \emptyset\}$$

We further prune this set to preemptively avert paths that would lead the ant into a dead end. In particular, we do not want to reach a situation where the remaining untranslated parts of the sentence cannot be segmented or would require steps exceeding the maximum distortion D. For the latter goal, we use the same conservative method as in Moses: we do not allow a jump forward when the index immediately following the segment corresponding to the destination node of the jump is further than D from any preceding untranslated word in the sentence. Formally, let u be the least index of a still-uncovered word, i.e., $u = \min(\{1, \dots, N\} \setminus \mathsf{cover}(p))$, then the pruned set is:

$$E\|_p = \big\{\langle n_k, n \rangle \in E|_p \mid \big(\exists \bar{V} \subseteq V : \mathsf{segmentation}(\bar{V}, \{1, \dots, N\} \setminus \mathsf{cover}(p.n))\big) \wedge \big(u \in \mathsf{cover}(n) \vee (|u - (\max(\mathsf{cover}(n)) + 1)| \leq D)\big)\big\}$$

where $\mathsf{segmentation}(\bar{V}, X)$ tests whether the given collection of nodes are pairwise non-overlapping and fully cover the given set of source word indexes, i.e.:

$$\mathsf{segmentation}(\bar{V}, X) = \big(\forall \bar{n}_1, \bar{n}_2 \in \bar{V} : (\bar{n}_1 \neq \bar{n}_2 \Rightarrow \neg \mathsf{overlap}(\bar{n}_1, \bar{n}_2))\big) \wedge \big(X = \biguplus_{\bar{n} \in \bar{V}} \mathsf{cover}(\bar{n})\big)$$

This concludes the identification of the candidate edges for the current step, but it still remains to assess their quality so that the ant can bias the selection towards the most promising ones. To this end, each edge $e = \langle n_k, n \rangle \in E\|_p$ has an associated weight w_e defined as:

$$w_e = \alpha \cdot \tau_e + (1 - \alpha) \cdot h(n, \{1, \dots, N\} \setminus \mathsf{cover}(p))$$

where $\alpha \in [0, 1]$ is a given constant and $h(n, Y) \in [0, 1]$ is a heuristic measure on the quality of choosing node n to extend the partial path when having the source words at

the indexes of Y still untranslated. This heuristic is based on the *future cost* of Koehn et al. (2003), which estimates the score attainable on the untranslated parts of the sentence. More precisely:

$$h(n, Y) = \frac{\max(\mathsf{futures}(Y, \{n\})) - \min(\mathsf{futures}(Y, \emptyset))}{\max(\mathsf{futures}(Y, \emptyset)) - \min(\mathsf{futures}(Y, \emptyset))}$$

where $\mathsf{futures}(Y, Z)$ is the following set of score estimates:

$$\left\{ \sum_{\bar{n} \in \bar{V}} \mathsf{estimate}(\bar{n}) \mid Z \subseteq \bar{V} \subseteq V \wedge \mathsf{segmentation}(\bar{V}, Y) \right\}$$

with $\mathsf{estimate}(\bar{n})$ measuring the score of using the translation option $\mathsf{target}(\bar{n})$ for the segment $\mathsf{cover}(\bar{n})$ of the source sentence. The function $\mathsf{estimate}$ is computed as an approximation of the models conforming our score function, except for the distortion model (Koehn et al., 2003). When the set $\mathsf{futures}$ has only one element, we assume $h(n, Y) = 1$ to avoid a division by 0.

Finally, the probability of each edge $e \in E\|_p$ of being selected by the ant in the current step is:

$$p_e = \frac{w_e}{\sum_{e' \in E\|_p} w_{e'}}$$

Nevertheless, the ant first decides whether the selection of an edge among all the candidates in $E\|_p$ is to be deterministic (the one with maximal weight w_e) or random (following the probabilities p_e). The case is decided randomly: with a given probability q_0 (called the *determinism rate*) the deterministic decision is performed.

Once an edge $e \in E\|_p$ has been chosen, the partial path p is extended with the destiny node of e. This extension clearly still satisfies (i)-(iii) due to how $E\|_p$ has been defined.

A.1.2 Detailed Steps of ACODec$^{\mathbf{mono}}_{\mathbf{indep}}$ and ACODec$^{\mathbf{mono}}_{\mathbf{share}}$

Step 1 – Graph Construction and Pheromone

Even though similar to the construction graph of ACODec, the one for ACODec$^{\text{mono}}_{\text{indep}}$ and ACODec$^{\text{mono}}_{\text{share}}$ is simpler. In particular, the set of nodes is the same except for the start node, which is now not needed. Also, the pheromone trail parameters in this case are associated to nodes instead of edges. Finally, the set of edges could, in principle, simply contain the edges of the form $\langle\langle i, j, t\rangle, \langle j + 1, k, t'\rangle\rangle$ that are needed for monotonic steps. Nevertheless, since edges do not have any associated data and are not needed for any special purpose, we leave them implicit in our implementation.

