Multilingual Adaptive Text Simplification

Kim Cheng Sheang

TESI DOCTORAL UPF / 2023

THESIS SUPERVISOR
Prof. Horacio Saggion
Department of Information and Communication Technologies

For my wife Ratana.

Acknowledgments

I am immensely grateful for all the guidance and support I have received from a number of people who, in some way or another, were involved and made this thesis happen.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Horacio Saggion, for his generous support and guidance throughout my research. This thesis would not have been made possible without him. He is the most patient, calm, and responsive I have ever met. I am very grateful for his kindness and for always making time for me whenever needed.

I would like to give a special thanks to Luis Espinosa Anke, who shared with me about the PhD opportunity at UPF and got me connected with Prof. Horacio Saggion.

I am also very grateful to the researchers who collaborated with me on the work presented in this thesis, Sanja Štajner and Daniel Ferrés, for believing in me. I have learned so much from working with them.

Furthermore, I would like to thank all the members of the TALN group, especially Leo Wanner, for leading this awesome group. Thanks to Mónica Dominguez for always organizing seminars, talks, and events that helped advance our research and kept us connected. Thanks to Ahmed Abura'ed, Alèx Bravo, and Pablo Accuosto for their help and advice. Thanks to current and previous TALN members for all their works and inspirations: Roberto Carlini, Juan Soler Company, Aleksandr Shvets, Jens Grivolla, Simone Mille, Francesco Barbieri, Seda Mut, Georgia Cistola, Alex Peiró-Lilja, Laura Pérez-Mayos, Paula Fortuna, Guillermo Cambara Ruiz, Joan Codina, Hamdi Alp Öktem, Luis Chiruzzo, Santiago Egea, Piotr Przybyła, Euan McGill, Alba Táboas, Nicolau Duran Silva, Kemalcan Bora, and others that I could not list them all here.

In addition, I would like to acknowledge the support and assistance provided by the people at the UPF. Their administrative support, access to resources, and technical assistance have been facilitating the smooth progress of my research. Especially thanks to Lydia Garcia for being in charge of all my administrative work and giving instant support to all my

problems. Thanks to Montse Brillas and Jana Safrankova for preparing and helping with all the paperwork for my NIE renewal every year.

In early 2022, I had the opportunity to do a research visit for three months at Lille University, France. I want to thank Natalia Grabar and Anaïs Koptient for hosting me and giving me a warm welcome. Despite the pandemic lockdown, we were able to meet from time to time, which led to our collaboration on the work of complex word identification for French medical texts.

I would also like to express my gratitude to the members of the thesis evaluation board, Nuria Gala, Mireia Farrús, and Itziar Gonzalez-Dios. I am truly honored that they have agreed to read and evaluate my thesis.

Lastly, I would like to thank my family and friends for their encouragement and support. Their love, understanding, and belief in my abilities have constantly motivated and inspired me. Their presence has provided me with the motivation and stability necessary to take on the challenges of doctoral research.

Abstract

Reading is an essential skill that plays a crucial role in our daily lives. It allows us to access information, gain knowledge, expand our understanding of the world around us, and build the foundation for learning, communication, and personal growth. However, many texts we encounter day after day often contain complex words or syntactic structures that can cause reading difficulties for certain groups of people; this motivates the need for Automatic Text Simplification (ATS). ATS is a Natural Language Processing (NLP) task that aims to reduce the linguistic complexity of a text while preserving its original information and meaning. It involves various operations, such as replacing complex words with simpler synonyms, splitting long sentences into shorter ones, and reorganizing the structure of the text. The goal of ATS is to make texts more accessible and understandable to a broader audience, including non-native speakers, children, and individuals with Dyslexia, Autism, Aphasia, Intellectual Disabilities, and Deaf and Hard of Hearing. In this work, we will discuss our proposed methods for Complex Word Identification (CWI), Lexical Simplification (LS), and Sentence Simplification (SS) in order to help improve reading comprehension. For CWI, we propose several systems based on different machine learning algorithms, such as Convolutional Neural Networks, CatBoost, and XGBoost with word embeddings and feature-engineered for identifying complex words in English, Spanish, German, and French texts. For LS, we propose two systems, monolingual English and multilingual system supporting English, Spanish, and Portuguese. For SS, we propose several systems to simplify English and Spanish texts. In both LS and SS, we explore the use of transfer learning and controllable mechanism, where the transfer learning help create the model that requires less amount of training data, and the controllable mechanism gives us the ability to adjust the outputs based on our preference, especially for different target audiences.

Resum

La lectura és una habilitat essencial que juga un paper crucial en la nostra vida quotidiana. La lectura ens permet accedir a la informació, adquirir coneixements, ampliar la nostra comprensió del món que ens envolta i construir les bases per a l'aprenentatge, la comunicació, i creixement personal. No obstant això, molts textos sovint contenen paraules complexes o estructures sintàctiques que poden provocar dificultats lectores per a determinats grups de persones; això motiva la necessitat de la simplificació automàtica de text (ATS). ATS es una tasca que pretén reduir la complexitat lingüística d'un text tot conservant la seva informació i significat originals. Implica diversos operacions, com ara substituir paraules complexes per sinònims més senzills, dividir les frases llargues en frases més curtes i reorganitzant l'estructura del text. L'objectiu d'ATS és fer que els textos siguin més accessibles i entenedors a un públic més ampli. En aquest treball, presentem nostra proposta de mètodes d'identificació de paraules complexes (CWI), simplificació lèxica (LS) i Simplificació de frases (SS) per tal de fer els textos més accessibles. Pel que fa la CWI, proposem diversos sistemes basats en algorismes d'aprenentatge automàtic, com ara xarxes neuronals de convolucions, "CatBoost" i "XGBoost" amb incrustacions de paraules i característiques dissenyades per identificar paraules complexes en anglès, espanyol, alemany i francès. Pel que fa la LS, proposem dos sistemes, un pel anglès i un multilingüe. Per a la SS, explorem l'ús de l'aprenentatge de transferència i el mecanismes de control, on l'aprenentatge de transferència ajuda a crear un model que requereix menys quantitat de dades d'entrenament mentre que el mecanisme de control ens dona la capacitat per ajustar les sortides en funció de la nostra preferència, especialment per a diferents públics objectiu.

Contents

List of Figures

List of Tables

Acronyms

ATS Automatic Text Simplification.

BERT Bidirectional Encoder Representations from Transformers.

BiLSTM Bidirectional Long Short-Term Memory.

CNN Convolutional Neural Networks.

CWI Complex Word Identification.

EW English Wikipedia.

GCN Graph Convolutional Network.

GRU Gated Recurrent Unit.

LS Lexical Simplification.

LSTM Long-Short Term Memory.

MLM Masked Language Model.

MT Machine Translation.

NLP Natural Language Processing.

NMT Neural Machine Translation.

PBMT Phrase-Based Machine Translation.

POS Part-of-Speech.

ReLU Rectified Linear Unit.

RNN Recurrent Neural Network.

SEW Simple English Wikipedia.

SMT Statistical Machine Translation.

SS Sentence Simplification.

SVM Support Vector Machine.

T5 Text-to-Text Transfer Transformer.

TF-IDF Term Frequency - Inverse Document Frequency.

WSD Word Sense Disambiguation.

Chapter 1

Introduction

Our research focuses on the topic of Automatic Text Simplification. In this chapter, we detail the problem statement, motivation, and objectives that inspired us to take on this research to accomplish our goals. Next, we describe our contributions and the organization of the thesis.

1.1 Motivation

Reading is an essential skill that plays a crucial role in our daily lives. It allows us to access information, gain knowledge, expand our understanding of the world around us, and build the foundation for learning, communication, and personal growth. More importantly, it is a key ingredient for success in school, work, and life in general. Reading is an activity that enhances knowledge transfer and increases skill, which ultimately leads to academic success (Rabia et al., 2017). Reading has been shown to have numerous benefits for our mental health, including reducing stress, improving cognitive function, and enhancing empathy (Deepti S, 2016). It can reduce stress levels and improve mental health by providing an escape from reality and promoting relaxation. Reading can broaden one's perspective and understanding of the world by exposing the readers to different cultures, ideas, and experiences (Stanovich, 2009). It can also improve empathy and emotional intelligence by allowing readers to

connect with characters and their experiences (Adrian et al., 2005). It has also been found to enhance cognitive abilities such as critical thinking, problem-solving, and creativity, as well as improve vocabulary, language skills, and communication abilities (Cartwright, 2007).

Now imagine what life would be like if the text we read was so hard to understand that we had to constantly use additional resources to grasp the text's essential meaning; life would be hard and limited. Reading difficulties can have various side effects on individuals. One of the most significant side effects is the difficulty comprehending written information, which can lead to a lack of access to important information, especially in today's digital age, where most information is presented in written form. Vast amounts of text produced every day are not accessible to everybody due to their complexity. Usually, the way text is written often contains both lexical and syntactical complexity, especially for those who have problems reading and understanding. One of the ways to solve the problem is to adapt the texts by simplifying them with Automatic Text Simplification (ATS) (Saggion, 2017). ATS aims to reduce the complexity of a text while preserving its original meaning. Research on ATS has gained momentum in the last few decades because of its benefits as a tool for reading aids, which could make the information more accessible to broader audiences (Saggion, 2017) or help improve the performance of other NLP tasks. ATS has been shown useful for developing reading aids for children (Watanabe and Iwasaki, 2009; Siddharthan, 2002), non-native speakers (Siddharthan, 2002), people with cognitive disabilities such as autism (Orăsan et al., 2013; Barbu et al., 2015), aphasia (Carroll et al., 1999) or dyslexia (Rello et al., 2013a; Matausch and Peböck, 2010). Moreover, ATS can also be used as a preprocessing step to improve the results of many NLP tasks, e.g., parsing (Chandrasekar et al., 1996), information extraction (Jonnalagadda and Gonzalez, 2010; Evans, 2011), semantic role labeling (Vickrey and Koller, 2008), question generation (Bernhard et al., 2012), text summarisation (Siddharthan et al., 2004), and machine translation (Štajner and Popovic, 2016).

Text simplification can be performed at different levels, including lexical, syntactic, and semantic. Lexical simplification involves replacing

complex words with simpler synonyms or explanations. Syntactic simplification involves modifying the sentence structure and grammar to make it easier to understand. Semantic simplification focuses on simplifying the meaning and content of the text, such as ambiguous or metaphorical expressions, idiomatic phrases, and domain-specific terminology. The goal of text simplification is to reduce the linguistic complexity of a text and make it more accessible and comprehensible.

1.2 Research Questions

The research of this thesis started with the following research questions:

- **RQ1** Is it possible to employ a deep learning method with word embeddings and engineered features to accurately identify lexical sources of complexity in sentences?
- **RQ2** Can we build an adaptive lexical simplification?
- **RQ3** Can we build an adaptive sentence simplification?
- **RQ4** Can transfer-learning methods be used to improve the performance of text simplification?

1.3 Contributions

This thesis makes a number of contributions to the field of automatic text simplification, including complex word identification, lexical simplification, and sentence simplification. Our main contributions can be narrowed down to the following:

- Complex word identification is one of the most important modules in lexical simplification, which is used to find difficult words that should be simplified. For this task, we have proposed two systems. The first system is based on Convolutional Neural Networks (CNN)

with engineered features and word embeddings to identify complex words in English, Spanish, and German texts. The second system is made for identifying complex words in French medical texts, which has more features than the first system and was trained with different algorithms such as CNN, XGBoost, and CatBoost.

- For the lexical simplification task, we have proposed two state-of-the-art models. The first model is a monolingual controllable lexical simplification for English, whereas the second model is a multilingual controllable lexical simplification for English, Spanish, and Portuguese. Both models are controllable, meaning that the outputs can be altered based on the token values embedded into each input sentence to match our desire, which for the evaluation purposes the control tokens are set to their optimal values.

- We proposed several state-of-the-art sentence simplification systems for English and Spanish, using Transformer-based models coupled with a simplification control mechanism. The approach is inspired by the knowledge-transfer idea, where the model has been trained on data-rich tasks or languages and then fine-tuned with specific data to perform a certain task. Therefore, we fine-tuned our systems with Transformer-based pre-trained models along with control tokens embedded into each input. The control tokens are intended to have control over different aspects of the outputs, such as word length, sentence length, amount of paraphrasing, lexical complexity, and syntactic complexity.

- This thesis follows an open science mandate by making all resources created available.[1,2,3,4,5,6]

[1] www.github.com/kimchengsheang/cwi_cnn
[2] www.github.com/kimchengsheang/MedCWI
[3] www.github.com/kimchengsheang/ConLS
[4] www.github.com/kimchengsheang/mTLS
[5] www.github.com/kimchengsheang/TS_T5
[6] www.github.com/kimchengsheang/TS-AAAI-2022

1.3.1 Publications

The following are the outcomes of this research:

- **Sheang, Kim Cheng**, and Horacio Saggion. 2023. "Multilingual Controllable Transformer-Based Lexical Simplification." In 39th International Conference of the Spanish Society for Natural Language Processing. Jaén, Spain.

  Contributions:

  - Kim Cheng Sheang: ideas, models' implementation, experiments, evaluations, and writing.
  - Horacio Saggion: ideas, supervision, review, and editing.

- **Sheang, Kim Cheng**, Daniel Ferrés, and Horacio Saggion. 2022. "Controllable Lexical Simplification for English." In Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), 199–206. Abu Dhabi, United Arab Emirates (Virtual): Association for Computational Linguistics.

  Contributions:

  - Kim Cheng Sheang: ideas, model's implementation, experiments, evaluations, and writing.
  - Daniel Ferrés: ideas and writing.
  - Horacio Saggion: ideas, supervision, review, and editing.

- Štajner, Sanja, **Kim Cheng Sheang**, and Horacio Saggion. 2022. "Sentence Simplification Capabilities of Transfer-Based Models." In Proceedings of the AAAI Conference on Artificial Intelligence, 36:12172–80. British Columbia, Canada (Virtual).

  Contributions:

  - Sanja Štajner: ideas, expert guidelines, expert evaluations, writing.

  - Kim Cheng Sheang: ideas, models' implementation, experiments, automatic evaluations, crowd-sourced human evaluations, and writing.
  - Horacio Saggion: ideas, supervision, review, and editing.

- **Sheang, Kim Cheng**, Anaïs Koptient, Natalia Grabar, and Horacio Saggion. 2022. "Identification of Complex Words and Passages in Medical Documents in French." In Actes de La 29e Conférence Sur Le Traitement Automatique Des Langues Naturelles. Volume 1 : Conférence Principale, 116–25. Avignon, France: ATALA.

  Contributions:

  - Kim Cheng Sheang: ideas, models' implementation, experiments, evaluations, and writing.
  - Anaïs Koptient: ideas, dataset creation, writing.
  - Natalia Grabar: ideas, writing, supervision.
  - Horacio Saggion: supervision, review, and editing.

- **Sheang, Kim Cheng**, and Horacio Saggion. 2021. "Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer." In Proceedings of the 14th International Conference on Natural Language Generation, 341–52. Aberdeen, Scotland, UK: Association for Computational Linguistics.

  Contributions:

  - Kim Cheng Sheang: ideas, model's implementation, experiments, evaluations, and writing.
  - Daniel Ferrés: ideas and writing.
  - Horacio Saggion: ideas, supervision, review, and editing.

- **Sheang, Kim Cheng**. 2019. "Multilingual Complex Word Identification: Convolutional Neural Networks with Morphological and

Linguistic Features." In Proceedings of the Student Research Workshop Associated with RANLP 2019, 83–89. Varna, Bulgaria: Incoma Ltd.. (**Best Paper Award**)

Contributions:

- Kim Cheng Sheang: ideas, model's implementation, experiments, evaluations, and writing.
- Horacio Saggion: ideas, supervision, review, and editing.

- **Sheang, Kim Cheng**. 2019. "Context-Aware Automatic Text Simplification." In Proceedings of the Doctoral Symposium of the XXXV International Conference of the Spanish Society for Natural Language Processing (SEPLN 2019), 56–62. Bilbao, Spain: CEUR.

Other Publications

- Saggion, Horacio, Sanja Štajner, Daniel Ferrés, **Kim Cheng Sheang**, Matthew Shardlow, Kai North, and Marcos Zampieri. 2022. "Findings of the TSAR-2022 Shared Task on Multilingual Lexical Simplification." In Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), 271–83. Abu Dhabi, United Arab Emirates (Virtual): Association for Computational Linguistics.

1.4 Thesis Structure

This thesis is divided into five parts. Each part contains a series of chapters covering different studies.

Part I (chapters 2 and 3) gives the background relevant to the thesis. Chapter 2 provides a detailed study of the previous approaches of ATS, and Chapter 3 introduces the various target audiences that could benefit from ATS, their characteristics, and the challenges they face.

Part II (chapters 4) presents two of our experiments on CWI. The first experiment, Section 4.3, describes our CWI model based on CNN with features engineered and word embeddings to identify complex words in English, Spanish, and German texts. The second experiment, Section 4.4, describes the extended work of the previous experiment incorporating more features as well as training with different types of Machine Learning algorithms, including CNN, CatBoost, and XGBoost to tackle the complex word identification problem in French biomedical documents.

The research presented in this chapter is based on the following papers:

- **Sheang, Kim Cheng**. 2019. "Multilingual Complex Word Identification: Convolutional Neural Networks with Morphological and Linguistic Features." In Proceedings of the Student Research Workshop Associated with RANLP 2019, 83–89. Varna, Bulgaria: Incoma Ltd..

- **Sheang, Kim Cheng**, Anaïs Koptient, Natalia Grabar, and Horacio Saggion. 2022. "Identification of Complex Words and Passages in Medical Documents in French." In Actes de La 29e Conférence Sur Le Traitement Automatique Des Langues Naturelles. Volume 1 : Conférence Principale, 116–25. Avignon, France: ATALA.

Part III (chapters 5 and 6) presents our lexical simplification approaches. Chapter 5 describes the controllable lexical simplification model for English. Chapter 6 describes our approaches to tackling both monolingual and multilingual lexical simplification.

The research presented in this chapter is based on the following papers:

- **Sheang, Kim Cheng**, Daniel Ferrés, and Horacio Saggion. 2022. "Controllable Lexical Simplification for English." In Proceedings

of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022), 199–206. Abu Dhabi, United Arab Emirates (Virtual): Association for Computational Linguistics.

- **Sheang, Kim Cheng**, and Horacio Saggion. 2023. "Multilingual Controllable Transformer-Based Lexical Simplification." In 39th International Conference of the Spanish Society for Natural Language Processing. Jaén, Spain.

Part IV (chapters 7 and 8) presents our sentence simplification approaches. Chapter 7 describes our controllable sentence simplification model fine-tuned with T5 pre-trained model and control tokens. Chapter 8 describes the extended work of the previous model by fine-tuning a new model for Spanish with the Newsela dataset.

The research presented in this chapter is based on the following papers:

- **Sheang, Kim Cheng**, and Horacio Saggion. 2021. "Controllable Sentence Simplification with a Unified Text-to-Text Transfer Transformer." In Proceedings of the 14th International Conference on Natural Language Generation, 341–52. Aberdeen, Scotland, UK: Association for Computational Linguistics.

- Štajner, Sanja, **Kim Cheng Sheang**, and Horacio Saggion. 2022. "Sentence Simplification Capabilities of Transfer-Based Models." In Proceedings of the AAAI Conference on Artificial Intelligence, 36:12172–80. British Columbia, Canada (Virtual).

Lastly, **Part V** summarizes all the work of this thesis and gives some potential future work.