Step 4.a – Ant Paths and Associated Translations

A *path* through G is, in this occasion, a sequence $p = n_1 \dots n_m$ of nodes of V, with $0 < m \leq N$, satisfying that (i) the nodes are pairwise non-overlapping and (ii) the nodes cover the whole sentence, i.e., $\mathsf{cover}(p) := \mathsf{cover}(n_1) \uplus \dots \uplus \mathsf{cover}(n_m)$ coincides with $\{1, \dots, N\}$. To construct such a path, first, each ant a takes a path from the previous iteration (when available), erases a part of it, and fills it again with a new monotonic translation for the created hole. Second, the sequence of nodes

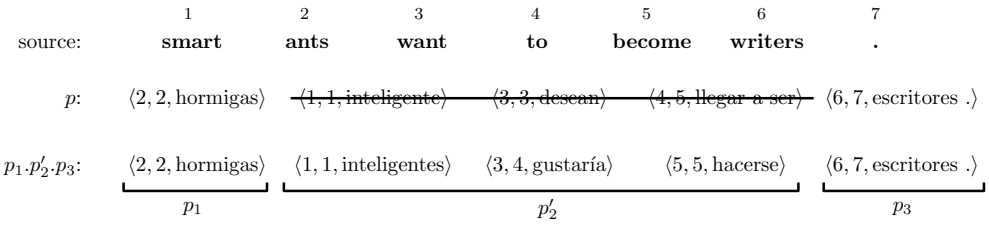

Figure A.2: Following the example in Figure A.1, the ant creates a hole of length $\ell = 3$ in the path p of the translation from the previous iteration (the hole is depicted as the stricken-through nodes). The affected source indexes for such hole are $K = \{1, 3, 4, 5\}$, which are partitioned into groups of serial indexes as $K_1 = \{1\}$ and $K_2 = \{3, 4, 5\}$. For the first group K_1, the ant fills the hole with the single node $\langle 1, 1, \text{inteligentes}\rangle$. For the second group K_2, it uses the two nodes $\langle 3, 4, \text{gustaría}\rangle$ and $\langle 5, 5, \text{hacerse}\rangle$. In this latter case, according to the graph in Figure A.1 it would also have been possible to fill the hole with one single node (e.g., $\langle 3, 5, \text{aspiran a convertirse en}\rangle$) or with three individual nodes.

thus obtained by the ant a is permuted to produce a new (possibly non-monotonic) sequence. The permutation is performed by means of a local search like the one by Hardmeier et al. (2012), which runs for a pre-defined amount R of steps. From the resulting permuted sequence we directly obtain the translation t_a^i associated to the ant a in the current iteration i.

It only remains to detail the tasks performed by the ant a. Its first task is to locate a previous path p to work on. In the case where t_{rb} is a fictitious translation with $-\infty$ score, there is no such path because the process is either at the very first iteration or at an iteration immediately following a restart. Otherwise, in $\text{ACODec}^{\text{mono}}_{\text{indep}}$ the ant a uses as p the path of its own translation t_a^{i-1} from the previous iteration, and in $\text{ACODec}^{\text{mono}}_{\text{share}}$ the ant a uses as p the path of the translation with highest score among all the translations $t_1^{i-1}, \ldots, t_A^{i-1}$ from the previous iteration. From this point onward, $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ behave identically.

The second task of the ant is to create a hole in p (see Figure A.2). This consists in splitting the sequence p into $p_1.p_2.p_3$, where p_1 and p_3 are to be preserved whereas the source words at the set of indexes $K = \mathsf{cover}(p_2)$ are to be monotonically re-translated. The split of p is performed randomly: the length ℓ for the infix p_2 is the minimum between the length $|p|$ of the whole sequence and a value chosen uniformly from a given range $\{\ell_1, \ldots, \ell_2\}$, the length for the prefix p_1 is chosen uniformly from $\{0, \ldots, |p| - \ell\}$, whereas the length of the suffix p_3 is univocally determined by the previous ones. In the special case where the ant has no previous path p to work on, we simply assume $|p_1| = |p_3| = 0$ and $K = \{1, \ldots, N\}$, i.e., as if the ant had created a hole covering the whole path.

Third, let K be partitioned into maximal sets $K_1, \ldots, K_k$ of serial indexes, i.e., the K_j's are non-empty, pairwise disjoint subsets of K, their union coincides with K, and each of them is of the form $\{j_0, \ldots, j_1\}$ such that $j_0 - 1, j_1 + 1 \notin K$. Furthermore, we assume that they are ordered, i.e., $\max(K_j) < \min(K_{j+1})$. Intuitively, p_2 was a

hole in the translation, whereas each K_j is a part of this hole when projected onto the source.

Finally, to produce the desired path $p_1.p'_2.p_3$, the ant iteratively grows p'_2 from an empty sequence of nodes by processing the K_j's in order. For each K_j, it proceeds as follows: while the set $P := K_j \setminus \mathsf{cover}(p'_2)$ is non-empty, the ant appends to p'_2 a node $n \in V$ chosen among those ones that cover only indexes of P and cover at least $\min(P)$. The selection of the node n follows the same ideas as to how an ant chooses an edge in ACODec. In particular, the set of candidate nodes $V|_P$ is defined like:

$$V|_P = \{n \in V \mid \min(P) \in \mathsf{cover}(n) \subseteq P\}$$

and each node $n \in V|_P$ has an associated weight w_n defined as:

$$w_n = \alpha \cdot \tau_n + (1 - \alpha) \cdot h(n, P)$$

where h is the same heuristic as in ACODec, transformed into a probability as follows:

$$p_n = \frac{w_n}{\sum_{n' \in V|_P} w_{n'}}$$

The ant chooses a node from $V|_P$ either deterministically (the one with maximal weight w_n) or randomly (following the probabilities p_n), and which strategy to use is decided with a given probability q_0 (i.e., *determinism rate*) for the former.

Note that, in contrast to ACODec, the ants ignore the maximum distortion D. This is not a problem in practice, since translations are usually quite monotonic, and thus, patching a hole monotonically has little chances of exceeding the limit D. Nevertheless, if it did happen, the hill-climbing reordering could correct the violation.

A.2 Time Complexity

For the asymptotic time complexity of ACODec, first note that the amount $|V|$ of nodes in the graph G is in $\mathcal{O}(N \cdot L \cdot T)$, where L is the maximum length of a source phrase in pt, i.e., $L = \max\{j - i + 1 : 1 \leq i \leq j \leq N \wedge s_i \dots s_j \in \mathrm{dom}(\mathsf{pt})\}$, and T is the maximum amount of translation options in pt for any source phrase, i.e., $T = \max\{|\mathsf{pt}(s_i \dots s_j)| : 1 \leq i \leq j \leq N\}$. Second, the amount $|E|$ of edges in G is bounded by $|V|^2$, but a tighter bound can be obtained by taking into account the maximum distortion D that restricts the connectivity of the graph. In particular, each node of G is connected with out-edges to at most $\mathsf{outdegree} = (2 \cdot D + 1) \cdot L \cdot T$ other nodes. Thus, $|E|$ is in $\mathcal{O}(|V| \cdot \mathsf{outdegree}) = \mathcal{O}(N \cdot L^2 \cdot T^2 \cdot D)$. This bound also holds for the size $|G|$ of the graph, since $|E|$ is the factor that dominates it. Clearly, the construction of the graph can be done with time linear in $|G|$. The time each iteration takes can be bounded as follows. For each ant, the time needed for constructing its path is proportional to the amount of source words and the maximum out-degree of the graph, so it is in $\mathcal{O}(N \cdot \mathsf{outdegree} \cdot H(N))$, where $H(.)$ denotes the time it takes to compute the heuristic information as a function of the source sentence length. Reconstructing a translation from a path is straightforward, but we also need to compute its associated score. We use $F(.)$ to denote the time it takes to

compute the score as a function of the sentence length. This length must take into account the source and the produced target. Nevertheless, the size of the target can be bounded by the size of the source multiplied by a factor depending on pt, and since pt is part of the scoring, we assume that any increase in size in the target side is already handled by F. So, reconstructing a translation from a path and scoring it takes time in $\mathcal{O}(N + F(N))$, which is equivalent to $\mathcal{O}(F(N))$ since F is at least linear. Since we use A ants, obtaining all the paths, reconstructing the associated translations, and scoring them takes time in $\mathcal{O}(A \cdot (N \cdot \mathsf{outdegree} \cdot H(N) + F(N)))$. The time needed for updating the pheromone is not completely subsumed in the previous expression, as the whole graph must be traversed for the evaporation. Thus, the time each iteration takes is in $\mathcal{O}(A \cdot (N \cdot \mathsf{outdegree} \cdot H(N) + F(N)) + |G|) = \mathcal{O}(N \cdot \mathsf{outdegree} \cdot (A \cdot H(N) + L \cdot T) + A \cdot F(N))$. Since the process is iterated I times, the overall time complexity is $\mathcal{O}(I \cdot (N \cdot \mathsf{outdegree} \cdot (A \cdot H(N) + L \cdot T) + A \cdot F(N)))$. Assuming L, T, D to be fixed constants, this reduces to $\mathcal{O}(I \cdot A \cdot (N \cdot H(N) + F(N)))$.