Part I

Background

Chapter 2

Literature Review

Early approaches to text simplification were based on manual intervention, where human editors would manually simplify the text (Siddharthan, 2014). However, manual text simplification is time-consuming and labor-intensive, which has led to the development of automated text simplification systems (Siddharthan, 2014). Automatic text simplification in the field of NLP dates back to the 1990s from the work of Chandrasekar and Srinivas (1997), which is based on Chandrasekar et al. (1996), highlighting that long and complicated sentences can pose significant challenges for various systems that rely on natural language input.

Generally, simplification tasks are separated into two categories: lexical simplification and higher-level simplification (sentence-level or document-level). Lexical simplification aims to reduce the complexity of a text by replacing complex words with simpler synonyms or more commonly used words with the same meaning (Horn et al., 2014). The following sentence shows a lexical simplification example where the word **trencherous** is selected and a complex word is replaced by the word **dangerous** [1].

[1] Example from Sian Gooding presented at Cambridge Assessment 2019 https://drive.google.com/file/d/1xt0C0diASUs-HY-bZJ16zOQ5uF7y0Fym/view

Snow has left many roads **treacherous**.
Snow has left many roads **dangerous**.

Higher-level simplification combines lexical, syntactic, and/or semantic simplification to achieve more comprehensive simplification (Siddharthan, 2006). Syntactic simplification is a task in text simplification that focuses on reducing the complexity of sentence structures (Siddharthan, 2006). The approach involves modifying the syntactic structure of a sentence to make it easier to understand while preserving the original meaning. It involves various operations, such as splitting long and complex sentences into short and simpler ones, removing or rephrasing subordinate and embedded clauses, and simplifying coordination and subordination. The following sentence contains ambiguity, which can be fixed with syntactic simplification [2].

The horse raced past the barn fell.
The horse raced past the barn. The horse fell.

Early approaches to syntactic simplification have often relied on handcrafted rules to capture syntactic transformations; however, more recent approaches have explored the use of neural networks and deep learning methods to automatically learn syntactic simplification patterns from large corpora (Alva-Manchego et al., 2020b). Syntactic simplification is an essential component of text simplification as it helps to break down complex sentences into more manageable and understandable units (Siddharthan, 2006), whereas semantic simplification aims to reduce the complexity of the meaning conveyed in a text while preserving the overall message. The approach involves identifying and simplifying complex semantic structures, such as ambiguous or metaphorical expressions, idiomatic phrases, and domain-specific terminology. Semantic simplification helps to bridge the gap between complex and simplified texts by

[2]Example from Sian Gooding presented at Cambridge Assessment 2019.

ensuring that the simplified version retains the essential meaning of the original text, and so far there is very little research has focused on this.

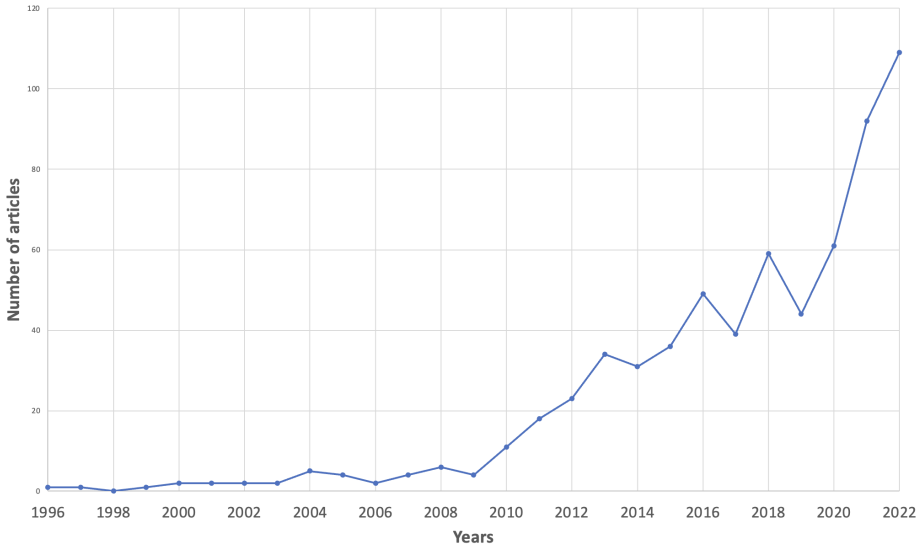


Figure 2.1: The graph shows the number of ATS articles published between 1996 and 2022.

The research on ATS has been getting more interest starting from the late 90s based on the number of publications in Figure 2.1, which shows the number of ATS articles published between 1996 and 2022. The graph is created by counting the search results on Google Scholar with the keywords: “text simplification”, “lexical simplification”, “syntactic simplification”, or “sentence simplification”.

2.1 Lexical Simplification

Carroll et al. (1998) proposed the first lexical simplification system based on the method of Devlin and Tait (1998) in the context of Aphasic readers. The system selects a list of synonyms from WordNet (Miller, 1994) and then chooses the one with the highest Kucera-Francis frequency as

obtained from the Oxford Psycholinguistic Database (Quinlan, 1992). Similar approaches of lexical simplification were also proposed by Lal and Rüger (2002) as a component of a text summarization pipeline and by De Belder et al. (2010) that employed Word Sense Disambiguation (WSD) instead of the Psycholinguistic Database, making it more scalable to other languages.

There are different approaches to lexical simplification have been proposed; therefore, we group them into different categories, such as rule-based, data-driven, unsupervised, and masked language modeling. More details are described in the following sections:

2.1.1 Data-Driven Approaches

One of the early data-driven approaches was proposed by Yatskar et al. (2010) that utilized edit histories from English Wikipedia (EW) and Simple English Wikipedia (SEW) to extract possible synonym pairs from the old and updated versions of the articles. They calculated the probability of different edit operations to identify which phrase from the edit history has been replaced with the aim of finding the simpler version, and the editors' metadata was also used to detect the trusted revisions.

Biran et al. (2011) proposed a system for learning simplification rules from EW and SEW without using the edit histories. The system consists of two phases: rule extraction and sentence simplification. In the rule extraction phase, the system extracts ordered word pairs from the EW and SEW and computes a similarity score between the words using their context vectors (cosine similarity) to ensure that the extracted pairs represent a complex-simple pair. WordNet is also used as a semantic filter for possible lexical substitution along with word complexity measure, which takes into account word length and word frequency. In the simplification phase, the system selects and simplifies words in a sentence based on the information received from the previous phase.

Horn et al. (2014) proposed a feature-based ranker approach. First, they extracted over 30,000 candidate lexical simplifications (the synonyms of each word) from a sentence-aligned corpus of EW and SEW us-

ing GIZA++ (Och and Ney, 2003). By identifying aligned words in these two versions of Wikipedia, they were able to identify potential simplifications. To apply these simplification rules, they trained a feature-based ranker model with Support Vector Machine (SVM) on a set of labeled simplifications. These labeled simplifications were collected using Amazon's Mechanical Turk, a crowd-sourcing platform. This training process allows the ranker to learn the patterns and features that are indicative of good simplifications.

Further subsequent developments of lexical simplification mostly followed the well-known four-step pipeline approach proposed by Shardlow (2014) (Paetzold and Specia, 2017; Al-Thanyyan and Azmi, 2021; Saggion et al., 2022; North et al., 2023). It consists of four modules: 1) Complex Word Identification (CWI) for detecting complex words, 2) Substitution Generation (SG) to generate candidates for replacement, 3) Substitution Selection (SS) for filtering candidates, and 4) Substitution Ranking (SR) for ranking candidates by simplicity.

2.1.2 Unsupervised Approaches

Traditionally, lexical simplification methods relied on manually created resources such as annotated data, dictionaries, WordNet, or Psycholinguistic Database. However, creating these resources is difficult, expensive, and time-consuming; therefore, different unsupervised approaches have been proposed. Glavaš and Štajner (2015) proposed an unsupervised lexical simplification system that relied on word vector representations. The method leverages the power of GloVe word embeddings (Pennington et al., 2014) to identify simpler alternatives for complex words without the need for manually annotated data. First, the model extracts the 10 most similar candidates based on cosine similarity. Then, the model ranks candidates by different features such as semantic similarity, context similarity, simplicity (word frequency), and language model features to make sure that the selected candidates fit into the context.

Paetzold and Specia (2016e) proposed a new context-aware model for word embedding to generate candidates for complex words. The model

was trained using CBOW Word2Vec model (Mikolov et al., 2013) on a corpus that was annotated with POS tags, such as adjectives, nouns, verbs, and adverbs. Given the vector representation of the target word and the POS tag, the model extracts n candidates with the closest cosine similarity. Then, a binary classifier was trained with features such as language model log probabilities, word-embedding cosine similarity, and the conditional probability of a candidate given the POS of the target word to filter the candidates. For candidate ranking, a 5-gram language model was trained with various datasets, and then the candidates were ranked by their uni-gram probabilities. The approach leverages two key resources: a corpus of subtitles and a word-embedding model that considers the ambiguity of words. The use of subtitles as a resource is motivated by the fact that they often contain simplified language to aid comprehension for viewers. Despite being an unsupervised model, the proposed method still compromises the use of annotated data for POS.

2.1.3 Masked Language Modeling Approaches

Masked Language Model (MLM) has become a popular approach in Natural Language Processing (NLP) tasks. MLM is trained on a large amount of data to predict masked words in a given sentence, which allows an effective encapsulation of the contextualized word representations. The concept of masked language modeling was first introduced with BERT (Bidirectional Encoder Representations from Transformers) (Devlin et al., 2019), which has become one of the most influential and widely used pre-trained language models. Prior to BERT, most substitution generation approaches relied on rules, dictionaries, WordNet, statistics, n-grams, and word embeddings such as GloVe, Word2Vec, or FastText (Bojanowski et al., 2017).

The use of MLM for lexical simplification, called LSBert, was first introduced by Qiang et al. (2020, 2021). The model employs two phases: 1) substitution generation and 2) substitution filtering and ranking. First, the model extracts substitution candidates from BERT pre-trained model by providing the sentence with the complex word masked and the orig-

inal sentence concatenated with it as a context. Then, all candidates are filtered and ranked by five features such as MLM prediction probabilities, language model, PPDB database (Ganitkevitch et al., 2013), corpus-based word frequency, and FastText word similarity.

Since the release of LSBert, MLM approach for lexical simplification has been widely adopted by subsequent research. For example, 6 out of 10 system reports submitted to the TSAR-2022 shared task (Saggion et al., 2022) are based on MLM (Whistely et al., 2022; Chersoni and Hsu, 2022; Wilkens et al., 2022; Nikita and Rajpoot, 2022; Li et al., 2022; North et al., 2022).

2.2 Higher-level Simplification

Since the inception of Automatic Text Simplification (ATS), researchers have focused on both lexical and beyond lexical aspects simultaneously. As the lexical part has been discussed above, so in this section, we will focus on sentence-level simplification. The research on simplification has evolved over time, with different approaches and techniques being developed to simplify texts. Initially, the approaches involved the use of handcrafted rules to perform syntactic simplification (Chandrasekar et al., 1996; Chandrasekar and Srinivas, 1997; Siddharthan, 2002, 2006, 2010, 2011; Bott et al., 2012). Here are some common operations: splitting sentences, simplifying relative clauses, simplifying appositive phrases, simplifying coordination, rearrangement and clause dropping, removing inessential phrases, converting passive sentences to active, and removing subordinate and embedded clauses. The use of handcrafted rules in syntactic simplification models has its limitations. While these approaches can encode precise and linguistically well-informed syntactic transformations, they do not account for lexical simplifications and their interaction with the sentential context (Narayan and Gardent, 2016).

2.2.1 Simplification as Monolingual Translation

Simplification as monolingual translation refers to the process of transforming a complex text into a simpler and more easily understandable version within the same language. This task is often approached using techniques similar to those used in Machine Translation (MT), where a source text is translated into a target text in a different language. However, in the case of simplification, the translation is performed within the same language, aiming to make the text more accessible to a specific audience or to reduce its complexity. The earliest contribution that put the research into this direct was from Zhu et al. (2010). They proposed a translation model based on parse tree transformations, which reduced the need for lexical-level translation operations in monolingual translation. This work laid the foundation for subsequent developments in the simplification-as-translation area. Later advancements in this field focused on the Statistical Machine Translation (SMT) paradigm and transitioned to neural-based methods.

Statistical Machine Translation (SMT) SMT is a machine-learning approach to translating natural language. It has made significant advancements in translation quality since the introduction of Phrase-Based Machine Translation (PBMT) (Marcu and Wong, 2002; Och and Ney, 2004). SMT treats translation as a statistical problem and uses statistical models to generate translations based on patterns and probabilities learned from large-scale parallel corpora. The models capture the statistical patterns and dependencies between words and phrases in the source and target languages, and the quality of translation in SMT usually depends on the availability of parallel data between the source and target language pair (Haffari et al., 2009). SMT has been widely adopted as a framework for text simplification such as Specia (2010) uses standard SMT to simplify Portuguese texts, Coster and Kauchak (2011a) implemented a modified PBMT to incorporate phrasal deletion, making the model produces shorter outputs, Wubben et al. (2012) focuses on dissimilarity to include diversity, as mentioned by Woodsend and Lapata (2011) that simplifica-

tion does not necessarily imply shortening. One of the main challenges in SMT for text simplification is the limited availability of high-quality and large-scale manually simplified corpora; therefore, Xu et al. (2016) propose an approach to use large-scale paraphrases learned from bilingual texts and a small amount of manual simplifications with multiple references. Another significant contribution is the release SARI metric, which has been used to evaluate syntactic simplification ever since. The use of SMT has been shown to work well for text simplification; however, it still has two limitations: one is its complex translation model, and the other is its poor handling of long-distance dependencies. These limitations have been addressed by the introduction of Neural Machine Translation (NMT) models, which have the ability to handle long-distance dependencies and have simplified modeling mechanisms (Li et al., 2015).

Neural Machine Translation (NMT) NMT uses a single large neural net that directly transforms the source sentence into the target sentence (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014a; Bahdanau et al., 2015). The models used in NMT typically consists of an encoder and a decoder. The encoder takes a variable-length input sentence and extracts a fixed-length representation from it. This representation is then used by the decoder to generate a correct translation (Cho et al., 2014a). NMT has a number of advantages over the existing SMT system. First, NMT requires a minimal set of domain knowledge. For example, the models proposed by Bahdanau et al. (2015), Sutskever et al. (2014), or Kalchbrenner and Blunsom (2013) do not assume any linguistic properties in both source and target sentences except that they are sequences of words. Second, the whole system is jointly trained to maximize the translation performance producing more fluent and grammatical output as well as capturing long-range dependencies, unlike the existing phrase-based system, which consists of many separately trained features whose weights are then tuned jointly. Lastly, the memory footprint of the NMT model is often much smaller than the existing system, which relies on maintaining large tables of phrase pairs.

Various researchers have successfully applied NMT for text simpli-

fication such as: 1) Nisioi et al. (2017) trained their model with Simple Wikipedia data compiled by Hwang et al. (2015), 2) Zhang and Lapata (2017) combined NMT with Deep Reinforcement Learning, and Scarton and Specia (2018) trained the model with school grade levels to simplify for specific target audiences.

2.2.2 Simplification as Sequence-Labeling

In this section, we review the existing literature on edit-based text simplification systems (sequence-labeling approaches), which focus on making simplifications through edit operations such as add, delete, and keep. These systems learn edit operations at the word level, allowing for fine-grained control over the simplification process, which provides advantages in terms of interpretability and adaptability.

Alva-Manchego et al. (2017) was the first to propose an edit-based method for text simplification using BiLSTM to predict edit labels sequentially. The model outperforms the translation-based models in terms of simplicity scores, although it may result in slightly lower meaning preservation and grammaticality.

The edit-based approach has become notable since the release of EditNTS proposed by Dong et al. (2019). EditNTS is a neural programmer-interpreter model that learns explicit edit operations (add, keep, and delete) in a sequential fashion. The model is an LSTM encoder-decoder model with an injection of POS tags as the syntactic information. The model addresses the lack of interpretability in previous approaches and allows for a meaningful explanation of the simplification process. EditNTS outperforms other state-of-the-art ATS systems in terms of simplicity, fluency, and adequacy.

Kumar et al. (2020) proposed an iterative, edit-based model based on RNN with GRU that performs word and phrase-level edits on complex sentences. The model is guided by a scoring function that considers fluency, simplicity, and meaning preservation. Unlike previous approaches, this model does not require a parallel training set.

Omelianchuk et al. (2021) proposed a method based on sequence tag-

ging, leveraging pre-trained Transformer-based encoders for simple and efficient text simplification. The model also employed data augmentation with Back Translation and knowledge distillation on ensemble teacher models to augment the training data.

Cumbicus-Pineda et al. (2021) proposed a method to enhance the edit-based text simplification system (the model is based on EditNTS) by incorporating syntactic information. The model relies on Graph Convolutional Network (GCN) module that mimics the dependency structure of the sentence, providing the model with an explicit representation of syntax. By incorporating syntactic information, the proposed model aims to capture long-range syntactic relations among words and improve the quality of simplification.

2.2.3 Transformer-Based Methods

Vaswani et al. (2017) proposed a new network architecture called Transformer, which relies solely on the use of self-attention, where the representation of a sequence (or sentence) is computed by relating different words in the same sequence. Unlike traditional sequence transduction models that rely on complex recurrent or convolutional neural networks, the Transformer eliminates the need for recurrence and convolutions entirely. Instead, it connects the encoder and decoder through attention mechanisms, resulting in a simpler and more efficient model. Another advantage of the Transformer is speed because it is parallelizable, meaning that a number of inputs can be passed through the network simultaneously, taking advantage of modern hardware (especially GPUs), whereas other recurrent networks like Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) or Gated Recurrent Unit (GRU) (Cho et al., 2014b) have to pass inputs sequentially. Originally, Transformer was designed for Machine Translation and has been widely applied in various NLP tasks; moreover, it has been successfully adopted in other domains, including Computer Vision, Audio Processing, and Multimodal Applications due to its flexible architecture (Lin et al., 2022). For NLP, Transformer has been explored in tasks such as Machine Translation (Vaswani

et al., 2017; Edunov et al., 2018; Raffel et al., 2020; Liu et al., 2020a), Text Summarization (Lewis et al., 2020; Raffel et al., 2020), Question Answering (Raffel et al., 2020), Text Classification (Howard and Ruder, 2018), Language Modeling (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020; Radford et al., 2019; Brown et al., 2020; Zhao et al., 2023), Named Entity Recognition (Ushio and Camacho-Collados, 2021; Yamada et al., 2020; Lu et al., 2022), and more (Lin et al., 2022).

Zhao et al. (2018) explored the model for sentence simplification that integrated the Transformer architecture and a custom loss function with Simple PPDB (A Paraphrase Database for Simplification) (Pavlick and Callison-Burch, 2016), an external paraphrase knowledge base for simplification that covers a wide range of real-world simplification rules. Moreover, the model also employed dynamic memory augmented method (Feng et al., 2017) to capture infrequent simplification rules. Martin et al. (2019) trained the Transformer model for controllable sentence simplification by embedding control tokens to each input in order to control different aspects of the outputs, such as the amount of compression, the amount of paraphrasing, lexical complexity, and syntactical complexity. Besides adopting the whole Transformer architecture, the encoder or decoder can be used separately, e.g., Omelianchuk et al. (2021) propose a sentence simplification model based on a sequence tagging that relied solely on encoders.