A similar analysis can be made for $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$. In this case, since we do not actually generate the edges of the graph, $|G|$ is in $\mathcal{O}(N \cdot L \cdot T)$ and the same bound holds for the time complexity of generating G. The time each iteration takes can be bounded as follows. For each ant, the time needed for incrementally constructing its path is proportional to the amount of source words and the maximum amount of options for each step, so it is in $\mathcal{O}(N \cdot L \cdot T \cdot H(N))$. Obtaining the translation from that path, scoring it, and reordering it through R steps of hill climbing takes time in $\mathcal{O}(N + F(N) + R \cdot F(N)) = \mathcal{O}(R \cdot F(N))$. This is a very loose upper bound, as the scoring in hill climbing is optimized by doing incremental computations in each step, as proposed by Hardmeier et al. (2013). Since we use A ants, obtaining all the paths, reconstructing the associated translations, and scoring and reordering them takes time in $\mathcal{O}(A \cdot (N \cdot L \cdot T \cdot H(N) + R \cdot F(N)))$. The time for updating the pheromone is completely subsumed in the previous expression. Since the process is iterated I times, the overall time complexity is $\mathcal{O}(I \cdot A \cdot (N \cdot L \cdot T \cdot H(N) + R \cdot F(N)))$, and assuming L, T to be fixed constants it reduces to $\mathcal{O}(I \cdot A \cdot (N \cdot H(N) + R \cdot F(N)))$.

A.3 Experiments

We perform English-to-Spanish translation experiments under settings comparable to the ones of Section 4.4.1, but with three modifications. First, since the size of the phrase table is crucial in the performance of our ACO approaches, we filter it to only retain for each source phrase, at most, the $T = 30$ target phrases with highest $p(tgt|src)$ probability. We do not alter the value of other related parameters that also affect performance, such as the maximum length $L = 7$ of the source phrases of the phrase table or the maximum distortion $D = 6$ of the distortion model. Second, we reuse the baseline Moses system, but disabling its lexical reordering feature. The rationale for this modification is that the other systems do not implement an equivalent feature function, and we want to compare the systems when facing the exact same optimization problem (i.e., the problem of finding the best translation according to a scoring function shared by all the systems). We perform a new feature weight optimization for this Moses system variation, using MERT (Och, 2003) against the

System	q_0	C	$[\tau_{\mathsf{min}}, \tau_{\mathsf{max}}]$	α
ACODec	0.8828125	0.999	$[0.001, 0.999]$	0.671875
$\text{ACODec}^{\text{mono}}_{\text{indep}}$	0.8671875	0.99999	$[0.0001, 0.9999]$	0.796875
$\text{ACODec}^{\text{mono}}_{\text{share}}$	0.7734375	0.99999	$[0.0001, 0.9999]$	0.671875

Table A.2: Grid-tuned parameter values of the three ACO variants.

System	WER↓	PER↓	TER↓	BLEU↑	NIST↑	$\text{METEOR}_{\text{pa}}$↑	ULC↑
MOSES	59.44	40.08	53.80	27.28	7.3262	49.97	53.12
LEHRER	61.71	41.31	56.10	24.78	7.0516	48.26	48.65
ACODec	62.30	43.48	57.41	21.99	6.8666	46.69	44.63
$\text{ACODec}^{\text{mono}}_{\text{indep}}$	60.46	41.85	55.19	24.73	7.1326	48.51	49.27
$\text{ACODec}^{\text{mono}}_{\text{share}}$	60.09	41.14	54.76	25.67	7.2125	48.95	50.67

Table A.3: Automatic evaluation of the systems working on sentences. The ULC is computed over the other metrics of the table.

BLEU metric (Papineni et al., 2002) on the NEWSCOMMENTARY2009 development corpus. Third, we reuse the baseline LEHRER system, but initialized randomly instead of using precomputed MOSES translations, with a maximum quota of 10^7 steps instead of 10^5, and using the feature weights obtained with the MERT-tuning for MOSES that we just mentioned. We still use NEWSCOMMENTARY2010 as test set.

We consider three ACO systems, one for each of the decoding variants, and for all of them we use the same features and weights as the MOSES system. Regarding the parameters specific to ACO, we have identified several non-problematic ones during preliminary experiments, and we have chosen for them values that are similar to the ones in the literature: 10 for the number A of ants and 0.1 for the learning rate ρ. We fix the number I of iterations to 500 as a trade-off between the computation speed and the quality of the obtained results. For the monotonic variants, we fix a limit of $R = 10^4$ steps for the hill-climbing reordering and $\{\ell_1 = 1, \ldots, \ell_2 = 12\}$ as the range for the length of the holes. We perform a grid search to tune the remaining parameters: for the determinism rate q_0 and the fraction α that pheromone contributes to the weights we explore the range $[0, 1]$ at regularly-spaced values, whereas for the convergence threshold C and the pheromone range $[\tau_{\mathsf{min}}, \tau_{\mathsf{max}}]$ we just vary the amount of decimal places of their values (these variables are defined like $C := 1 - 10^{-x}$, $\tau_{\mathsf{min}} := 10^{-y}$, $\tau_{\mathsf{max}} := 1 - 10^{-y}$, and thus, it suffices to test possible values for the positive naturals x and y). Table A.2 shows the obtained values.