2.2.4 Controllable Simplification

Early work on controllable simplification was proposed by Scarton and Specia (2018), an MT-based model trained with control tokens (grade level, operation) embedded in the inputs in order to adjust the outputs for specific audiences based on their reading levels. Similarly, Martin et al. (2019) proposed a Transformer-based sequence-to-sequence model with four control tokens embedded in each input to control different aspects, such as compression level, amount of paraphrasing, and lexical and syntactic complexity.

Nishihara et al. (2019) argued that adding the grade-level token to the inputs only controls the syntactic complexity and tends to output difficult words beyond the target grade level; therefore, they proposed a method for controllable text simplification with lexical constraint loss. They introduced the concept of using the grade level of the US education system as the target level of the sentence. Their approach considered both the sentence level and the word level to achieve the desired level of simplification. The sentence level was incorporated by adding the target grade level as input, while the word level was considered by adding weights to the training loss based on words that frequently appear in sentences of the desired grade level. The results of their experiments showed improvements in both BLEU and SARI metrics.

Previous systems primarily rely on sequence-to-sequence models that are trained end-to-end to perform all these operations simultaneously. However, these systems often struggle to adapt to the specific requirements of different target audiences; hence, Maddela et al. (2021) proposed a hybrid approach to text simplification that combines linguistically-motivated rules for splitting and deletion with a neural paraphrasing model. This approach allows for the production of varied rewriting styles and provides more control over the degree of each simplification operation applied to the input texts.

In Chapter 7 and Chapter 8, we will report our works on this specific area.

2.3 Evaluation of Sentence Simplification Systems

Automatic sentence simplification systems are usually evaluated in two ways: (1) automatically, for the similarity of their output to the gold standard manual simplifications; and (2) manually, for grammaticality, simplicity, and meaning preservation of their output sentences.

For automatic evaluation, studies commonly use 'gold standard' manually simplified test sentences and calculate the BLEU (Papineni et al.,

2002) and SARI (Xu et al., 2016) scores. BLEU is originally designed for Machine Translation and has been commonly used previously. BLEU has lost its popularity in text simplification due to the fact that it correlates poorly with human judgments and often penalizes simpler sentences (Sulem et al., 2018). We keep using it so that we can compare our system with previous systems. SARI compares system outputs with the references and the source sentence. It measures the performance of text simplification on a lexical level by explicitly measuring the goodness of words that are added, deleted, and kept. Automatic evaluation is useful for quickly getting rough estimates of the performances of different system configurations. Nevertheless, although both scores show some correlations with human assessments (Štajner et al., 2014; Xu et al., 2015), they are not reliable enough for comparing performances of different simplification systems (Sulem et al., 2018; Vásquez-Rodríguez et al., 2021). Some studies also use Flesch-Kincaid Grade Level (FKGL) (Kincaid et al., 1975) for automatic evaluation. Although well-known in readability research, this metric is considered inadequate for sentence simplification (Saggion, 2017; Stajner, 2021; Tanprasert and Kauchak, 2021).

In the ideal scenario, grammaticality and meaning preservation should be evaluated by native speakers with high literacy levels, as the original sentences can be too complex to understand for an average reader. Simplicity, in contrast, should be evaluated by non-native speakers, experts in text simplification or production of easy-to-read texts, or carers of the intended target population (Stajner, 2021). All three evaluations are usually performed using a five-point Likert scale (Alva-Manchego et al., 2020b). Crowd-sourced human evaluation involves collecting judgments from a diverse group of individuals to assess the quality and fluency of the generated language. This type of evaluation often has the following: (1) in most of the studies, all three evaluations are performed by the same people, usually Amazon Mechanical Turk workers, whose literacy levels are unknown and who thus might not be the optimal evaluators of grammaticality and meaning preservation; (2) if the pool of evaluators is comprised of a mixture of native and non-native speakers, or people with different literacy levels, the notion of simplicity and grammaticality might differ

among them.

2.4 Summary

This chapter provides an overview of the history and different approaches to automatic text simplification. It begins by discussing the emergence of automatic text simplification and the motivation behind it. The chapter then focuses on lexical simplification, which involves simplifying the vocabulary of a text. It discusses various approaches to lexical simplification, including rule-based, data-driven, unsupervised, and masked language modeling. The chapter also highlights higher-level simplification, which involves simplifying the lexical and syntactic structure of a text. It discusses different methods of syntactic simplification, starting from rule-based approaches to more modern techniques such as monolingual machine translation, edit-based and Transformer-based methods.

Additionally, the division between lexical simplification and syntactic simplification is somehow artificial since syntax and lexicon are closely related. They both contribute to the overall structure and meaning of a sentence. The relationship between syntax and lexicon can be seen in the way that changes in one can affect the other. For example, when a complex word is replaced with a simpler word in lexical simplification, it may also require changes in the sentence structure to maintain grammaticality and coherence. Similarly, when a sentence is restructured in syntactic simplification, it may involve the substitution of certain words to maintain the intended meaning.

Finally, we end the chapter with a discussion of the evaluation methods for sentence simplification systems.

Chapter 3

Target Audiences

Text Simplification can provide numerous benefits for users with different needs, including those with Autism (Evans et al., 2014), Dyslexia (Rello et al., 2013b), Aphasia (Carroll et al., 1998), low literacy levels (Children and Second Language Learners) (Paetzold and Specia, 2016e), Deaf and Hard of Hearing (Inui et al., 2003), and Intellectual Disabilities (Chen et al., 2017). In the following, we will discuss the characteristics of each group and the effects on reading difficulties.

3.1 Autism

Autism, also known as Autism Spectrum Disorder (ASD), is a complex neurodevelopmental disorder that affects communication, social interaction, and behavior. It is a lifelong condition that typically appears in early childhood and affects individuals differently, with varying degrees of severity. The exact cause of Autism is not yet fully understood, but it is believed to be a combination of genetic and environmental factors (Halepoto and Al-Ayadhi, 2014).

The symptoms of Autism can vary widely but generally include impairments in social interaction, difficulty with communication, and restrictive and repetitive behaviors. Individuals with Autism may have difficulty understanding social cues, making eye contact, and engaging in con-

versation. They may also have difficulty with nonverbal communication, such as facial expressions and body language. Additionally, individuals with Autism may engage in repetitive behaviors, such as hand-flapping or rocking, and may have a strong attachment to routines and sameness (Bilbili, 2013).

Autism is a highly heritable condition, and genetic relatives of people with Autism often show milder expression of traits characteristic of Autism, referred to as the Broader Autism Phenotype (BAP) (Sucksmith et al., 2011). However, environmental factors may also play a role in the development of Autism. For example, some studies have suggested a possible link between certain prenatal and perinatal factors, such as maternal infection or exposure to certain chemicals, and an increased risk of Autism (Mohammed et al., 2022).

Diagnosis of Autism is typically based on a combination of behavioral and developmental assessments, as well as medical and family history. There are several different types of Autism, including Classical Autism or Autistic Disorder, Asperger's Syndrome, William's Syndrome, Pervasive Developmental Disorder (Not Otherwise Specified), Rett's Syndrome, Landau Kleffner Syndrome, and Children Dis-integrative Disorder (Ivo Paclt and Anna Strunecka, 2010). Each type of Autism has its own set of diagnostic criteria and specific symptoms.

There is currently no cure for Autism, but early intervention and treatment can help individuals with Autism to develop communication and social skills, manage their behavior, and improve their quality of life. Treatment options may include behavioral therapy, speech therapy, occupational therapy, and medication (Volkmar et al., 2004).

Children with autism are at risk of reading and learning difficulties (Vale et al., 2022). Research shows that children with Autism manifest abnormalities in the use of gaze and have difficulties in the comprehension of mental states (Johnson et al., 2007). Norbury and Nation (2011) investigated the reasons for variability in reading skills in people with Autism. They used the simple view of the reading model to investigate both word decoding and text comprehension processes in two well-established subtypes within the Autism spectrum. They found that reading outcomes in

ASD are related to variations both in decoding and comprehension and in the oral language skills that support the development of these processes.

There has been a growing interest in developing language technologies to make documents more accessible for individuals with autism. An early approach by Evans et al. (2014) focuses on evaluating a set of syntactic simplification rules designed to address the challenges faced by individuals with Autism when processing syntactically complex and compound sentences. The study developed 127 rules for simplifying complex sentences and 56 rules for simplifying compound sentences. The evaluation aimed to assess the accuracy of these rules and determine the reliability of the fully automatic conversion of sentences into a more accessible form.

Another research by Barbu et al. (2015) explores a simplification system to help Autism individuals. First, they discuss the linguistic limitations faced by individuals with Autism, mentioning that they often struggle with inferring the meaning of ambiguous words from context and linking objects in sentences with pronouns and anaphoric references. These difficulties become more pronounced as the syntactic complexity of a sentence increases. To address these limitations, Barbu et al. (2015) propose a simplification system called Open Book. Open Book employs strategies commonly used by successful simplifiers, such as image retrieval, document summarization, and document topic modeling. The system allows users to select words they do not comprehend and retrieve images that illustrate them. It also ranks sentences in a document according to their relevance and provides a summary of the document's content. Additionally, Open Book presents users with excerpts and expressions that best summarize the overall topic of a document using a topic modeling approach called Latent Dirichlet Allocation (Blei et al., 2003). To evaluate the effectiveness of the system, they conducted a study with 243 autistic patients. The participants were asked to read both the original and simplified versions of various documents and then answer a questionnaire about the content. The researchers claim that subjects who read the simplified versions of the documents achieved noticeably higher comprehensibility scores compared to those who read the documents in their original form.

Yaneva et al. (2015) conducted an eye-tracking study to investigate how simplification could aid individuals with Autism. They compared the fixation times of subjects while reading various documents and found that individuals with Autism tend to fixate on photographs and images more than non-autistic individuals. The findings suggest that adding visual components that describe key components in a text can help individuals with Autism comprehend its meaning. Additionally, they found that modifying texts according to the Plain English guidelines of Freyhoff et al. (1998a) resulted in higher comprehension. However, the study also discovered that replacing the text with ads and/or images that are not descriptive can hinder their comprehension.

3.2 Dyslexia

Reading difficulties, also known as Dyslexia, are characterized by the inability to read words accurately and fluently. It is a specific learning disability that affects reading skills, including difficulties in word recognition, decoding, and spelling. It was first reported in 1896 by a physician, W. Pringle Morgan, and since then, major medical journals have continued to publish research furthering the scientific understanding of dyslexia (Romberg et al., 2016). Dyslexia is a complex disorder that varies from person to person, and it is estimated to affect between 5 and 10% of the worldwide population (Al-Shidhani and Arora, 2012). According to Roitsch and Watson (2019) and Yunus and Ahmad (2022), common characteristics of dyslexia include difficulty with phonological skills, low accuracy, and fluency of reading, poor spelling, and/or rapid visual-verbal responding. Additionally, dyslexia is often associated with poor handwriting, written expression difficulties, and difficulty associating sounds with letters (Yunus and Ahmad, 2022).

Dyslexia is a learning disorder that affects a person's ability to read, write, and spell. Their reading challenges often include long and less-frequent words (Rello et al., 2013b), homophones words that are orthographically similar, new words, and non-words (Rello et al., 2013a).

Dyslexia is not limited to reading difficulties with alphabetic material. Cornoldi et al. (2022) found that dyslexia is specifically related to difficulties in reading and writing not only alphabetic material but also numerical material. Furthermore, dyslexia is not limited to reading difficulties. According to Al-Dawsari and Hendley (2022), there are non-reading difficulties that co-occur with dyslexia, such as memory problems and low levels of self-esteem.

The identification of dyslexia at a preliminary phase comes from the ability to notice dyslexia characteristics. Yunus and Ahmad (2022) suggest that poor handwriting, written expression difficulties, spelling difficulties, reading fluency, and difficulty associating sounds with letters are some of the characteristics that can help identify dyslexia. Additionally, reading difficulties in most children with developmental dyslexia are related to phonological disorders (Kurinna et al., 2022).

Dyslexia is a specific learning disability that affects a significant number of individuals. According to Shaywitz et al. (1992), reading difficulties, including dyslexia, occur as part of a continuum that also includes normal reading ability. Dyslexia is not a result of low intelligence or lack of motivation. It is a result of neurobiological differences that affect the way the brain processes language (Irdamurni et al., 2018).

Several studies have explored the benefits of text simplification techniques for individuals with dyslexia, focusing on lexical simplification, visual support, and text presentation parameters. In a study by Rello et al. (2013b), they measured reading time and comprehension using eye-tracking and questionnaires and found that a system that presented synonyms on demand was preferred over a system that automatically replaced difficult words. The study also found that using more frequent words improves reading speed, and shorter words help them understand the text better. These results suggest that interactive tools that perform lexical simplification may benefit people with dyslexia. Similar results have also been found by Gala and Ziegler (2016). The study involved testing the reading performance of dyslexic children on original and manually simplified texts, as well as assessing their comprehension through multiple-choice questions. The results showed that the simplified texts

led to increasing reading speed, reduced reading errors (particularly lexical ones), and did not result in a loss of comprehension. Similarly, examined the influence of visual support and lexical simplification on sentence processing through eye movements (Rivero-Contreras et al., 2021). The study found that visual support and lexical simplification were effective in facilitating sentence processing, particularly by enhancing lexical semantic access. The study also found that participants with lower print exposure and lower vocabulary benefited more from word-level lexical simplification.

3.3 Aphasia

Aphasia is a language disorder that occurs due to damage to a specific area of the brain that controls language expression and comprehension (Elias et al., 2023). It affects multiple modalities of language use, including reading, auditory comprehension, and expressive language (Ingram et al., 2020). Aphasia can lead to social isolation and loss of social roles (Żulewska-Wrzosek, 2021). There are different types of aphasia, including Broca (Non-Fluent), Wernicke (Fluent), Anomic (Dysnomia), and Global (Fazeli et al., 2008). The most widely accepted neurologic and/or neuropsychological definition of aphasia is the loss of ability to use speech or to understand speech as the result of disease or injury affecting the brain (Kumar et al., 2017). Aphasia is linked to impairments in the lexical/semantic and grammatical systems of language, which are associated with Wernicke-type Aphasia and Broca-type Aphasia, respectively (Ardila, 2010).

Reading difficulties in aphasia can have a significant impact on everyday activities and social interactions. People with aphasia may struggle to read stories to children, read emails, or participate in conversations about newspaper articles. The increasing prevalence of technology-based written communication further exacerbates the communication gap between people with aphasia and the rest of the world (Caute et al., 2015).

Different types of aphasia can also affect reading comprehension dif-

ferently. People with non-fluent aphasia often have difficulty understanding complex sentences, especially those that do not follow the typical word order for their language. In contrast, people with fluent aphasia may have difficulty producing semantically specific words, which can impact their reading comprehension (Gordon, 2008). Other common reading difficulties include high information density, long sentences, long sequences of adjectives, passive voice, and compound nouns (Carroll et al., 1999).

Technological advancements have provided alternative ways to compensate for reading difficulties associated with aphasia. A review by Cistola et al. (2021) covered research on technologies explicitly developed to compensate for reading difficulties associated with aphasia and research into which accessibility features included in mainstream high-tech systems are helpful for people with aphasia when trying to access written material.

3.4 Children

Children's ability to read and comprehend text is a crucial aspect of their personal and academic development. However, not all children have the same level of proficiency in reading, and some may require additional help to facilitate their learning. One approach is to have adapted text by using text simplification to simplify texts based on children's reading levels. E.g., controlling vocabulary, sentence length, and the complexity of sentence form, etc. The simplified texts should not be too simple, as the reading encourages the learning of new words, and not be too hard to make them feel comfortable and more engaged in the reading. This could help children become more independent in their reading and learning, making them rely less on others for help and feel more confident in their abilities.

One area of focus for text simplification is improving reading comprehension and fluency in children. Text reading fluency, which refers to the ability to read text smoothly and accurately, plays a crucial role in reading development and comprehension (Kim and Wagner, 2015). Research

has shown that text reading fluency mediates the relationship between word reading fluency, listening comprehension, and reading comprehension in children. In the early stages of reading development, text-reading fluency may not be independently related to reading comprehension, but as children develop their word-reading proficiency, text-reading fluency becomes a significant factor in mediating the relationship between word-reading fluency and reading comprehension. A study by Javourey-Drevet et al. (2022) has found that text simplification can improve reading fluency and comprehension in beginning readers. Moreover, simplified texts have been shown to benefit poor readers and children with weaker cognitive skills to a greater extent than good readers and children with stronger cognitive skills.

3.5 Second Language Learners

Second language (L2) learners are individuals who are learning a language that is not their first language. These learners have unique characteristics that affect their language acquisition process. One of the most significant factors that influence second language acquisition is the age of the learner. According to the key age hypothesis, younger learners have a better chance of acquiring a second language than older learners (Deng and Wang, 2019). However, this hypothesis has been challenged by research that shows that adult learners can also acquire a second language successfully (Herschensohn, 2007).

Another characteristic of second language learners is their attitude towards the target language, culture, and speakers. Learners who have a positive attitude towards the target language and culture tend to be more successful in acquiring the language (Kusmiatun and Liliani, 2020). Additionally, learners who are motivated to learn the language and have the willingness to communicate in the target language are more likely to succeed in language acquisition (Yang and Wu, 2017).

Individual differences also play a significant role in second language acquisition. Learners differ in their aptitude, learning style, and cogni-

tive abilities, which can affect their language acquisition process (Yang and Chen, 2018). Some learners may progress rapidly, while others may struggle with language acquisition (Mohammed, 2020).

Learning a second language is a complex process that requires a lot of effort and dedication. Second language learners face various challenges that affect their ability to read and write in the target language. According to Olajide (2010) and Malebese et al. (2019), all categories of ESL learners display difficulty in learning to read and write. The transition from the use of the home language to the second language, namely English first additional language, is complexly related to the learners' inability to read text meaningfully. Differentiation of whether English language learners' struggles are symptomatic of reading disability or related to second language acquisition is often challenging (Gorman, 2009).

One of the challenges that second language learners face is reading difficulties. Hmeljak Sangawa (2016) notes that reading is one of the bases of second language learning, and it can be most effective when the linguistic difficulty of the text matches the reader's level of language proficiency. Moreover, knowing the vocabulary is also an important factor for learning a new language, and most of the vocabulary is usually acquired through reading, so having slightly more difficulty than the current reader's level allows the progress of language development (Krashen, 1989).

In the context of second language learning, text simplification has been found to have several benefits. It can facilitate comprehension and processing of open educational resources in English (Rets and Jekaterina Rogaten, 2020). A study conducted on adult English L2 users showed that simplification led to better text comprehension, particularly at lower English proficiency levels. Eye-tracking measures also revealed that text simplification resulted in changes in processing time during reading, indicating its impact on reading behavior.

Different approaches have been proposed for text simplification. The structural approach focuses on using traditional readability formulas and involves replacing rare words with more frequent ones and shortening sentences. On the other hand, the intuitive approach relies on the expe-

riences of a language teacher, learner, or materials writer to guide the simplification process (Crossley et al., 2011). Both approaches have been widely used in text simplification for second language learners.

Recently, Degraeuwe and Saggion (2022) proposed a lexical simplification system targeting Dutch speakers learning Spanish by simplifying all the words except the vocabulary that needs to be studied. The model consists of four modules: complex word identification (CWI), substitution generation, substitution selection, and substitution ranking. The complex word identification module is a lexicon-based model with different features, including frequency, and word familiarity, fine-tuned with RoBERTa-BNE (Gutiérrez-Fandiño et al., 2022). The CWI measures the word complexity of all words in a sentence based on the levels of the Common European Framework (CEFR). Next, the substitution generation module generates candidates using a masked language model approach as described in Chapter 2 Section 2.1.3, the candidates are selected by POS filter, and ranked by different criteria, such as MLM probability, language model score (Qiang et al., 2021), lemma frequency in SCAP corpora, token frequency in SUBTLEX-ESP (Cuetos et al., 2012), cosine similarity with FastText and RoBERTa-BNE.