Table A.3 shows the result of the automatic evaluation of the systems when translating the whole test set sentence by sentence. MOSES obtains the best results in all the metrics and, of the remaining systems, $\text{ACODec}^{\text{mono}}_{\text{share}}$ systematically outperforms the rest. Except for the pair LEHRER and $\text{ACODec}^{\text{mono}}_{\text{indep}}$, the differences between the systems are statistically significant.[1] Besides this automatic evaluation against the

[1] According to bootstrap resampling (Koehn, 2004) over BLEU and NIST metrics with a p-value of 0.05.

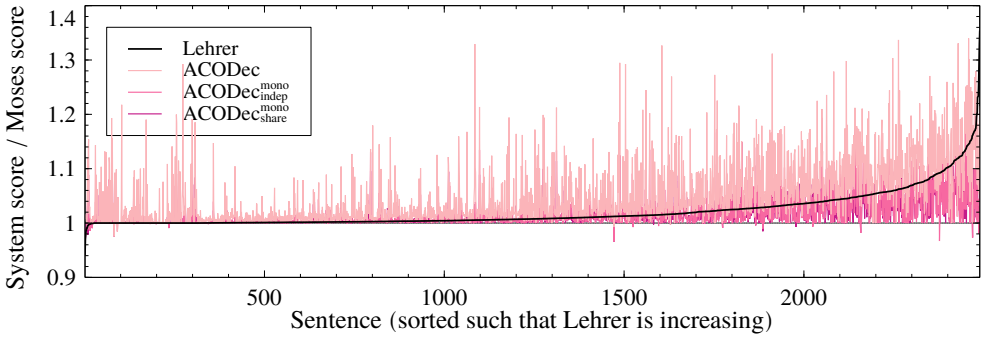

Figure A.3: Normalized scores obtained on each sentence of the test set by each of the compared systems. The score is normalized by dividing by the score of the baseline MOSES; lower is better. The abscissa is sorted to make the plot for LEHRER (black line) increasing.

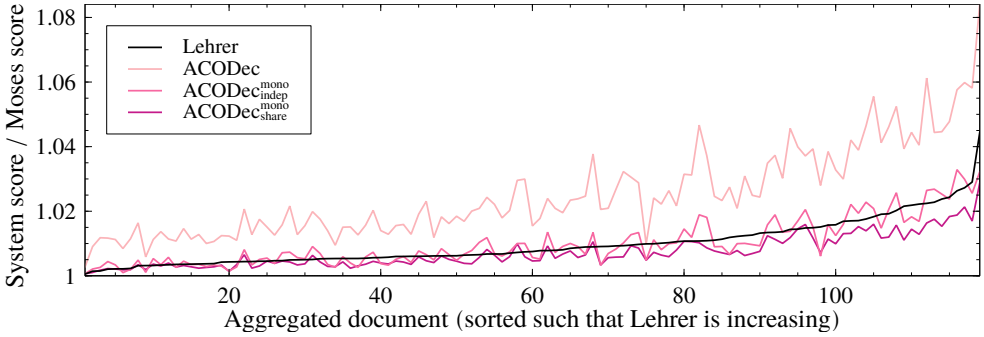

Figure A.4: Same as Figure A.3, but adding together into a single data point the scores of all sentences belonging to the same document.

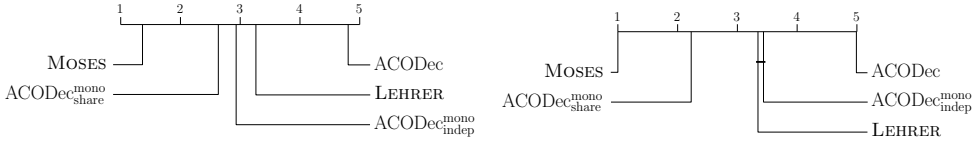

Figure A.5: Critical difference plots for the scores obtained by the systems on the sentences of the test set, both when considering each score individually (left) and when adding together the scores of all sentences belonging to the same document (right). The horizontal axis shows the average ranking of the systems; lower is better. Bold horizontal bars connect systems that are not statistically different for a significance level of 0.05.