3.6 Deaf and Hard of Hearing

Deaf and hard-of-hearing individuals have unique characteristics that are shaped by their experiences and identity. According to research (Mikhailova et al., 2019), personality characteristics in deaf and hard-of-hearing individuals can vary depending on their self-identification type. Some individuals may identify as culturally Deaf, using sign language and affiliating with Deaf culture, while others may identify as audio-logically deaf or hard of hearing.

Deaf and hard-of-hearing individuals may face barriers in education (Scherer et al., 2023; Adoyo and Maina, 2019; Anderson et al., 2021; A. El-Zraigat, 2012; Wurst et al., 2005; Ristić et al., 2021; Roksandić et al., 2018; Alsraisri et al., 2020). These barriers can include societal myths,

teacher incompetence in the language of instruction, low expectations, and difficulties in acquiring reading and writing skills (Adoyo and Maina, 2019; Wurst et al., 2005). In addition, deaf and hard-of-hearing children are at risk of exclusion from community life and education, which may increase their risk of mental health conditions (Scherer et al., 2023).

Children who are deaf and hard of hearing often struggle with grammar (Lederberg et al., 2013), syntactically complex sentences (Siddharthan, 2006), limited vocabulary, and difficulties generalizing word's meaning in different contexts (Fabbretti et al., 1998).

Individuals who are deaf or hard of hearing often experience difficulties with reading and literacy skills. Research has shown that more than 90% of students who are deaf or hard of hearing have hearing parents who do not share an effective mode of communication with their children, which can result in a lack of explicit teaching of literacy skills (Howell and Luckner, 2003). This lack of exposure to language and literacy can lead to poor reading abilities and low levels of reading achievement. While some students who are deaf or hard of hearing may be skilled in critical areas such as vocabulary and grammar, many find acquiring reading and writing skills to be the most difficult academic hurdles they face (Wurst et al., 2005).

It is important to note that the difficulties experienced by individuals who are deaf or hard of hearing with reading and literacy skills can have significant impacts on their academic success and overall well-being. Without strong reading and writing skills, individuals who are deaf or hard of hearing may struggle to fully engage in classroom activities and may experience academic failure. However, research has also shown that factors such as psychological well-being and subjective well-being may be related to higher mastery in reading comprehension and mathematics, highlighting the importance of considering the whole person in supporting their literacy development (Gonzalez and Camacho-Vega, 2021).

Research has explored the use of text simplification to improve the accessibility of deaf and hard-of-hearing individuals. A study by Kushalnagar et al. (2016) developed a two-step approach for simplifying cancer and other health text for deaf people who use American Sign Language.

The study tested the approach with a sample of deaf and hearing college students and found that the simplified text significantly improved the comprehension of deaf college students. This research highlights the importance of text simplification in making health information more accessible for deaf individuals.

3.7 Intellectual Disabilities

Intellectual disability is a condition that affects an individual's cognitive and adaptive functioning. According to Millichap and Millichap (2014) and Vasilakopoulou and Tzvetkova-Arsova (2021), intellectual disability is characterized by significant limitations in intellectual functioning and adaptive behavior expressed in conceptual, social, and practical adaptive skills. Individuals with intellectual disabilities have limited intelligence, social and other mental functions, and difficulty in learning, reasoning, and problem-solving (Phytanza et al., 2018).

People with intellectual disabilities also have unique physiological and psychological characteristics. They may have difficulty in expressing their emotions and understanding the emotions of others (Sivasubramanian, 2020). Emotional and behavioral impairments are common in individuals with intellectual disabilities (Harris, 2005). Psychopathology of mental and behavioral disorders is also prevalent in this population (Dębska et al., 2020).

People with intellectual disabilities often face challenges in reading and developing reading skills. These challenges can be attributed to various characteristics of individuals with intellectual disabilities. Firstly, individuals with intellectual disabilities may have difficulties in phonological awareness, which is the ability to recognize and manipulate sounds in words. This can make it challenging for them to decode words and understand their meanings (Sun and Kemp, 2006). Secondly, individuals with intellectual disabilities may have limited vocabulary and comprehension skills, which can affect their ability to understand what they are reading (Cox-Magno et al., 2018). Thirdly, individuals with intellectual disabili-

ties may have difficulty with metacognition, which is the ability to reflect on and regulate one's own thinking processes. This can make it challenging for them to monitor their own reading comprehension and adjust their reading strategies accordingly (Cox-Magno et al., 2018).

Moreover, individuals with intellectual disabilities may have difficulties in communication skills, including speech and language, which can affect their ability to understand and express themselves through reading (Licardo et al., 2021). They may also have limitations in writing, reading, speaking, and listening skills, which can affect their ability to learn a new language, such as English (Dalilan et al., 2021). Additionally, individuals with intellectual disabilities may have difficulty accessing age-appropriate texts due to their reading skills (Shurr and Taber-Doughty, 2017).

Intellectual disabilities can make it difficult to process complex language and vocabulary, which can lead to difficulties in reading and comprehension. Text simplification can be particularly beneficial, as simplified text can help to reduce these barriers and make the content more accessible (Saggion et al., 2011).

3.8 Summary

This chapter provides a brief introduction to various targeted users group that could be benefited from ATS. Some of the benefits are 1) Improved Comprehension: text simplification could help users with Autism, Dyslexia, Aphasia, and Children to understand better the content they are reading. By using simpler language, shorter sentences, and more straightforward vocabulary, users can more easily comprehend the information presented to them. 2) Increased Independence: for users with Autism, Dyslexia, Aphasia, and Children, text simplification can help them become more independent in their reading and learning. By providing them with content that is easier to understand, they can rely less on others for help and feel more confident in their abilities. 3) Enhanced Learning: text simplification can also benefit Second Language Learners by providing them with content that is easier to understand and learn from. By using

simpler language and vocabulary, learners can more easily grasp new concepts and improve their language skills. 4) Accessibility: for users who are Deaf or Hard of Hearing, text simplification can provide an accessible alternative to audio or video content. By providing written content that is easier to understand, users can access information that might otherwise be difficult or impossible to access. Overall, text simplification can provide numerous benefits for users with different needs, including improved comprehension, increased independence, enhanced learning, and accessibility.

Part II

Complex Word Identification

Chapter 4

Complex Word Identification

In this chapter, we describe two of our experiments on Complex Word Identification (CWI). The first experiment is based on Convolutional Neural Networks (CNN) in combination with different features such as word embeddings, morphological features, and linguistic features to identify complex words in English, Spanish, and German texts. The second experiment explores two additional algorithms along with a new set of features selected specifically to tackle complex words in French biomedical documents.

4.1 Introduction

Complex Word Identification (CWI) is an essential task in helping Lexical Simplification (LS) identify the difficult words that should be simplified. LS simplifies text mainly by substituting difficult and less frequently-used words with simpler equivalents. The majority of works on CWI are either feature-engineered or neural networks with word embeddings. Both approaches have advantages and limitations, so here we combine both approaches in order to achieve higher performance while still supporting multilingualism.

In the first experiment, we carried out our experiments on the data from the CWIG3G2 dataset used in Complex Word Identification Shared

Task 2018 (Yimam et al., 2017b). The following sentence shows an English example of the CWIG3G2 dataset. The target text **flexed their muscles** is annotated as complex by at least one annotator.

Both China and the Philippines **flexed their muscles** on Wednesday.

The second experiment is focused on a French biomedical dataset. Medical documents are often considered one of the most challenging texts to read and understand by a large population. The difficulty is mainly on a lexical level since the vocabulary of medical texts is very specific and because of the document's rich terminological status.

For example, the following sentence from a biomedical document contains difficult words such as (*OPA (acute pulmonary edema), résolutif (resolvent), VNI (NIV), oxygénothérapie (oxygen therapy)*) which, if simplified or explained, could make the text more understandable.

Le patient est donc transféré en réanimation : l'OAP est résolutif sous VNI et oxygénothérapie.
Hence, the patient is transferred to intensive care: acute pulmonary edema is resolvent with NIV and oxygen therapy.

We make the following contributions:

- We propose an approach that combines feature engineering and deep learning (CNN) to identify complex words in English, Spanish, German, and French texts[1].
- We make an analysis comparing the deep learning approach with two classical machine learning approaches.

[1]The code is available at https://github.com/kimchengsheang/cwi_cnn and https://github.com/kimchengsheang/MedCWI

4.2 Related Work

Complex word identification has attracted the attention of researchers recently, either as part of text simplification systems (Shardlow, 2013; Paetzold and Specia, 2015) or as an independent task, such as promoted by SemEval 2016 (Paetzold and Specia, 2016d), 2018 CWI shared task (Yimam et al., 2018), or SemEval 2021 (Shardlow et al., 2021).

The first complex word identification SemEval was introduced in 2016 to identify difficult words in an English corpus. The task was a binary classification task in which the model had to decide whether the target word was difficult or not for non-native speakers (Paetzold and Specia, 2016d). Participants exploited different classifiers such as Support Vector Machine (SVM), Decision Tree, Random Forest, Logistic Regression, and Recurrent Neural Network (RNN) with features like word embeddings, lexical, morphological, and psycholinguistic properties of the target words and Part-of-Speech (POS) tags.

In 2018, the CWI shared task extended its purpose to the identification of difficult words in different languages (English, German, French, and Spanish), as well as in a multilingual corpus (Yimam et al., 2018). Two tasks were proposed: binary classification task (to decide whether the word is difficult or not) and probabilistic classification task (to assign a probability to a given word as being difficult). Participants used different classifiers (e.g., SVM, Naive Bayes, Random Forest) and different combinations of features such as word frequency, semantics, lexical, morphological, and psycholinguistic properties. In addition, some participants started to use the context of target words and word embeddings. The results of the binary classification task, F-measures, were between 0.176 and 0.874 for the English corpus and between 0.577 and 0.745 for other languages.

The purpose of the second SemEval 2021 task was also to identify difficult words in texts from different genres in English: European Parliament, Bible, and biomedical texts (Shardlow et al., 2021). Two main differences from previous tasks: words are annotated using a 5-point Likert scale (from *very easy* to *very difficult*) and consideration of polylex-

ical units. Participants used language models, such as BERT (Devlin et al., 2019) or RoBERTa (Liu et al., 2019), and Gradient Boosted Regression. The evaluation metric was Pearson's Correlation and the participants obtained scores between 0.7886 and -0.0272 for the processing of single words and scores between 0.8612 and 0.1860 for the processing of polylexical units.

Besides challenge papers, Gala et al. (2014) proposed an approach to predict the lexical complexity of French words for non-native learners. The authors use SVM with different features (e.g., word length, number of phonemes, number of syllables, phoneme/spelling coherence). They obtain an accuracy between 43% and 63%. In another work, three methods for the detection of difficult words in general language English corpora are evaluated (Shardlow, 2013): (1) simplify everything, (2) exploit frequency using reference corpus, (3) train SVM model. The SVM-based method shows the highest performance with 0.771 precision, while the simplifying everything method has 0.738 precision and the frequency-based method has 0.709 precision. The frequency threshold is exploited in another work for the detection of familiarity with medical terms (Zeng et al., 2005). The experience obtains 0.196 mean absolute error and 0.293 root mean square error. Yet, another way to determine word complexity is based on the rarity of words: the words that are not found in different lexica are considered as difficult (Borst et al., 2008). This method shows 92% accuracy.

Many techniques have been introduced so far to identify complex words. It is obvious that feature-based approaches remain the best; however, deep-learning approaches have become more popular and achieved impressive results. Our deep learning-based model follows that of Aroyehun et al. (2018) in combination with word embeddings and linguistic features.

4.3 Multilingual complex word identification with Convolutional Neural Networks

In this section, we describe our CWI approach based on Convolutional Neural Networks (CNN) with word embeddings and engineered features. In the following, we describe the datasets, how to prepare the data, and the training details.

4.3.1 Methodology

In this section, we describe our CWI approach based on CNN with word embeddings and engineered features. Figure 4.1 shows the overall architecture of our model. It is implemented using pure Tensorflow deep learning library version 1.14.[2].

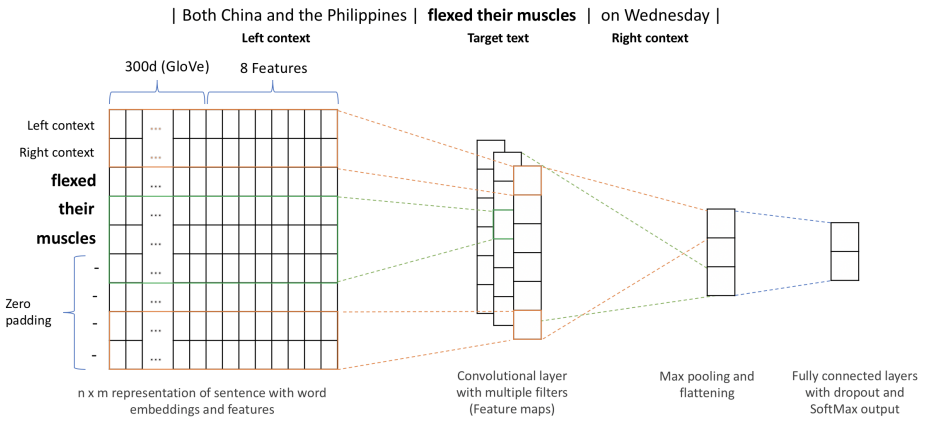


Figure 4.1: The CWI model architecture based on CNN.

[2] https://www.tensorflow.org

Features

In this experiment, we employ word embeddings, morphological and linguistic features. word embedding feature gives us a details representation of each word, whereas morphological and linguistic features provide the characteristics of each word, which are correlated with word complexity. Below we describe the details of each feature.

Word Embedding Feature: We use pre-trained word embeddings GloVe (Pennington et al., 2014) with 300 dimensions to extract the word vector representation of each word for all three languages. For English, we use the model trained on Wikipedia 2014 and Gigaword 5 model (6B tokens, 400K vocab).[3] For Spanish, we use the model (Cardellino, 2016) trained on 1.5 billion words of data from different sources: dumps from the Spanish Wikipedia, Wikisource, and Wikibooks on date 2015-09-01, Spanish portion of SenSem, Spanish portion of Ancora Corpus, Tibidabo Treebank and IULA Spanish LSP Treebank, Spanish portion of the OPUS project corpora, and Spanish portion of the Europarl.[4] For German, we use the model trained on the latest dumps of German Wikipedia.[5]

Morphological Features: Our morphological feature set consists of word frequency, word length, number of syllables, number of vowels, and TF-IDF.

- Word frequency: the frequency of each word is extracted from the latest Wikipedia dumps as the raw count and then normalized to between 0 and 1.
- Word length: the number of characters in the word.
- Number of syllables: the number of syllables of the word, calculated using Pyphen.[6]
- Number of vowels: the number of vowels in the word.

[3] https://nlp.stanford.edu/projects/glove
[4] https://github.com/dccuchile/spanish-word-embeddings
[5] https://deepset.ai/german-word-embeddings
[6] https://pyphen.org

- TF-IDF: Term frequency-inverse document frequency, calculated using scikit-learn library.[7]

Linguistic Features: The linguistic features consist of part-of-speech, dependency, and stop words.

- Part-of-Speech (POS): a category to which a word is assigned in accordance with its syntactic functions, e.g. noun, pronoun, adjective, verb, etc.
- Dependency: a syntactic structure consists of relations between words, e.g. subject, preposition, verb, noun, adjective, etc.
- Stop word: a commonly used word such as "the", "a", "an", "in", "how", "what", "is", "you", etc.

All these features are extracted using SpaCy (Honnibal and Montani, 2017).

4.3.2 Experiments

In the following, we describe the datasets, how the data are prepared, and the training details.

Datasets

We use the CWIG3G2 datasets from (Yimam et al., 2017a,b) for our CWI system for both training and evaluation. The datasets are collected for multiple languages (English, Spanish, German). The English dataset contains news from three different genres: professionally written news, WikiNews (news written by amateurs), and Wikipedia articles. For Spanish and German, they are collected from Spanish and German Wikipedia articles. For English, each sentence is annotated by 10 native and 10 non-native speakers. For Spanish dataset, it is mostly annotated by native speakers, whereas German is annotated by more non-native than native

[7] https://scikit-learn.org

speakers. Each sentence contains a target text which is selected by annotators, and it is marked as complex if at least one annotator annotates as complex. Table 4.1 shows all the details about each dataset used in the experiments.

Lang	Genre	Train	Dev	Test	Positive
English	News	14,002	1,764	2,095	40%
	Wikinews	7,746	870	1,287	42%
	Wikipedia	5,551	694	870	45%
Spanish	Wikipedia	13,750	1,622	2,233	40%
German	Wikipedia	6,151	795	959	42%

Table 4.1: Statistics of the CWIG3G2 dataset (Yimam et al., 2017a,b). The column positive shows the percentage of selected target text labeled as complex.

Data Preprocessing

We separate each sentence into three parts: target text, left context, and right context. The target text is a word or phrase that is selected and marked as complex or non-complex by the annotators. The left context and the right context are words that appear to the left and the right of the target text.

First, we remove all special characters, digits, and punctuation marks. Then, each word is replaced by its word vector representation using pre-trained word embeddings from the GloVe model as described in Section 4.3.1. Words that do not exist in the pre-trained word embeddings are replaced with zero vectors. Afterward, we transform the left context and right context into a 300-dimensional vector calculated as the average of the vectors of all the words in the left context and the right context. If the left context or right context is empty (when the target text is at the beginning or the end of the sentence), we replace it with a zero vector.

Next, we initialize a matrix X of size $n * m(n = h + 2, m = 308)$ where the first row corresponds to the left context vector, the second row corresponds to the right context vector, and the last r rows are given by the embedding vectors of the words contained in the target text, where r is the number of words in the target text. In order to have a fixed-size matrix, we pad the remaining rows p with zero vectors, where $p = h - r$ and h is the maximum value of r in the corpus.

To convert each feature into a vector representation, first, we need to transform its values. For example:

- Part-of-speech and Dependency are given the values such as N as 1, V as 2, ADJ as 3, ADV as 4, and PREP as 5, and then are normalized to between 0 and 1.
- Stop word: 1-stop word, 0-otherwise.
- All the values of word frequency, word length, number of syllables, number of vowels, and tf-idf are numbers, so we just normalize it to between 0 and 1.

For each feature, we initialize a matrix of one column and n rows where the first row corresponds to the average value of the left context, the second row corresponds to the average value of the right context, and the last r rows are the values of the feature for each word in the target text, and the remaining rows are padded with zero. Then, we append the matrix to the matrix X (the representation of the matrix X can be found in Figure 4.1).

Hyperparameters and Training

We train our model using CNN with a number of filters of 128, a stride of 1, and a kernel size of 3, 4, and 5. We then apply the ReLU activation function with Max Pooling to the output of this layer, and it is often called feature maps. The feature maps are flattened and pass through three Fully-Connected layers (FC) with dropouts between each layer. The first two FC layers use the ReLU activation function with 256 and 64 outputs. The last

FC layer uses the Softmax activation function, which provides the output as complex (1) or non-complex (0). Figure 4.1 shows the representation of the architecture.