reference translations, we also evaluate the systems by comparing the scores of their output as computed by the systems themselves, i.e., we directly compare the score given to the produced translations by the feature functions in use when decoding. Figure A.3 depicts the final scores obtained by each system in each sentence and Figure A.4 presents the same data, but aggregated into fewer data points. In both plots, the scores are normalized by dividing by the score of the respective Moses output. Therefore, since plain scores are negative numbers, all the normalized scores become positive. Furthermore, *lower* normalized values are better. In the latter figure it is possible to observe that ACODec obtains worse translations than Lehrer, and that the score of the translations of $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ follow the same trend as the ones of Lehrer, with a slight advantage for $\text{ACODec}^{\text{mono}}_{\text{share}}$. More precisely, Lehrer is outperformed by ACODec in 12.01% of the sentences, by $\text{ACODec}^{\text{mono}}_{\text{indep}}$ in 54.92%, and by $\text{ACODec}^{\text{mono}}_{\text{share}}$ in 62.76%; when aggregating the sentences into their respective documents as done in Figure A.4, Lehrer is outperformed by ACODec only in 0.84% of the cases, by $\text{ACODec}^{\text{mono}}_{\text{indep}}$ in 46.22%, and by $\text{ACODec}^{\text{mono}}_{\text{share}}$ in 87.39%. To confirm these observations on the score trends, we test statistical differences[2] between the systems (if any) for subsets of the considered inputs. To that end, all systems are compared simultaneously using Friedman's test and afterwards, provided that such test rejects in all cases the hypothesis that the systems perform equally, all possible pairwise comparisons are performed using the Nemenyi post-hoc test (García and Herrera, 2008). The obtained results are displayed in Figure A.5. The shown ranking confirms the findings described above and note that only the results of Lehrer and $\text{ACODec}^{\text{mono}}_{\text{indep}}$ for aggregated scores cannot be considered statistically different. This is in agreement with the evaluation performed with automatic metrics.

The performance of our ACO systems degrades when using full documents as input, leading to Lehrer outperforming the three ACO variants in the automatic metrics (see Table A.4). The same tendency can also be observed when looking at the final scores of the translations as computed by the feature functions in use. Figures A.4 and A.6 correspond to experiments that only differ in that the former has translated each document sentence by sentence whereas the latter has treated each document as a unit. In the latter figure, the normalized score of Lehrer in (almost) every document is a lower bound for—thus better than—the three ACO variants. More precisely, Lehrer is not outperformed by ACODec in any of the documents, by $\text{ACODec}^{\text{mono}}_{\text{indep}}$ in only 0.84%, and by $\text{ACODec}^{\text{mono}}_{\text{share}}$ in 5.04%. To confirm these observations on the score trends, we again test statistical differences (see Figure A.7) and obtain once more an agreement with our analysis and with the automatic metric evaluation.

Table A.5 summarizes the resource usage[3] of the systems when translating the whole test set, proceeding both sentence by sentence and treating each document as a unit. The best runtimes in both scenarios are obtained by Lehrer, with ACODec

[2] All tests are performed with R's scmamp package (Calvo and Santafé, 2016).

[3] All the measurements have been taken on a computer cluster with heterogeneous machines. Its nodes have CPUs from the Intel® Xeon® family (E5450, 5150, 5160, X5550, X5650, X5660, X5670, X5675, E5-2470, and E5-2450 v2) and from the AMD Opteron™ family (2350). Although the performance varies from one machine to another and thus runtimes are not directly comparable, the differences are amortized by the amount of executions taken into account for the averages presented in Table A.5.

System	WER↓	PER↓	TER↓	BLEU↑	NIST↑	METEOR$_{pa}$↑	ULC↑
Moses	59.44	40.08	53.80	27.28	7.3262	49.97	55.97
Lehrer	62.25	41.66	56.74	24.33	6.9813	47.95	50.64
ACODec	67.05	45.16	61.99	18.30	6.3776	43.65	40.25
ACODec$^{mono}_{indep}$	64.76	45.01	60.15	19.99	6.4822	44.84	43.04
ACODec$^{mono}_{share}$	64.07	43.76	59.09	21.34	6.6264	45.82	45.43

Table A.4: Automatic evaluation of the systems working on full documents, except for Moses, which is the same as in Table A.3. The ULC is computed over the other metrics of the table.

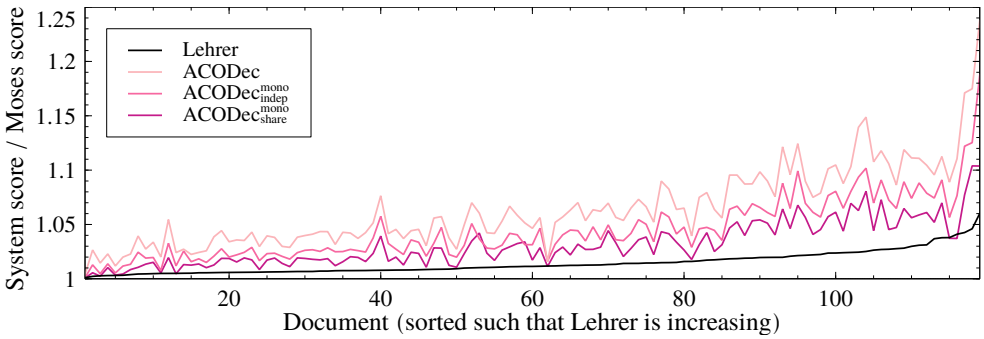

Figure A.6: Normalized scores obtained on each document of the test set by each of the compared systems. The score is normalized by dividing by the score of the baseline Moses; lower is better. The abscissa is sorted to make the plot for Lehrer (black line) increasing.

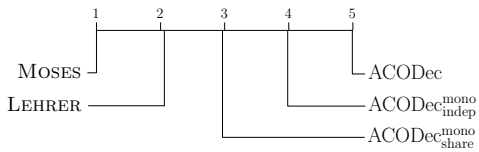

Figure A.7: Critical difference plot for the scores obtained by the systems on the documents of the test set, with same interpretation as Figure A.5.

Input kind	System	CPU (seconds)	RAM (GiB)
Sentence	Moses	29.67	8.73
	Lehrer	2.70	7.49
	ACODec	7.42	7.49
	$\text{ACODec}^{\text{mono}}_{\text{indep}}$	197.06	7.48
	$\text{ACODec}^{\text{mono}}_{\text{share}}$	198.38	7.48
Document	Moses	–	–
	Lehrer	18.97	7.50
	ACODec	122.94	8.14
	$\text{ACODec}^{\text{mono}}_{\text{indep}}$	606.84	7.49
	$\text{ACODec}^{\text{mono}}_{\text{share}}$	588.27	7.49

Table A.5: Mean resource usage of the systems when decoding the test set, working either with sentences or with full documents as input.

Input kind	Tokens	Nodes	Edges
Sentence	24.88	1,235.81	586,536.37
Document	520.37	25,848.27	12,268,486.90

Table A.6: Mean amount of tokens per input of the test set and corresponding mean size of the graphs constructed by ACODec.

achieving a performance close to it when working on sentences. Runtimes for the monotonic ACO variants are significantly higher than for all the other systems, mainly due to the R hill-climbing steps that each ant performs per iteration. With respect to the memory usage, note that it is quite uniform across the systems since it is dominated by the size of the language model loaded as a feature function. Although Moses shows the highest memory usage, it is probably just caused by implementation details. On our systems, the only remarkable datum is that ACODec uses 0.65 GiB of additional memory when working on full documents: this extra space is required since the size of the graphs in this scenario increases significantly with respect to working with individual sentences (see Table A.6). As expected, this increase is not so noticeable for the monotonic ACO variants thanks to the edge elision when constructing the graphs.

To conclude, we show an illustrative example of how the decoding behaves on an input sentence. As a reference, Figure A.8 shows how the score of that sentence translation evolves through time when using Lehrer. Figure A.9 depicts the analogous score evolution with each of the three ACO decoding variants. Note that ACODec converges 8 times (each time can be identified by the sudden jump in the convergence factor, from almost 1 to almost 0), that the mean scores of the translations produced by the A-ant swarm remains rather flat throughout the process, but that the maximum score of each iteration slightly improves over time until convergence is detected. On the other hand, $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ do not converge as fast since they use a higher value for C (the former converging just once and the latter twice) and

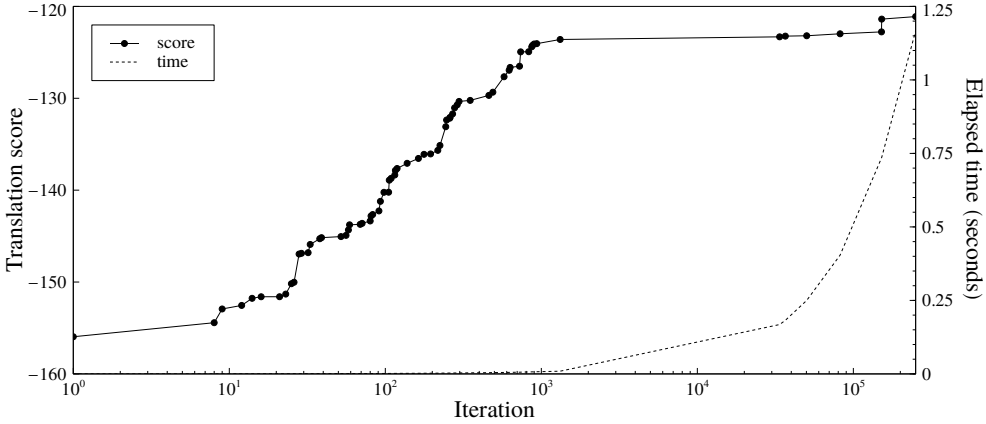

Figure A.8: Evolution of the score as a function of the iteration (in logarithmic scale) when decoding with Lehrer.

both the mean and maximum scores of each iteration improve over time until convergence is detected (the maximum plateauing during some intervals). Also note that $\text{ACODec}^{\text{mono}}_{\text{indep}}$ and $\text{ACODec}^{\text{mono}}_{\text{share}}$ are, roughly, two orders of magnitude slower than ACODec. The runtimes for $\text{ACODec}^{\text{mono}}_{\text{share}}$ shown in the figure are not very stable at the beginning of the process due to the heavy load of the machine at the time of the execution. As a closing remark, the final scores for this input sentence obtained with each decoding approach are as follows: Moses reaches −120.97, Lehrer −121.11, ACODec −124.11, $\text{ACODec}^{\text{mono}}_{\text{indep}}$ −121.86, and $\text{ACODec}^{\text{mono}}_{\text{share}}$ −121.80.