For all datasets, the training is done through Stochastic Gradient Descent over shuffle mini-batches using Adam optimizer (Kingma and Ba, 2015) with the learning rate of 0.001, a dropout rate of 0.25, mini-batch size of 128. Also, we use weighted cross-entropy as a loss function with a weight of 1.5 for the positive since our datasets are imbalanced; it contains roughly 60% negative examples and 40% positive examples as shown in Table 4.1. We train the system for 200 epochs, and for every 20 iterations, we validate the system with the shuffle development set. Then, if the model achieves the highest F1-score, we save the model and use it for our final evaluation with the test set. In our case, all the hyperparameters are selected via a grid search over the English development set.

We train and evaluate each language separately. For English, the dataset has three different genres, so we combine and train all at once. For Spanish and German datasets, there is only one genre, so we use it directly for training.

4.3.3 Results and Discussion

System	English			Spanish	German
	News	WikiNews	Wikipedia		
Camb (Gooding and Kochmar, 2018)	**87.36**	**84**	**81.15**	-	-
TMU (Kajiwara and Komachi, 2018)	86.32	78.73	76.19	76.99	74.51
NLP-CIC (Aroyehun et al., 2018)	85.51	83.08	77.2	76.72	-
ITEC (De Hertog and Tack, 2018)	86.43	81.10	78.15	76.37	-
NILC (Hartmann and Santos, 2018)	86.36	82.77	79.65	-	-
CFILT_IITB (Wani et al., 2018)	84.78	81.61	77.57	-	-
SB@GU (Alfter and Pilán, 2018)	83.25	80.31	78.32	72.81	69.92
Gillin Inc.	82.43	70.83	66.04	68.04	55.48
hu-berlin (Popović, 2018)	82.63	76.56	74.45	70.80	69.29
UnibucKernel (Butnaru and Ionescu, 2018)	81.78	81.27	79.19	-	-
LaSTUS/TALN (AbuRa'ed and Saggion, 2018)	81.03	74.91	74.02	-	-
Our CWI	86.79	83.86	80.11	**79.70**	**75.89**

Table 4.2: The evaluation results based on macro-averaged F1-score, higher means better.

Table 4.2 shows the results of our model against others (all the results are based on macro-averaged F1-score).

Our evaluation has shown that when training with a dataset that has more training examples, the model achieves a better result. For example, the model achieves F1-score of 86.79 on the English News dataset with 14,002 examples compared to a score of 83.86 on the English WikiNews dataset with 7,746 examples and a score of 80.11 on the English Wikipedia dataset with 5,551 examples.

After an analysis of the results, we have found some samples that contain a word that can be both complex and non-complex at the same time, depending on the selection of the target. The following are two examples:

Example 1:

> The distance, chemical composition, and age of Teide 1 could be established because of its membership in the young **Pleiades** star cluster.

"**Pleiades**" is selected as the target text, and annotators annotated as complex, and our system also predicts it as complex.

> The distance, chemical composition, and age of Teide 1 could be established because of its membership in the young **Pleiades star cluster**.

"**Pleiades star cluster**" is selected as the target text, and annotators annotated as non-complex, but our system predicts it as complex.

Example 2:

> Definitions have been determined such that the super **casino** will have a minimum customer area of 5000 square metres and at most 1250 unlimited-jackpot slot machines.

"**casino**" is selected as the target text, and annotators annotated as non-complex, and our system predicts it as non-complex.

> Definitions have been determined such that the **super casino** will have a minimum customer area of 5000 square metres and at most 1250 unlimited-jackpot slot machines.

"**super casino**" is selected as the target text, and annotators annotated as complex, but our system predicts it as non-complex.

Based on these examples, it is a good indication that the surrounding context is very important for identifying complex words.

4.4 Identification of complex words in French medical documents

In this section, we further investigate two additional approaches along with a new set of features selected specifically to tackle complex words in French biomedical documents.

4.4.1 Methodology

In this experiment, we train three models based on CNN (LeCun et al., 1995), CatBoost (Dorogush et al., 2018), and XGBoost (Chen and Guestrin, 2016). More details about each model will be described in Section 4.4.2. Each model is incorporated with a set of features, as described below:

Features

Our models rely on a number of features to perform the classification. We adopt some features from the previous experiments, such as word length, number of syllables, number of vowels, and TF-IDF. The rest of the features are newly proposed and selected specifically for this experiment.

1. *FastText Embedding* (Bojanowski et al., 2016) a language model pre-trained on large unlabeled corpora is used as word representation. In the previous experiment, we used GloVe word embeddings;

however, in this experiment, we use FastText because it is trained with both word and subword information, which can deal better with rare or unknown words, which we believe that it is suitable for medical texts;

2. *CamemBERT Embedding* (Martin et al., 2020b) is a contextualized word embedding, which is a French version of BERT (Devlin et al., 2019). It is pre-trained on a large amount of text using a word masking approach. Having contextualized word embedding would give additional context information to the model, which in the previous experiment was lack of. In this experiment, we use Flair (Akbik et al., 2019) to extract word embeddings for the whole sentence from the 12 embedding layers and then compute the average. After that, we extract the embedding of each word in the sentence with a dimension of 768 each;

3. *Word Length* is the number of characters in a word;

4. *Word Syllable* is the number of syllables in each word, extracted using PyHyphen[8];

5. *Vowel Count* is the number of vowels in each word;

6. *Word Rank* is the frequency order taken from FastText pre-trained model;

7. *TF-IDF* (Salton, 1991) permits to measure how a sentence is relevant to a document;

8. *LangGen Frequency* is a frequency computed from French Wikipedia;

9. *Clear Frequency* is a frequency computed from a French medical corpus (Grabar et al., 2018).

[8]https://github.com/dr-leo/PyHyphen

4.4.2 Experiments

In this section, we present the dataset, the preprocessing steps, and the models we propose in this chapter. The goal of the experiments is to classify the target text in a sentence as complex or not complex and evaluate it based on the manual annotation of the dataset.

Dataset

We utilized a sample of 100 clinical cases from the CAS corpus (Grabar et al., 2018), which is a collection of clinical cases in French. These clinical cases are similar to clinical reports and provide detailed information about patient's medical backgrounds, reasons for consultation, healthcare processes, treatments, and outcomes. The CAS corpus contains a total of 4,900 clinical cases in French, with nearly 1.7 million word occurrences, which are extracted from different freely accessible sources, and all the patient's information is fully anonymous.

The corpus with clinical cases is pre-processed. The documents are syntactically analyzed by the Cordial parser (Laurent et al., 2009) to divide them into syntactic groups (chunks). When a given word belongs to a chunk within another chunk, we keep a minimal chunk. The corpus contains in total of 15,053 chunks.

Documents are then annotated manually by nine annotators in order to mark up the chunks with understanding difficulty (e.g., understand, not understand, not fully understand). The annotators are all native French speakers, and they have no medical knowledge or training. Few of them are chronically ill. During the annotation process, the annotators were advised not to use dictionaries or information available on the Internet. They had to do the annotations on the basis of their own knowledge. The annotators are presented with whole documents, where chunks are between brackets, such as shown in Figure 4.2. For each chunk, the annotators have to indicate whether they cannot understand it (in red) or whether they are not sure to understand it (in blue). In the case they understand a given chunk, they do not have to annotate it.

In the end, each document is annotated by at least four annotators,

[Ses antécédents médicaux] [montrent] [notamment] [un diabète gestationnel probable] [et une HG] [lors de sa première grossesse]. [La patiente] [avait alors été hospitalisée] et [avait recu] [un traitement intraveineux] [de métoclopramide associé] [à de la diphénhydramine suivi] [d'un relais] [par voie orale] [au métoclopramide et] [à l'hydroxyzine]. [Une réaction extrapyramidale] ([rigidité] [de la mâchoire et] [difficulté] [à parler]) [avait nécessité] [l'arrêt] [du métoclopramide]. [L'hydroxyzine] [avait] [ensuite été remplacée] [par l'association] [de doxylamine] [et de pyridoxine] (DiclectinMD).

[Her medical background] [shows] [a probable gestational diabetes] [and an HG] [during her first pregnancy]. [The patient] [had then been hospitalized] and [received] [an intravenous treatment] [of metoclopramide with] [diphenhydramine followed] [by oral treatment] [with metoclopramide and] [hydroxyzine]. [An extrapyramidal reaction] ([jaw] [stiffness and] [difficulty] [to talk]) [caused] [the cessation] [of metoclopramide]. [Hydroxyzine] [had] [then been replaced] [by the combination] [of doxylamine] [and pyridoxine] (Di-clectinMD).

Figure 4.2: An example of an annotated clinical case. Each chunk of the text is grouped by the parser. The colors are marked by an annotator: black means understand, red means not understand, and blue means not fully understand.

Annotators	*Kappa*
all (1-4)	0.175
1 & 2	0.093
1 & 3	0.292
1 & 4	0.100
2 & 3	0.316
2 & 4	0.115
3 & 4	0.048

Table 4.3: Kappa scores for different annotators

while some documents are annotated by up to six annotators. We computed the kappa of Fleiss (Fleiss, 1971) for four annotators who annotated all the documents, which gives a low 0.175 Kappa. For some pairs of annotators, Kappa shows slightly higher values (0.292 and 0.316). Table 4.3 shows all the Kappa scores between the four annotators. This means that the annotation task is very subjective and heavily depends on own knowledge and experience of each person.

The corpus is then segmented into sentences containing one target chunk per sentence. In total, we have 9,709 sentences with 3,482 complex and 6,227 non-complex chunks. Then for each training and evaluation, we shuffle the data with a different seed number and split it into three parts: 70% for training, 15% for validation, and 15% for testing.

Data Preprocessing

The preprocessing steps follow the same procedures as described in Section 4.3.2 with all the features mentioned in Section 4.4.1. Figure 4.3 shows an example of the vector representations of how a sentence with a selected target text is preprocessed.

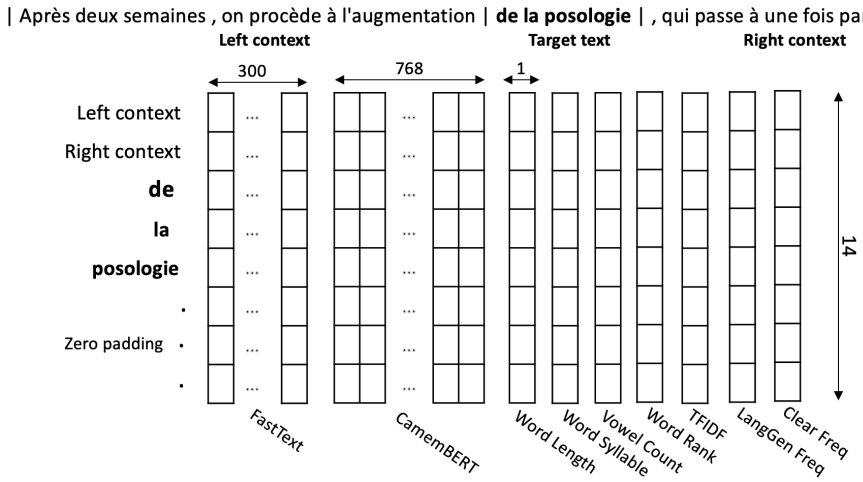


Figure 4.3: An example of the vector representations of how a sentence with a selected target text is preprocessed with all the features.

Models

We use three models, a CNN already described in Section 4.3 and two classifiers described below:

CatBoost (Dorogush et al., 2018) is a gradient boosting on. The decision Trees library is often used for ranking, regression, and classification tasks. CatBoost is an ensemble learning library that combines multiple machine learning algorithms (Decision Trees) to obtain a better model. To train the model, we follow the same data preparation step as in the CNN model and then flatten it into a long vector. The model is trained for 1200 iterations with a learning of 0.03.

XGBoost (Chen and Guestrin, 2016) stands for Extreme Gradient Boosting is a scalable, distributed gradient-boosted decision tree machine learning library similar to CatBoost. It is one of the leading machine-learning libraries for regression, classification, and ranking problems. To train the model, the data is prepared the way as in CatBoost model, and then trained with the max depth of 10, the learning rate of 0.03, and the number of estimators of 500.

4.4.3 Results and Discussion

Table 4.4 shows the results of our three models (CNN, CatBoost, and XGBoost) trained with different feature sets. The results are indicated in a macro average of precision (P), recall (R), and F1-score (F1). The models are trained with each feature set five times: the average values are indicated. The column *Features* indicates the combinations of features used in each model. Each feature number corresponds to those in Section 4.4.1. The CNN model performs better in all cases except when the CNN model is trained without word embeddings, it performs lower than CatBoost and XGBoost. We get up to 0.854 F1-score.

We started with the CNN model trained with only FastText word embedding, and then we kept adding more features one by one. We could see that the result was improved each time we added a new feature. Then, we tried removing the embedding feature, and the result dropped significantly. Next, we tried BERT embedding (CamemBERT), and the result was way better than all models with FastText. It can be due to the fact that BERT embedding has captured better information on words than FastText. Next, we combined FastText with BERT; as a result, the CNN model performed worse than the model with BERT alone, whereas CatBoost and XGBoost models performed better, especially the CatBoost model performed the best among all of its models. Even though CatBoost and XGBoost models perform pretty well, it is still significantly lower than the CNN model.

In comparison with all the models, the results show that in most cases the CNN model performs better than CatBoost and XGBoost, except in the model without the word embeddings; this could be an indication that CatBoost and XGBoost learn better than the CNN model when having less information.

4.5 Conclusion

In this chapter, we have presented two experiments for complex word identification for two different datasets. The first experiment utilized a

Features	CNN			CatBoost			XGBoost		
	P	R	F1	P	R	F1	P	R	F1
1	0.818	0.811	0.814	0.798	0.773	0.783	0.798	0.770	0.780
1, 3	0.817	0.813	0.815	0.799	0.771	0.781	0.796	0.770	0.779
1, 3, 4	0.819	0.815	0.816	0.804	0.781	0.790	0.800	0.772	0.782
1, 3, 4, 5	0.820	0.817	0.818	0.803	0.780	0.788	0.797	0.770	0.780
1, 3, 4, 5, 6	0.826	0.818	0.821	0.829	0.812	0.819	0.799	0.774	0.783
1, 3, 4, 5, 6, 7	0.823	0.820	0.821	0.830	0.813	0.820	0.826	0.811	0.817
1, 3, 4, 5, 6, 7, 8	0.824	0.824	0.824	0.826	0.809	0.816	0.825	0.812	0.818
1, 3, 4, 5, 6, 7, 8, 9	0.827	0.826	0.826	0.831	0.817	0.823	0.826	0.812	0.818
3, 4, 5, 6, 7, 9	0.787	0.778	0.781	0.803	0.786	0.793	0.802	0.789	0.795
1, 2	0.848	0.843	0.845	0.838	0.824	0.830	0.834	0.817	0.824
2	0.848	0.847	0.847	0.823	0.807	0.814	0.824	0.799	0.809
2, 3	0.848	0.847	0.847	0.825	0.811	0.817	0.825	0.802	0.811
2, 3, 4	0.852	0.851	0.851	0.826	0.812	0.818	0.825	0.802	0.811
2, 3, 4, 5	**0.854**	0.852	0.853	0.822	0.808	0.814	0.824	0.798	0.808
2, 3, 4, 5, 6	0.851	0.850	0.851	0.821	0.806	0.812	0.820	0.796	0.805
2, 3, 4, 5, 6, 7	0.851	0.847	0.849	0.829	0.817	0.822	0.834	0.816	0.823
2, 3, 4, 5, 6, 7, 8	0.851	0.846	0.848	0.822	0.810	0.815	0.827	0.815	0.820
2, 3, 4, 5, 6, 7, 8, 9	0.849	0.844	0.846	0.832	0.817	0.824	0.835	0.822	0.827
2, 3, 4, 5, 6, 8, 9	0.853	**0.856**	**0.854**	0.826	0.815	0.820	0.834	0.822	0.827

Table 4.4: This table shows the results in a macro average of precision (P), recall (R), and F1-score (F1) of our three models trained with different combinations of features. All models are trained with each feature set five times and computed the average. A higher value means better. The column feature lists all combinations of features used in the training of each model, and each number represents the corresponding feature listed in Section 4.4.1.

deep learning model (CNN) with word embeddings and engineered features. The evaluations have shown that the model performs quite well compared to the state-of-the-art system for English, which depends on a lot of engineered features and is better than the state-of-the-art systems for both Spanish and German. In the second experiment, we have proposed three classifiers based on CNN, CatBoost, and XGBoost trained with different feature sets selected specifically to detect complex words in French biomedical documents. The results have shown that the CNN model performs better than CatBoost and XGBoost.

Part III

Lexical Simplification

Chapter 5

Controllable Lexical Simplification for English

In this chapter, we present our lexical simplification approach based on a Transformer-based model along with a controllable mechanism for monolingual English. The controllable mechanism gives us the ability to control the outputs based on the token values (see Chapter 2 Section 2.2.4) embedded into each input sentence, such as word length, word frequency, and candidate ranking.

5.1 Introduction

Recently, fine-tuning Transformer-based approaches have shown exciting results in sentence simplification (Sheang and Saggion, 2021; Martin et al., 2022). However, so far, no research has applied similar approaches to the lexical simplification task. Figure 5.1 shows an example of a lexical simplification where the word "hiatus" is selected as a complex word and is replaced by the word "break". In this chapter, we present ConLS, a Controllable Lexical Simplification system fine-tuned with T5 (Raffel et al., 2020). T5 is an encoder-decoder model pre-trained on multiple tasks: unsupervised tasks such as BERT-style span masking (Devlin et al., 2019), and supervised tasks such as machine translation, document sum-

Complex Sentence: The Hush Sound is currently on **hiatus**.
Simplified Sentence: The Hush Sound is currently on **break**.

Figure 5.1: A lexical simplification example taken from the LexMTurk dataset (Horn et al., 2014) with the complex word and the substitute word in bold.

marization, question answering, classification tasks, and reading comprehension. T5 is trained on Colossal Clean Crawled Corpus (C4), a dataset created by applying a set of filters to English texts sourced from the public Common Crawl web scrape.

Our proposed system harnesses the power of transfer learning by fine-tuning the T5 pre-trained model where first the T5 model was trained as a language model and then trained with multiple NLP tasks, such as machine translation, question answering, paraphrasing, summarization, etc. The T5 pre-trained model has already captured some knowledge that would be beneficial to our task, which would require less amount of data for training. As for the controllable mechanism, the idea is to train an end-to-end model (a single model) that does candidate generation, selection, and ranking through the training process. The control tokens are embedded in each input sentence in order to help the model generate simpler candidates and rank them by simplicity.

The systems evaluated in this chapter do not perform complex word identification. We use datasets that already had a complex word tagged for each instance. Moreover, we do not address the morphological and context adaptation task because our model usually returns the correct inflected candidates.

The evaluation results on three datasets (LexMTurk, BenchLS, and NNSeval) have shown that our model performs comparable to LSBert (the current state-of-the-art) see Section 5.4 and even outperforms it in some cases. We also conducted a detailed comparison of the effectiveness of control tokens to give a clear view of how each token contributes to the model.

We make the following contributions:

- To the best of our knowledge, we are the first to introduce a controllable mechanism for LS and to fine-tune a Transformer-based model for LS[1].
- We conduct an extensive evaluation of several metrics that allow us to understand the system better when applied to real-world scenarios.