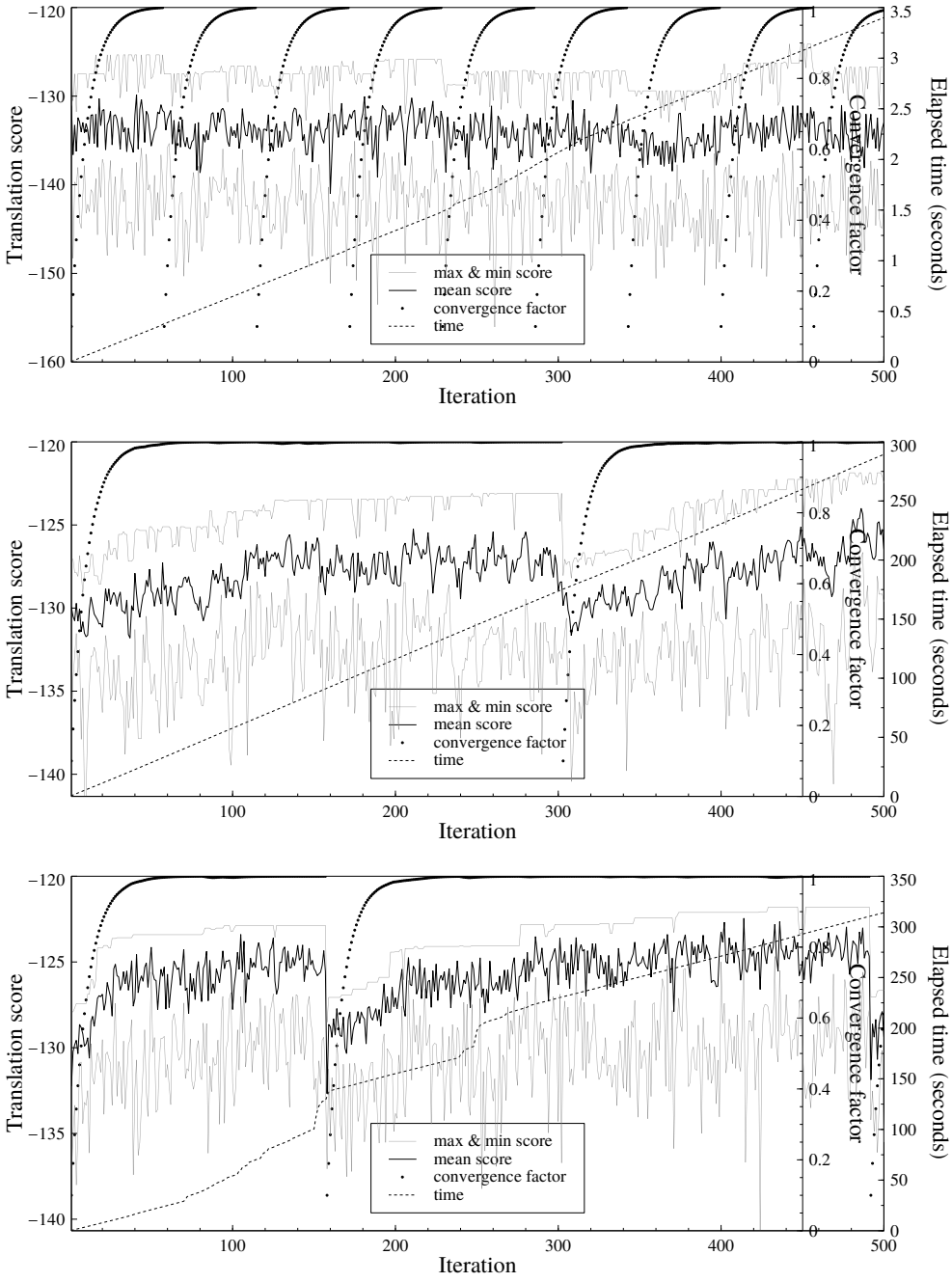

Figure A.9: Evolution of the score as a function of the iteration when decoding with ACODec (top), ACODec$^{\text{mono}}_{\text{indep}}$ (middle), and ACODec$^{\text{mono}}_{\text{share}}$ (bottom).