5.2 Methodology

Following recent works on controllable sentence simplification of Martin et al. (2019), Martin et al. (2022), Sheang and Saggion (2021), and Štajner et al. (2022b), we are inspired to apply a similar approach in lexical simplification task. Specifically, our model is based on Sheang and Saggion (2021), a model originally developed for sentence simplification[2]. We propose a controllable mechanism for lexical simplification because we believe that the embedded token values extracted from training data could give additional information to the model about the relations between the source and the target word; so that at inference, we could define different token values that fulfill our objectives, which in this case is to find the best candidates. In the following paragraphs, we describe all the details about control tokens and the reason why they are chosen.

Word Length (WL) is the character length ratio between the complex word and the target word. It is the number of characters of the target word divided by the number of characters of the complex word. Based on our analysis of the training dataset (TSAR-EN), 65.71% of the time complex word is longer than the best candidate, 21.30% the complex word is shorter than the best candidate, and 12.99% both are the same length.

[1]The code and data are available at https://github.com/kimchengsheang/ConLS

[2]https://github.com/kimchengsheang/TS_T5

Word Rank (WR) is the inverse frequency of the target word divided by that of the complex word. The inverse frequency order is extracted from the FastText pre-trained model. Based on our analysis of the TSAR-EN dataset, 85.45% of the time, the complex word has a lower frequency than the best candidate. Therefore, this token is a good indicator to help guide the model to predict simpler candidates.

Candidate Ranking (CR) is the ranking order extracted from the training data. The values are given to candidates by the ranking order. E.g., the best-ranking candidate is given a value of 1.00, the second 0.75, the third 0.50, the fourth 0.25, and starting from the fifth is given a value of 0.00. We used only five different values to avoid overloading the model, as the training data is relatively small. In addition, the rationale behind using these values is that we want the model to learn candidates ranking from data through the training process rather than injecting additional information or doing post-processing.

5.3 Experiments

In our experiments, we compare our model with the current state-of-the-art model LSBert (Qiang et al., 2020). We used the original LSBert configurations and resources, and we made the following changes to have a detailed comparison with our model. By default, LSBert returns only a single best candidate for each complex word, so we made the changes to return the 10 best-ranked candidates. We changed the number of BERT mask selections from 10 to 15 so that after removing duplicate candidates, we still have around 10 candidates. Moreover, we filtered out all the candidates that were equal to the complex word. Due to the fact that all the used datasets have gold annotated simpler substitutions in all instances, we could assume that returning the complex word would be incorrect.

5.3.1 Datasets

This subsection describes all the lexical simplification datasets for English that we used in our experiments. We used LexMTurk (Horn et al., 2014), BenchLS[3] (Paetzold and Specia, 2016c), and NNSeval [4] (Paetzold and Specia, 2016e) for testing and TSAR-EN (Štajner et al., 2022a) dataset for training and validation. LexMTurk has 500 sentences that were obtained from Wikipedia. This dataset contains the marked complex words and their replacements suggested by 50 English-speaking annotators. The BenchLS dataset is a union of the LSeval (De Belder and Moens, 2012) and LexMTurk datasets in which spelling and inflection errors were automatically corrected. The NNSeval dataset is a filtered version of the BenchLS adapted to evaluate LS for non-native English speakers.

Sentence
European Union foreign ministers agreed Monday to impose fresh sanctions on Syria as a U.N.-backed peace plan – along with all other diplomatic efforts – has yet to stop the **carnage** that mounts every day.
Simpler Substitutes
destruction:6, bloodshed:3, massacre:3, slaughter:3, carnage:2, brutality:1, butchering:1, butchery:1, damage:1, death:1, slaying:1, violence:1, war:1

Figure 5.2: An example taken from the TSAR-EN dataset (Štajner et al., 2022a) with the target word in bold. The numbers after ':' represents the number of workers that suggested the substitution. Each instance has 25 substitutes suggested by 25 crowd-sourced workers.

TSAR-EN dataset has 386 instances with 25 gold-annotated substitutions. Figure 5.2 shows an example. The instances and their target com-

[3] https://doi.org/10.5281/zenodo.2552393
[4] https://doi.org/10.5281/zenodo.2552381

plex words were extracted from the Complex Word Identification shared task 2018 (Yimam et al., 2018). The instances were annotated using Amazon's Mechanical Turk by 25 annotators. A native English annotator reviewed all suggestions.

5.3.2 Evaluation Metrics

We evaluated the systems with several metrics that could take into account the results for different numbers of K candidates (from 1 up to 10). The metrics used are the following:

- **Accuracy@1**: is the ratio of instances with the top-ranked candidate in the gold standard list of annotated candidates.
- **Accuracy@K@Top1**: The ratio of instances where at least one of the top K predicted candidates match the most frequently suggested synonym/s[5] in the gold list of annotated candidates.
- **Potential@K**: the percentage of instances for which at least one of the top K substitutes predicted is present in the set of gold annotations.
- **Mean Average Precision@K (MAP@K)**: This metric evaluates the relevance and ranking of the top K predicted substitutes.
- **Precision@K**: the percentage of top K-generated candidates that are in the gold standard.
- **Recall@K**: the percentage of gold-standard substitutions that are included in the top K-generated substitutions.

5.3.3 Data Preprocessing

For each instance, we have a sentence, a complex word, and a list of ranked candidates. We compute all the ratios and the ranking, then

[5]Ties in the most repeated gold-annotated candidates are taken into account.

prepend it to the source sentence. We also use special tokens [T] and [/T] to mark the boundary of the complex word in the source sentence and the simple word in the target sentence. Moreover, these special tokens help us identify the candidates during the inference. Figure 5.3 shows an example of source and target sentences embedded with token values and boundary tokens.

Source: <CR_1.00> <WL_0.54> <WR_0.90> The Obama administration has seen what The New York Times calls an [T]unprecedented[/T] crackdown on leaks of government secrets.

Target: The Obama administration has seen what The New York Times calls an [T]unusual[/T] crackdown on leaks of government secrets.

Figure 5.3: A training example. The control token values are extracted from the complex word (unprecedented) and one substitute word (unusual). The word unusual is the best-ranked candidate suggested by annotators, so the CR value is 1.00. We used all the candidates in each instance to generate parallel sentences for training. One candidate per training example.

5.3.4 Model Details

The model's implementation is based on Huggingface Transformers (Wolf et al., 2020) and Pytorch-lightning[6].

In our experiments, we fine-tuned the T5-large model on the TSAR-EN dataset. We also compared the difference sizes of T5 pre-trained models; the results are in Table 5.3. We split the dataset to 90% for training and 10% for validation. This 10% validation set is also used in the token values search at the inference, as described in the following section. For the training data, we preprocessed by extracting and adding control tokens

[6]https://www.pytorchlightning.ai

to the source sentence along with the boundary tokens to the complex word and substitute word, as shown in Figure 5.3. We set the maximum sequence length (number of tokens) to 128, as all our datasets contain less than 128 in tokens length.

We fine-tuned the model for 8 epochs, using AdamW (Loshchilov and Hutter, 2019) optimizer with the learning rate 1e-5 and Adam epsilon of 1e-8. We set the batch size to 8 for training and validation. We fine-tuned the model on a machine with an NVidia RTX 3090, Intel core i9 8950HK CPU, with 32G of RAM; usually, it took around 2 hours. In addition, we used Optuna (Akiba et al., 2019) for hyper-parameters search.

5.3.5 Inference

First, we performed token values search on the validation set that maximizes the Accuracy@1@top1 score using Optuna (Akiba et al., 2019). We searched the values ranging between 0.5 and 1.25; at each iteration, we changed the value by 0.05. We searched only WL and WR, whereas, for CR, we set it to 1.00 because we already knew that the best-ranking candidates were given the value of 1.00. Then we kept these values fixed for all sentences at the inference. Finally, at the inference, we set the beam search to 15 and the number of return sequences to 15 so that after filtering out some duplicate candidates, the remaining would be around 10. The ranking order of the candidates is chosen from the return orders of sequences produced by the model.

5.4 Results and Discussion

In Table 5.1 we present the results for the metrics: *Accuracy@1*, *Accuracy@K@Top1*, and *Potential@K*. In Table 5.2 we present the results for the metrics: *MAP@K*, *Precision@K*, and *Recall@K*. The results of ConLS presented here are based T5-Large.

Our experiments show that the modified LSBert had improved its *Accuracy@1* metric results with respect to the ones seen in the original LS-

Dataset	System	ACC@1	ACC@K@Top1					Potential@K				
			@1	@2	@3	@4	@5	@2	@3	@4	@5	@10
BenchLS	LSBert	**67.59**	**40.68**	**51.45**	57.37	59.84	61.57	**77.07**	**81.27**	83.32	84.28	85.47
	ConLS	62.00	37.99	51.34	**59.31**	**64.90**	**68.46**	74.92	**81.27**	**84.82**	**87.08**	**90.31**
NNSeval	LSBert	**44.76**	**28.03**	**38.49**	43.93	46.86	49.79	**59.00**	**64.85**	**67.78**	**71.55**	74.48
	ConLS	41.00	26.77	34.30	**45.18**	**50.20**	**52.71**	53.14	61.09	65.69	69.87	**79.08**
LexMTurk	LSBert	**84.80**	**44.00**	54.80	60.40	61.80	62.80	**91.00**	93.20	94.60	95.00	95.80
	ConLS	80.60	43.80	**56.39**	**65.40**	**71.20**	**76.60**	90.00	**95.60**	**97.40**	**98.20**	**99.60**

Table 5.1: The results of LSBert and ConLS for the metrics: Accuracy@1, Accuracy@K@Top1, and Potential@K.

Dataset	System	MAP@K					Precision@K			Recall@K		
		@2	@3	@4	@5	@10	@3	@5	@10	@3	@5	@10
BenchLS	LSBert	**52.26**	**42.29**	34.79	29.25	15.74	**46.46**	34.62	24.90	**25.74**	29.80	32.41
	ConLS	49.73	41.37	**35.01**	**30.54**	**18.84**	46.34	**37.11**	**26.20**	25.59	**32.25**	**41.89**
NNSeval	LSBert	**34.93**	**27.84**	23.18	19.97	10.73	32.84	26.16	18.78	**19.55**	23.40	26.14
	ConLS	31.69	27.31	**23.23**	**20.30**	**12.53**	**32.91**	**27.02**	**19.51**	18.80	**23.80**	**32.08**
LexMTurk	LSBert	**67.05**	54.41	45.83	39.01	21.29	58.03	45.25	33.43	20.52	24.61	27.52
	ConLS	65.45	**55.45**	**48.04**	**42.52**	**27.59**	**60.16**	**49.89**	**36.94**	**21.32**	**27.51**	**37.15**

Table 5.2: The results of LSBert and ConLS for the metrics: *MAP@K*, *Precision@K*, and *Recall@K*.

Bert paper (Qiang et al., 2021): *Accuracy@1* has improved from 79.20 to 84.80 for LexMTurk, from 61.60 to 67.59 for BenchLS, and from 43.60 to 44.76 for NNSeval. On the other hand, for the *Accuracy@1* metric the ConLS system does not improve the results of the modified LSBert system but improves the results of the original LSBert for the LexMturk and BenchLS datasets. The results of the *Accuracy@K@Top1* metrics show that the modified LSBert achieves better results at K=$\{1, 2\}$ and the ConLS achieves better results at K=$\{3, 4, 5\}$ for all datasets. This indicates that with more candidates allowed (3, 4, and 5 candidates) the ConLS is able to generate more instances with candidates within the top-1(s) gold annotated substitution(s) with respect to LSBert. The results of the *Potential@K* metric show these facts: 1) in LexMturk and BenchLS, the ConLS is outperforming LSBert gradually and increasingly from K=3 to K=10; 2) in NNSeval, ConLS improves the potential of LSBert only at K=10. For the MAP@K metric, we show that ConLS is able to improve the results of the metric at K=$\{4, 5, 10\}$ in all the datasets with respect

to the modified LSBert. Finally, the results of the *Precision@K* and *Recall@K* metrics show the same pattern: 1) for LexMTurk, ConLS outperforms the LSBert in all K=$\{3, 5, 10\}$; 2) for BenchLS and NNSEval, ConLS outperforms the LSBert only in K=$\{5, 10\}$.

T5 Model	ACC@1	ACC@K@Top1		
		@1	@2	@3
T5-Small	23.40	7.80	11.80	15.40
T5-Base	60.00	28.80	40.40	48.40
T5-Large	**80.60**	**43.80**	**56.39**	**65.40**

Table 5.3: The results of ConLS trained all tokens using different T5 models. The models were trained with TSAR-EN and evaluated with LexMTurk.

We also conducted a comparison on the effect of different T5 models trained with TSAR-EN and evaluated with LexMTurk. Table 5.3 shows that the T5-Large model performs a lot better than the T5-Base and the T5-Small models in all metrics (Accuracy@1, Accuracy@K@Top1). Therefore, we believe that the performance of our model would improve if we could go with a larger model, for example, T5-3b or T5-11b. We have tried with the T5-3b model, but unfortunately, it was unable to fit into our GPU memory (NVidia RTX 3090) even though we had set the batch size to as small as one.

To evaluate the effectiveness of the control tokens, we conducted further experiments with different sets of combinations. We trained and evaluated each set of tokens using T5-Large with TSAR-EN for training and LexMTurk for evaluation. The results in Table 5.4 have shown that the model trained with no tokens performs worse than the model with all tokens in all metrics, especially for the Accuracy@1@Top1 metric; the model with all tokens performs +2 points higher. Moreover, the **all tokens** model performs better than all other models in all metrics. This indicates that each token contributes to the selection and ranking of the candidates, which leads to better performance.

Tokens	ACC@1	ACC@K@Top1		
		@1	@2	@3
No Tokens	79.20	41.80	55.20	62.60
CR	79.00	41.00	54.40	62.60
WL	79.40	43.00	55.20	65.00
WR	78.60	41.20	54.60	63.20
CR+WL	78.40	41.40	54.40	62.40
CR+WR	78.60	42.80	54.60	62.20
WL+WR	78.60	41.00	54.20	62.20
All Tokens	**80.60**	**43.80**	**56.39**	**65.40**

Table 5.4: The results of ConLS trained with a different set of tokens. Each model was trained with TSAR-EN and evaluated with LexMTurk.

5.5 Conclusion

This chapter presents ConLS, the first approach for Controllable Lexical Simplification. The chapter also describes the evaluation of LSBert and ConLS for English with the LexMTurk, BenchLS, and NNSeval datasets for testing and the TSAR-EN dataset for training. The results of our evaluation show that the modified LSBert improves the *Accuracy@1* metric results with respect to the ones seen in the original LSBert paper in all three datasets. ConLS also improves it for the LexMturk and BenchLS datasets. Moreover, the ConLS system is able to achieve: 1) more potential to capture correct answers at K=$\{3, 4, 5, 10\}$ for BenchLS and LexMturk and at K=10 for NNSeval with respect to LSBert, 2) with more candidates retrieved (4 or 5) is able to generate more candidates within the top-1 more frequent gold-annotated suggestions with respect to LSBert, 3) with K=$\{5, 10\}$ candidates are able to generate (according to the gold-annotations) more correct and different candidates.

Chapter 6

Multilingual Controllable Transformer-Based Lexical Simplification

In Chapter 5, we presented a lexical simplification system for English, which lacks multilinguality; therefore, in this chapter, we present mTLS, a multilingual controllable Transformer-based lexical system for English, Spanish, and Portuguese. The novelty of this work lies in the use of language-specific prefixes, control tokens, and candidates extracted from pre-trained masked language models to learn simpler alternatives for complex words. The evaluation results on three well-known LS datasets – LexMTurk, BenchLS, and NNSEval – show that our model outperforms the previous state-of-the-art models like LSBert and ConLS. Moreover, further evaluation of our approach on the part of the recent TSAR-2022 multilingual LS shared-task dataset shows that our model performs competitively when compared with the participating systems for English LS and even outperforms the GPT-3 model on several metrics. Moreover, our model obtains performance gains also for Spanish and Portuguese.

6.1 Introduction

In Chapter 5, we introduced ConLS, the first controllable lexical simplification system fine-tuned with T5 using three tokens: Word Length token, Word Rank token, and Candidate Ranking token. The three tokens were used to control different aspects of the generated candidates: Word Length is often correlated with word complexity, Word Rank is the frequency order (word complexity is also correlated with frequency), and Candidate Ranking is for the model to learn how to rank the generated candidates through training. The model was fine-tuned with T5-large on TSAR-EN dataset (Saggion et al., 2022) and tested on LexMTurk (Horn et al., 2014), BenchLS (Paetzold and Specia, 2016a), and NNSeval (Paetzold and Specia, 2016b). However, this model lacks multilinguality.

In this work, we were interested in assessing knowledge transfer and multilinguality for lexical simplification. Due to the relatively small size of the available lexical simplification datasets for Spanish and Portuguese, it is quite challenging to train the model. To address this issue, we propose a multilingual lexical simplification model that jointly learns three languages simultaneously: English, Spanish, and Portuguese. We believe that the knowledge learned from one language could be shared with another; therefore, the model that is trained with a resource-rich language like English could be used to help other languages that have lower resources, like Spanish and Portuguese. In addition, we propose two additional tokens (Word Syllable and Sentence Similarity) on top of the three tokens from ConLS and Masked Language Model candidates to improve the model's performance.

We make the following contributions:

- We improve the English monolingual LS model and propose a new multilingual LS model for English, Spanish, and Portuguese[1].
- We show the way to fine-tune a multilingual LS model by adding language-specific prefixes, control tokens, and Masked Language

[1]The source code and data are available at https://www.github.com/kimchengsheang/mTLS

Model (MLM) candidates extracted from BERT-based pre-trained models.

- We conduct an analysis to capture the strengths and weaknesses of our approach.

6.2 Methodology

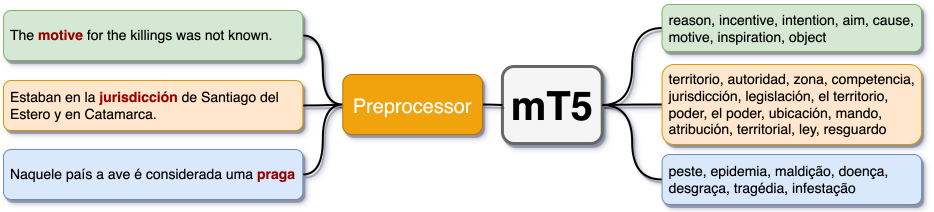


Figure 6.1: Illustration of the mTLS model with three simplification examples from the three languages.

Building upon the work of ConLS, we propose a new multilingual controllable transformer-based lexical simplification model that integrates language-specific prefixes alongside the control tokens and masked language model candidates to leverage the input-level information. We adopted the same three tokens from ConLS (Word Length, Word Rank, and Candidate Ranking) and incorporated two additional tokens (Word Syllables and Sentence Similarity), where word syllables correlate with word complexity and sentence similarity to guide the model to select relevant candidates based on semantic similarity. We fine-tuned our English monolingual model with T5 (Raffel et al., 2020) and multilingual model with mT5 (Xue et al., 2021). For more details about T5, see Section 5.1. mT5 is a multilingual model based on T5 trained on the multilingual colossal dataset (mC4), a dataset with over 100 languages also extracted from the public Common Crawl web scrape. Figure 6.1 shows an overview of our multilingual model where each input is a sentence with a complex word annotated, and the output is a list of substitutes ranked

from the most relevant and simplest to the least. The details of the Preprocessor are described in Section 6.3.4.

Language-specific Prefixes are embedded into each input so that the model knows and learns to differentiate the three languages. We used three prefixes: "simplify en:" for English, "simplify es:" for Spanish, and "simplify pt:" for Portuguese. In addition, these prefixes serve another purpose. Due to the limited data for Spanish and Portuguese, training individual models for Spanish and Portuguese would make the model unable to generalize well, so to tackle this issue, we jointly trained the three languages in just one model. This way, all the weights are learned and shared between the three languages during the training.

Control Tokens The following are the control tokens that were employed in our model to control different aspects of the generated candidates. Word Length, Word Rank (word frequency), and Word Syllables are known to be correlated well with word complexity, so we use them to help select simpler candidates. Candidate Ranking is used to help the model learn how to rank candidates through the training process so that, at the inference, the model could generate and sort candidates automatically, whereas Sentence Similarity is intended to help select relevant candidates based on semantic similarity.

- **Word Syllables (WS)** is the ratio of the number of syllables of the substitute divided by that of the complex word. It is extracted using PyHyphen library[2]. The study of Alva-Manchego et al. (2020a) shows that syllable count could help predict lexical complexity.
- **Sentence Similarity (SS)** is the normalized sentence similarity score between the source and the target sentence. The target sentence is the source sentence with the complex word replaced by its substitute. The score is calculated with the cosine similarity between the embeddings of the two sentences extracted from

[2]https://github.com/dr-leo/PyHyphen

Sentence-BERT (Reimers and Gurevych, 2019, 2020). This similarity score gives us a measure of the relation between the two sentences. In the experiments, we used the pre-trained model called "multi-qa-mpnet-base-dot-v1"[3] because it achieved the best performance on semantic search (tested on 6 datasets) and supported different languages such as English, Spanish, Portuguese, and more.

Masked Language Model (MLM) Candidates We believe that adding MLM candidates at the input level could give the model additional context on how to select better candidates than just repeating the same sentence twice as in LSBert. The candidates are extracted using the masked language model approach following the same style as LSBert candidates generation. For each input sentence and its complex word, we give the model (e.g., BERT, RoBERTa) the sentence and the same sentence with the complex word masked. E.g.,

> The **motive** for the killings was not known. </s> The **[MASK]** for the killings was not known.

We then ask the model to predict the [MASK] token candidates and rank them by the returned probability scores. After that, we select only the top-10 ranked candidates and append them to the end of each input. We believe that adding the MLM candidates to the input sentence could help the model find and select better candidates. More details about how we chose the best pre-trained model for each dataset are described in Section 6.3.4.

6.3 Experiments

In this section, we describe in detail all the datasets, baselines, evaluation metrics, data preparation steps, model details, training, and evaluation procedures.

[3] https://www.sbert.net/docs/pretrained_models.html

6.3.1 Datasets

Lang	Text	Target	Ranked Substitutes
EN	The motive for the killings was not known.	motive	reason:16, incentive:2, intention:2, aim:1, cause:1, motive:1, inspiration:1, object:1
ES	Estaban en la jurisdicción de Santiago del Estero y en Catamarca.	jurisdicción	territorio:5, autoridad:5, zona:3, competencia:2, jurisdicción:1, legislación:1, el territorio:1, poder:1, el poder:1, ubicación:1, mando:1, atribución:1, territorial:1, ley:1, resguardo:1
PT	Naquele país a ave é considerada uma praga	praga	peste:9, epidemia:5, maldição:3, doença:2, desgraça:2, tragédia:1, infestação:1

Table 6.1: Three examples from the TSAR-2022 shared-task dataset. Target is the complex word that is already annotated in the datasets. The number after the ":" indicates the number of repetitions suggested by crowd-sourced annotators.

As in the previous experiment, we use monolingual English datasets such as LexMTurk, BenchLS, NNSeval, and the multilingual dataset, TSAR-2022 shared task dataset. TSAR-2022 dataset contains three subsets: TSAR-EN for English, TSAR-ES for Spanish, and TSAR-PT for Brazilian Portuguese. Table 6.1 shows three examples from the TSAR-2022 dataset, one from each language, and Table 6.2 shows some statistics of the datasets. The average number of tokens (Avg #Tokens) shows that, on average, TSAR-ES has the longest text length, and TSAR-PT has the shortest text length.

All datasets that are used in the experiments already have complex words annotated, so the complex word identification module is not needed.

6.3.2 Baselines

We compare the proposed models with the following strong baselines, **LSBert** and **ConLS** as described in Chapter 5, and the systems from the TSAR-2022 shared task:

Dataset	Lang	#Instances	#Tokens		
			Min	Max	Avg
TSAR	EN	386	6	83	29.85
	ES	381	5	138	35.14
	PT	386	3	57	23.12
LexMTurk	EN	500	6	78	26.23
BenchLS	EN	929	6	100	27.90
NNSEval	EN	239	7	78	27.95

Table 6.2: Some statistics of the datasets.

- **CILS** (Seneviratne et al., 2022) generates candidates using language model probability and similarity score and ranks them by candidate generation score and cosine similarity.

- **PresiUniv** (Whistely et al., 2022) uses the Masked Language Model (MLM) for candidate generation and ranks them by cosine similarity and filters using the part-of-speech check.

- **UoM&MMU** (Vásquez-Rodríguez et al., 2022) uses a Language Model with prompts for candidate generation and ranks them by fine-tuning the Bert-based model as a classifier.

- **PolyU-CBS** (Chersoni and Hsu, 2022) generates candidates using MLM and ranks them by MLM probability, GPT-2 probability, sentence probability, and cosine similarity.

- **CENTAL** (Wilkens et al., 2022) generate candidates using MLM and ranks them by word frequency and a binary classifier.

- **teamPN** (Nikita and Rajpoot, 2022) generates candidates using MLM, VerbNet, PPDB database, and Knowledge Graph and ranks them by MLM probability.

- **MANTIS** (Li et al., 2022) generates candidates using MLM and ranks them by MLM probability, word frequency, and cosine similarity.
- **UniHD** (Aumiller and Gertz, 2022) uses prompts with GPT-3 (few-shot learning) for candidate generation and ranks them by aggregating the results.
- **RCML** (Aleksandrova and Brochu Dufour, 2022) generates candidates using lexical substitution and ranks them by part of speech, BERTScore, and SVM classifier.
- **GMU-WLV** (North et al., 2022) generates candidates using MLM and ranks them by MLM probability and word frequency.
- **TSAR-LSBert** is a modified version of the original LSBert to support Spanish and Portuguese and produce more candidates.
- **TSAR-TUNER** is an adaptive version of the TUNER system (a rule-based system) (Ferrés et al., 2017) for the TSAR-2022 shared task.

6.3.3 Evaluation Metrics

We adopt the same evaluation metrics used in TSAR-2022 shared task (Saggion et al., 2022) as described in Section 5.3.2.

6.3.4 Preprocessing

For each instance in the training set, there is a sentence, a complex word, and a list of ranked gold candidates. Thus, we compute the token values between the complex word and each candidate (we used all the candidates), which means if there are 9 candidates, there will be 9 training examples created.

Figure 6.2 shows the preprocessing steps of an English sentence taken from the TSAR-EN dataset. The sentence contains the complex word

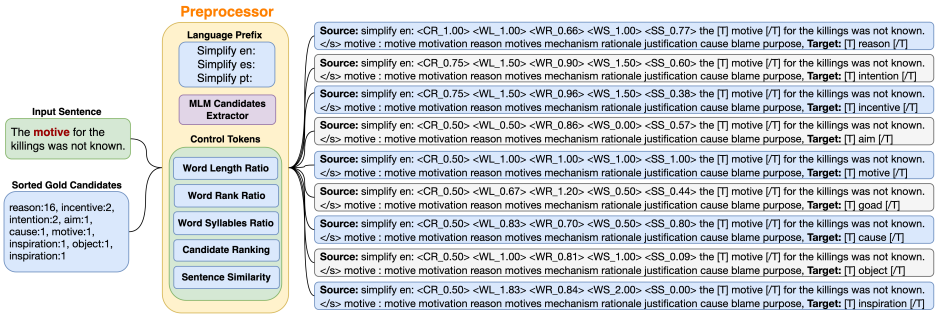


Figure 6.2: Preprocessing steps of an English training example. For Spanish and Portuguese, the process follows the same procedures.

"motive" and 9 ranked gold candidates; therefore, 9 training examples will be created. For each candidate and the complex word, we compute the tokens value, extract MLM candidates, and put all the values in the following format. Language prefix + Control Tokens + the input sentence with the complex word embedded in between [T] and [/T] + </s> + complex word + MLM candidates.

For Spanish and Portuguese datasets, we follow the same process and change the prefix to "simplify es:" for Spanish and "simplify pt:" for Portuguese.

For the validation set, we follow the same format as the training set, except all the token values are set with the values of 1.00. E.g., <CR_1.00> <WL_1.00> <WR_1.00> <WS_1.00> <SS_1.00>. We used these default values so that we could validate the model during the fine-tuning process and save the best model for evaluation.

To choose the best pre-trained models for MLM candidates extraction, we ran a series of experiments on some of the most popular BERT-based pre-trained models (the popularity is based on the number of downloads available on Huggingface website[4]). We compared them using the Potential metric since this metric measures the presence of the predicted

[4] https://huggingface.co/models

TSAR-EN		TSAR-ES		TSAR-PT	
Model	Potential	Model	Potential	Model	Potential
roberta-base	0.970	PlanTL-GOB-ES/roberta-large-bne	0.846	neuralmind/bert-large-portuguese-cased	0.835
bert-large-uncased	0.945	PlanTL-GOB-ES/roberta-base-bne	0.840	neuralmind/bert-base-portuguese-cased	0.808
bert-large-cased	0.939	dccuchile/bert-base-spanish-wwm-cased	0.818	xlm-roberta-large	0.616
roberta-large	0.939	dccuchile/albert-xxlarge-spanish	0.772	xlm-roberta-base	0.582
bert-base-uncased	0.936	dccuchile/albert-base-spanish	0.725	rdenadai/BR_BERTo	0.485
distilbert-base-uncased	0.915	dccuchile/distilbert-base-spanish-uncased	0.654	josu/roberta-pt-br	0.451
bert-base-cased	0.915	xlm-roberta-large	0.648	bert-base-multilingual-cased	0.390
albert-base-v2	0.863	dccuchile/bert-base-spanish-wwm-uncased	0.620		
xlm-roberta-large	0.780	bert-base-multilingual-uncased	0.580		
		distilbert-base-multilingual-cased	0.410		

Table 6.3: The comparison of different pre-trained models on candidate generation using masked language model ranked by Potential metric on TSAR dataset. Higher is better.

candidates, which are matched with the gold candidates. For each model and each instance of a dataset (we use only the training and development sets), we extracted the top 10 candidates and computed the Potential. Table 6.3 reports the results of the TSAR dataset, and Table 6.4 shows the results of the LexMTurk, BenchLS, and NNSeval dataset. We did the experiments on the top 5, 10, 15, 20, 30, 40, and 50 candidates, and we found that the top 10 candidates worked the best in all of our experiments. So, these are the selected models that produce the best score in each dataset: "roberta-base" for TSAR-EN, "PlanTL-GOB-ES/roberta-base-bne" for TSAR-ES, "neuralmind/bert-large-portuguese-cased" for TSAR-PT, "bert-large-cased" for LexMTurk and BenchLS, and "bert-base-uncased" for NNSeval.

6.3.5 Model Details

In the experiments, we fine-tuned four different models: TLS-1, TLS-2, TLS-3, and mTLS. Each model was fine-tuned with the language prefix, control tokens, and MLM candidates, except for the TLS-3 model, which was without the MLM candidates.

The following are the details of each model:

- TLS-1 is an English monolingual based on T5-large. It was fine-

LexMTurk		BenchLS		NNSeval	
Model	Potential	Model	Potential	Model	Potential
bert-large-cased	0.974	bert-large-cased	0.918	bert-base-uncased	0.887
bert-base-uncased	0.972	bert-large-uncased	0.909	roberta-base	0.883
bert-large-uncased	0.970	roberta-base	0.906	bert-large-uncased	0.879
roberta-base	0.970	bert-base-uncased	0.899	bert-base-cased	0.870
bert-base-cased	0.962	bert-base-cased	0.893	bert-large-cased	0.858
distilbert-base-uncased	0.950	distilbert-base-uncased	0.869	distilbert-base-uncased	0.791
xlm-roberta-large	0.934	albert-base-v2	0.850	albert-base-v2	0.762
albert-base-v2	0.926	roberta-large	0.830	roberta-large	0.745
roberta-large	0.904	xlm-roberta-large	0.813	xlm-roberta-large	0.711

Table 6.4: The comparison of different pre-trained models on candidate generation using masked language model ranked by Potential metric on LexMTurk, BenchLS, and NNSeval dataset. Higher is better.

tuned and validated with the TSAR-EN dataset (we split the dataset to 80% train, 20% validation) and then tested with LexMTurk, BenchLS, and NNSeval. This model is intended to compare with LSBert and ConLS.

- TLS-2 is an English monolingual based on T5-large. It was fine-tuned, validated, and tested on the same dataset (TSAR-EN). The dataset was split into a 70% train, a 15% validation, and a 15% test.

- TLS-3 (without MLM candidates) is an English monolingual based on T5-large. It was fine-tuned, validated, and tested on the TSAR-EN dataset. The dataset was split into a 70% train, a 15% validation, and a 15% test.

- mTLS is a multilingual based on mT5-large. It was fine-tuned, validated, and tested with the whole TSAR-2022 dataset (TSAR-EN, TSAR-ES, TSAR-PT). We split the dataset of each language into a 70% train, a 15% validation, and a 15% test. We then preprocessed, randomized, and combined the data of all languages into one training and one validation sets. During the fine-tuning process, the model is randomly fed with parallel data (the source and target data created by the preprocessing steps as shown in Figure 6.2) from

the three languages, allowing the model to learn and share all the weights.

- The model TLS-2, TLS-3, and mTLS are intended to compare with the models from the TSAR-2022 shared task. In order to have a fair comparison between our model and the shared-task models, we only compared the results of the same 15% test sets.

We implemented our approach using Huggingface Transformers library[5] and Pytorch-lightning[6]. Then we fine-tuned each model on an NVidia RTX 3090 GPU with a batch size of 4 (except mTLS, the batch size was set to 1 due to out-of-memory issues), gradient accumulation steps of 4, the max sequence length of 210 (it was based on the number of tokens/wordpiece from all datasets), learning rate of 1e-5, weight decay of 0.1, Adam epsilon of 1e-8. We fine-tuned it for 30 epochs, and if the model did not improve for four epochs, we saved the best model based on the highest validation score ACC@1@Top1 and stopped the fine-tuning process. All of our models took less than 15 epochs to converge. As in the previous experiments, we used Optuna to perform hyperparameters search on T5-small and T5-base to speed up the process and then employed the same hyperparameters in the final larger models like T5-large and mT5-large. For the generation, we used beam search and set it to 15 to generate 15 candidates so that it is left with around 10 candidates after some filtering (duplicate or the candidate the same as the complex word). In addition, in our experiments, the performance of the models based on T5-small and T5-base performed lower than the model based on T5-large in all metrics. The same with the multilingual models mT5-small, mT5-base, and mT5-large, so for that reason, we only report the results of the models that are based on T5-large and mT5-large.

[5]https://huggingface.co
[6]https://lightning.ai

6.3.6 Inference

For each model, we performed a tokens value search on the validation set of each corresponding dataset using Optuna (the same tool used for hyperparameters search). We searched the value of each token ranging between 0.5 and 2.0 with the step of 0.05, but we skipped the search for the Candidate Ranking token as we already knew the best value of it would be 1.00 to obtain the best candidates. We ran the search for 200 trials, then selected the top 10 sets of values that maximized ACC@1@Top1 and used them for the evaluation of the test set. For each set of tokens, we kept them fixed for all instances of the whole test set. Finally, we report the results of the set that maximized ACC@1@Top1. Figure 6.3 shows an example from the TSAR-EN test set and the simpler substitutes generated by our TLS-2 model.

Source: simplify en: <CR_1.00> <WL_1.25> <WR_1.05> <WS_1.60> <SS_1.00> #8-8 I want to continue playing at the highest level and win as many [T] **trophies** [/T] as possible. </s> **trophies** : trophies titles trophy competitions championships tournaments prizes awards cups medals
Predicted candidates: awards, medals, prizes, honors, accolades, titles, crowns, rewards, achievements, certificates

Figure 6.3: An example of the input taken from TSAR-EN test set and the candidates predicted by TLS-2 model.

6.4 Results and Discussion

In our experiments, we compared our model with all the systems submitted to the TSAR-2022 shared task on the TSAR dataset and the other two state-of-the-art models, LSBert and ConLS, on LexMTurk, BenchLS, and NNSeval datasets. We compared all of them with the same metrics

Dataset	System	ACC@1	ACC@1 @Top1	ACC@2 @Top1	ACC@3 @Top1	MAP @3	MAP @5	MAP @10	Potential @3	Potential @5	Potential @10
LexMTurk	LSBert	0.8480	0.4400	0.5480	0.6040	0.5441	0.3901	0.2129	0.9320	0.9500	0.9580
	ConLS	0.8060	0.4380	0.5639	0.6540	0.5545	0.4252	0.2759	0.9560	0.9820	0.9960
	TLS-1	**0.8580**	**0.4440**	**0.6080**	**0.7020**	**0.6629**	**0.5393**	**0.3591**	**0.9900**	**1.0000**	**1.0000**
BenchLS	LSBert	0.6759	0.4068	0.5145	0.5737	0.4229	0.2925	0.1574	0.8127	0.8428	0.8547
	ConLS	0.6200	0.3799	0.5134	0.5931	0.4137	0.3054	0.1884	0.8127	0.8708	0.9031
	TLS-1	**0.7255**	**0.4133**	**0.5952**	**0.6749**	**0.5187**	**0.4015**	**0.2539**	**0.8848**	**0.9257**	**0.9612**
NNSeval	LSBert	0.4476	0.2803	0.3849	0.4393	0.2784	0.1997	0.1073	0.6485	0.7155	0.7448
	ConLS	0.4100	0.2677	0.3430	0.4518	0.2731	0.203	0.1253	0.6109	0.6987	0.7908
	TLS-1	**0.5313**	**0.3263**	**0.4644**	**0.5397**	**0.3486**	**0.2762**	**0.1791**	**0.7824**	**0.8828**	**0.9414**

Table 6.5: Results of TLS-1 in comparison with LSBert and ConLS on the Accuracy@1, Accuracy@N@Top1, Potential@K, and MAP@K metrics. The best performances are in bold.

used in the TSAR-2022 shared task, such as ACC@1, ACC@N@Top1, Potential@1, and MAP@K where N $\in$ {1, 2, 3} and K $\in$ {3, 5, 10}.

Table 6.5 presents the results of our model TLS-1 (a monolingual English model fine-tuned and validated on the TSAR-EN dataset) in comparison with LSBert and ConLS on LexMTurk, BenchLS, and NNSeval datasets. Our model achieves better results in all metrics across the board, and the results on Potential@K and MAP@K show a significant improvement.

Table 6.6 shows the results of our three models, English monolingual models (TLS-2, TLS-3), and multilingual model (mTLS), compared with all the systems from the TSAR-2022 shared task on the TSAR-EN dataset. Since all the models from the shared task are unsupervised approaches, we only compare the results on the same 15% test set. Our TLS-2 outperforms all the models in all metrics and performs equally to GPT-3 model (UniHD) on ACC@1 and ACC@1@Top1; it also performs significantly better on ACC@{2,3}@Top1 and MAP@{3,5,10} but lower on Potential@{3,5}.

TLS-2 performs better than TLS-3 in all metrics except ACC@3@Top1, showing that adding MLM candidates does improve the model's performance.

Our multilingual model (mTLS) performs better than the previous ap-

Model	ACC @1	ACC@1 @Top1	ACC@2 @Top1	ACC@3 @Top1	MAP @3	MAP @5	MAP @10	Potential @3	Potential @5	Potential @10
TLS-2	**0.8750**	**0.5536**	**0.6964**	0.6964	**0.6379**	**0.5126**	**0.3069**	0.9643	0.9643	**1.0000**
TLS-3	0.8393	**0.5536**	0.6786	**0.7500**	0.5933	0.4506	0.2842	0.9643	0.9821	0.9821
mTLS	0.6607	0.3929	0.5000	0.6071	0.4871	0.3651	0.2173	0.8571	0.9286	0.9643
UniHD	**0.8750**	**0.5536**	0.6429	0.6786	0.5913	0.4055	0.2284	**1.0000**	**1.0000**	**1.0000**
UoM&MMU	0.6964	0.4107	0.5536	0.5714	0.4315	0.3234	0.2020	0.8393	0.8571	0.8929
RCML	0.6071	0.2321	0.4107	0.4821	0.3978	0.3032	0.1959	0.8214	0.9286	0.9464
LSBERT	0.5893	0.2679	0.4821	0.5714	0.4385	0.3136	0.1860	0.8750	0.9107	0.9286
MANTIS	0.5714	0.3036	0.4643	0.5179	0.4613	0.3463	0.2097	0.8393	0.9107	0.9464
GMU-WLV	0.5179	0.2143	0.2500	0.4107	0.3700	0.2936	0.1716	0.7321	0.8393	0.9107
teamPN	0.4821	0.1964	0.3571	0.3750	0.3065	0.2320	0.1160	0.6786	0.8036	0.8036
PresiUniv	0.4643	0.1786	0.2857	0.3214	0.3075	0.2417	0.1396	0.6607	0.7500	0.7857
Cental	0.4464	0.1250	0.2500	0.3393	0.3016	0.2210	0.1385	0.6607	0.7143	0.7857
CILS	0.4107	0.1786	0.2500	0.2679	0.2817	0.2198	0.1378	0.5893	0.6071	0.6250
TUNER	0.3929	0.1607	0.1607	0.1607	0.1865	0.1158	0.0579	0.4643	0.4643	0.4643
PolyU-CBS	0.3571	0.1607	0.2321	0.3036	0.2579	0.1887	0.1118	0.6250	0.7500	0.8214

Table 6.6: Official results from TSAR-2022 shared task in comparison with our models TSAR-EN dataset. The best performances are in bold.

proaches, except for UniHD. The fact that the model's performance is notably inferior to its monolingual counterparts could be attributed to the following facts. First, the use of a multilingual model can reduce performance, as it contains a lot of irrelevant information from other languages. Second, the mT5-large pre-trained model is significantly larger than the T5-large, with around 1.2 billion parameters compared to 737 million of the T5-large. Given the large number of parameters that need to be updated, the mT5-large model requires significantly more data to learn from; therefore, we could not fine-tune the mT5-large model individually for Spanish or Portuguese. We had to fine-tune a multilingual model (mTLS) by randomly feeding the data from the three languages, allowing the model to learn and share all the weights.

Table 6.7 and Table 6.8 present the results of our mTLS model in comparison with the TSAR-2022 official results on TSAR-ES and TSAR-PT datasets. Our model performs significantly better than all the participating systems in all metrics. However, there were unofficial results of UniHD that outperformed our mTLS model on TSAR-ES and TSAR-PT datasets.

Model	ACC @1	ACC@1 @Top1	ACC@2 @Top1	ACC@3 @Top1	MAP @3	MAP @5	MAP @10	Potential @3	Potential @5	Potential @10
mTLS	**0.5357**	**0.2857**	**0.3929**	**0.4821**	**0.3790**	**0.2852**	**0.1685**	**0.7500**	**0.8036**	**0.9107**
PolyU-CBS	0.4107	0.2143	0.2143	0.2143	0.2153	0.1479	0.0918	0.5000	0.5536	0.5893
GMU-WLV	0.3929	0.1786	0.2679	0.3036	0.2560	0.1945	0.1167	0.5714	0.6607	0.7321
UoM&MMU	0.3571	0.1964	0.2679	0.3214	0.2391	0.1699	0.0979	0.5714	0.6250	0.7143
PresiUniv	0.3214	0.1964	0.3214	0.3929	0.2361	0.1574	0.0860	0.6429	0.6786	0.7679
LSBERT	0.3036	0.0893	0.1429	0.1786	0.1994	0.1504	0.0910	0.4643	0.6250	0.7500
Cental	0.2679	0.1429	0.1786	0.2143	0.1865	0.1449	0.0851	0.5000	0.5536	0.5714
TUNER	0.1429	0.0714	0.1071	0.1071	0.0843	0.0506	0.0253	0.1964	0.1964	0.1964

Table 6.7: Official results from TSAR-2022 shared task in comparison with our model on the TSAR-ES dataset. The best performances are in bold.

Model	ACC @1	ACC@1 @Top1	ACC@2 @Top1	ACC@3 @Top1	MAP @3	MAP @5	MAP @10	Potential @3	Potential @5	Potential @10
mTLS	**0.6607**	**0.4464**	**0.5536**	**0.5714**	**0.4216**	**0.2940**	**0.1842**	**0.8214**	**0.9107**	**0.9464**
GMU-WLV	0.4464	0.2143	0.3750	0.4107	0.2579	0.1926	0.1143	0.6429	0.7679	0.8571
PolyU-CBS	0.3571	0.1071	0.1429	0.1607	0.1905	0.1455	0.0847	0.4643	0.5536	0.6071
Cental	0.3214	0.0714	0.1250	0.1964	0.2153	0.1554	0.0910	0.5714	0.6786	0.8214
LSBERT	0.3036	0.1607	0.2321	0.3036	0.1895	0.1364	0.0816	0.5179	0.6250	0.7321
TUNER	0.2321	0.1429	0.1607	0.1607	0.1071	0.0688	0.0344	0.2857	0.2857	0.2857
PresiUniv	0.2321	0.1071	0.1786	0.1964	0.1409	0.0952	0.0532	0.3750	0.4643	0.5179
UoM&MMU	0.1071	0.0357	0.0536	0.0714	0.0704	0.0553	0.0338	0.1964	0.2500	0.2857

Table 6.8: Official results from TSAR-2022 shared task in comparison with our model on TSAR-PT dataset. The best performances are in bold.

6.5 Conclusion

This chapter proposed a new multilingual Controllable Transformer-based Lexical Simplification that integrates language-specific prefixes alongside dynamic control tokens and masked language model candidates to leverage the input-level information. This approach allows us to have the candidate generation and ranking within one model as well as multilingual. Moreover, our method enables the model to learn more effectively on the complex word and to have finer control over the generated candidates, leading the model to outperform all the previous state-of-the-

art models in all datasets, including the GPT-3 model (UniHD) on some metrics.

Part IV

Sentence Simplification

Chapter 7

Controllable Sentence Simplification with a Unified Text-To-Text Transfer Transformer

In this chapter, we explore the use of T5 (Raffel et al., 2020) for sentence simplification in combination with a controllable mechanism to regulate the system outputs. The control mechanism is intended to help the model generate adapted text for different target audiences. Our experiments show that our model achieves remarkable results with gains of between +0.69 and +1.41 over the current state-of-the-art (BART+ACCESS) at the time of writing. We argue that using a pre-trained model such as T5, trained on several tasks with large amounts of data, can help improve Text Simplification.[1]

[1]The code and data are available at https://github.com/kimchengsheang/TS_T5

7.1 Introduction

Sentence simplification can be regarded as a natural language generation task where the generated text has a reduced language complexity in both vocabulary and sentence structure while preserving its original information and meaning. It is often tackled as a monolingual translation problem (Zhu et al., 2010; Coster and Kauchak, 2011b; Wubben et al., 2012) as explained in our Chapter 2, where the models are trained on parallel complex-simple sentences extracted from English Wikipedia and Simple English Wikipedia (Zhu et al., 2010).

Lately, there has been increased interest in conditional training with sequence-to-sequence models. It has been applied to some NLP tasks such as controlling the length and content of summaries (Kikuchi et al., 2016; Fan et al., 2018), politeness in machine translation (Sennrich et al., 2016), and linguistic style in text generation (Ficler and Goldberg, 2017). Scarton and Specia (2018) introduced the controllable text simplification model by embedding grade-level token <grade> into the sequence-to-sequence model. Martin et al. (2019) took a similar approach adding 4 tokens into source sentences to control different aspects of the output, such as length, paraphrasing, lexical complexity, and syntactic complexity. Kariuk and Karamshuk (2020) adopted the idea of using control tokens from Martin et al. (2019) and used it in an unsupervised approach by integrating those control tokens into the back translation algorithm, which allows the model to self-supervise the process of learning interrelations between a control sequence and the complexity of the outputs. The results of Scarton and Specia (2018), Martin et al. (2019), and Kariuk and Karamshuk (2020) have shown that adding control tokens does help improve the performance of sentence simplification models quite significantly.

In recent years, research in text simplification has been mostly focused on developing models based on deep neural networks (Vu et al., 2018; Zhao et al., 2018; Martin et al., 2019). However, to the best of our knowledge, very few studies of transfer learning –where a model is first pre-trained on a data-rich task and then fine-tuned on downstream tasks–

have been explored in text simplification.

In this chapter, we propose a controllable sentence simplification model with transfer learning that harnesses the power of the T5 pre-trained model (a Unified Text-to-Text Transfer Transformer) (Raffel et al., 2020), combining with control tokens to provide a way to generate outputs that adapt well to different target users without having to rebuild the model from the ground up.

We make the following contributions:

- We introduce a transfer learning approach combined with a controllable mechanism for sentence simplification task.
- We make an improvement to the performance of the sentence simplification system.
- We introduce a new control token W to help the model generate sentences by replacing long complex words with shorter alternatives.
- We conduct an evaluation and comparison between different sizes of pre-trained models and a detailed analysis of the effect of each control token.
- We show that by choosing the right control token values and pre-trained model, the model achieves state-of-the-art performance in two well-known benchmarking datasets.

7.2 Methodology

In this work, we fine-tune T5 pre-trained model with the controllable mechanism for sentence simplification. The details about T5 were described in Chapter 5.

7.2.1 Control Tokens

We adopt the tokens introduced in Martin et al. (2019) to control different aspects of the simplification, such as compression ratio, amount of paraphrasing, lexical complexity, and syntactic complexity. In addition, we propose a new token *word ratio*, which is used to control word length. We argue that word ratio is another important control token because normally word frequency correlates well with familiarity, and word length can be an additional factor as long words tend to be hard to read (Rello et al., 2013c), and corpus studies of original and simplified texts show that simple texts contain shorter and more frequent words (Drndarević and Saggion, 2012). Therefore, we add a word ratio to help the model generate simplified sentences with a similar amount of words and shorter word lengths. The *compression ratio* alone could help the model regulate sentence length but not word length.

- **C**: character length ratio between source sentence and target sentence. The number of characters in the target is divided by that of the source.
- **L**: normalized character-level Levenshtein similarity (Levenshtein, 1965) between the source and target.
- **WR**: inverse frequency order of all words in the target divided by that of the source.
- **DTD**: maximum depth of the dependency tree of the target divided by that of the source.
- **W**: the number of words ratio between source sentence and target sentence. The number of words in the target is divided by that of the source. Word frequency correlates well with familiarity, and word length can be an additional factor as long words tend to be hard to read (Rello et al., 2013c).

The first token (C) controls the compression level during simplification, the second token (L) controls the level of modifications performed,

the third token (WR) controls the lexical complexity at a word level, and the fourth token (DTD) controls the syntactic complexity of the sentence (Martin et al., 2019). Lastly, the fifth token (W) controls the length of words in the sentence (e.g., mathematics => math).

Figure 7.1 shows an example of a sentence is processed and embedded with control tokens for training.

Source
simplify: W_0.58 C_0.52 L_0.67 WR_0.92 DTD_0.71 In architectural decoration Small pieces of colored and iridescent shell have been used to create mosaics and inlays, which have been used to decorate walls, furniture and boxes.
Target
Small pieces of colored and shiny shell has been used to decorate walls, furniture and boxes.

Figure 7.1: An example of how the control tokens are embedded into the source sentence for training. The keyword **simplify** is added at the beginning of each source sentence to mark it as a simplification task.

7.3 Experiments

Our model is developed using the Huggingface Transformers library (Wolf et al., 2019)[2] with PyTorch[3] and Pytorch lightning[4].

7.3.1 Datasets

We use the WikiLarge dataset (Zhang and Lapata, 2017) for being the largest and most commonly used text simplification dataset containing

[2] https://huggingface.co/transformers/model_doc/t5.html
[3] https://pytorch.org
[4] https://pytorchlightning.ai

296,402 sentence pairs from automatically aligned complex-simple sentence pairs English Wikipedia and Simple English Wikipedia, which is compiled from Zhu et al. (2010); Woodsend and Lapata (2011); Kauchak (2013).

For validation and testing, we use TurkCorpus (Xu et al., 2016), which has 2000 samples for validation and 359 samples for testing, and each complex sentence has 8 human simplifications. We also use ASSET (Alva-Manchego et al., 2020a) for testing, which contains 2000/359 samples (validation/test) with 10 simplifications per source sentence.

7.3.2 Evaluation Metrics

Following previous research (Zhang and Lapata, 2017; Martin et al., 2020a), we use automatic evaluation metrics widely adopted in text simplification such as SARI, BLEU, and FKGL as described in Section 2.3. We compute SARI, BLEU, and FKGL using EASSE (Alva-Manchego et al., 2019)[5], a simplification evaluation library.

7.3.3 Training Details

We performed hyperparameters search using Optuna (Akiba et al., 2019) with the smallest T5 pre-trained model (T5-small) and reduced-size dataset to speed up the process. All models are trained with the same hyperparameters, such as a batch size of 6 for T5-base and 12 for T5-small, a maximum token of 256, a learning rate of 3e-4, weight decay of 0.1, Adam epsilon of 1e-8, 5 warm-up steps, 5 epochs, and the rest of the parameters are left with default values from Transformers library. Also, we set the random seed to 12 for reproducibility. For the generation, we use a beam search of 8. Our models are trained and evaluated using Google Colab Pro, which has a random GPU, T4 or P100. Both have 16GB of memory, up to 25GB of RAM, and a time limit of 24 hours per execution. Training of the T5-base model for 5 epochs usually takes around 20 hours.

[5] https://github.com/feralvam/easse

7.3.4 Choosing Control Token Values at Inference

Control tokens were intended to control the aspects of the simplifications for different target audiences; however, in this case, we use them to generate simplified outputs that maximize the SARI evaluation metric. To find the best control tokens values, first, we need to perform the values search. We use Optuna (the same tool used for hyperparameters search) to find the optimal values that produce the outputs with the best SARI score on the validation set and then keep those values fixed for all sentences in the test set. We repeat the same process for each evaluation dataset.

7.3.5 Baselines

We benchmark our model against several well-known state-of-the-art systems:

YATS (Ferrés et al., 22)[6] Rule-based system with linguistically motivated rule-based syntactic analysis and corpus-based lexical simplifier which generates sentences based on part-of-speech tags and dependency information.

PBMT-R (Wubben et al., 2012) Phrase-based MT system trained on a monolingual parallel corpus with candidate re-ranking based on dissimilarity using Levenshtein distance.

UNTS (Surya et al., 2019) Unsupervised Neural Text Simplification is based on the encode-attend-decode style architecture (Bahdanau et al., 2015) with a shared encoder and two decoders and trained on unlabeled data extracted from English Wikipedia dump.

Dress-LS (Zhang and Lapata, 2017) A Seq2Seq model trained with deep reinforcement learning combined with a lexical simplification model to improve complex word substitutions.

[6] http://able2include.taln.upf.edu

DMASS+DCSS (Zhao et al., 2018) A Seq2Seq model trained with the original Transformer architecture (Vaswani et al., 2017) combined with the simple paraphrase database for simplification (PPDB) (Pavlick and Callison-Burch, 2016).

ACCESS (Martin et al., 2019) Seq2Seq system trained with four control tokens attached to source sentence: character length ratio, Levenshtein similarity ratio, word rank ratio, and dependency tree depth ratio between source and target sentence.

BART+ACCESS (Martin et al., 2020a) The system fine-tunes BART (Lewis et al., 2020) and adds the simplification control tokens from ACCESS.

7.3.6 Results

We evaluate our models automatically on two different datasets TurkCorpus and ASSET. In addition, we also perform a human evaluation on one of our models, human evaluation – see Section 7.4. Table 7.1 reports the results of the automatic evaluation of our models compared with other state-of-the-art systems. Our model **T5-base**+C+WR+L+DTD performs best on TurkCorpus with the SARI score of 43.31, while the other model **T5-base**+All Tokens performs best on ASSET with SARI score of 45.04 compared to the current state-of-the-art BART+ACCESS with the SARI score of 42.62 on TurkCorpus and 43.63 on ASSET. Following these results, our models outperform all the state-of-the-art models in the literature in all approaches: rule-based, supervised, and unsupervised approaches, even without using any additional resources.

7.4 Human Evaluation

In addition to automatic evaluation, we perform a human evaluation of the outputs of different systems. Following recent works (Alva-Manchego

Model	Data	ASSET			TurkCorpus		
		SARI↑	BLEU↑	FKGL↓	SARI↑	BLEU↑	FKGL↓
YATS	Rule-based	34.4	72.07	7.65	37.39	74.87	7.67
PBMT-R	PWKP (Wikipedia)	34.63	79.39	8.85	38.04	82.49	8.85
UNTS	Unsup. Data	35.19	76.14	7.60	36.29	76.44	7.60
Dress-LS	WikiLarge	36.59	86.39	7.66	36.97	81.08	7.66
DMASS+DCSS	WikiLarge	38.67	71.44	7.73	39.92	73.29	7.73
ACCESS	WikiLarge	40.13	75.99	7.29	41.38	76.36	7.29
BART+ACCESS	WikiLarge	43.63	76.28	6.25	42.62	78.28	6.98
T5-base+C+WR							
+L+DTD	WikiLarge	44.91	71.96	6.32	**43.31**	66.23	6.17
T5-base+All Tokens	WikiLarge	**45.04**	71.21	5.88	43.00	64.42	5.63

Table 7.1: We report SARI, BLEU, and FKGL evaluation results of our model compared with others on TurkCorpus and ASSET test set (SARI and BLEU higher the better, FKGL lower the better). BLEU and FKGL scores are not quite relevant for sentence simplification, and we keep them just to compare with the previous models. All the results of the literature are taken from Martin et al. (2020a), except YATS which is generated using its web interface.

et al., 2017; Dong et al., 2019; Zhao et al., 2020), we run our evaluation on Amazon Mechanical Turk by asking five workers to rate using a 5-point Likert scale on three aspects: (1) Fluency (or Grammaticality) - *is it grammatically correct and well-formed?*; (2) Simplicity - *is it simpler than the original sentence?*; and (3) Adequacy (or Meaning preservation) - *does it preserve the meaning of the original sentence?*. For this evaluation, we randomly select 100 sentences from different systems trained on the WikiLarge dataset, except YATS, which is rule-based. The evaluation is first presented with the consent form (Appendix A.1) and then followed by the instructions (Appendix A.2) and finally, the simplification outputs from four systems displayed with random order (Appendix A.3). Table 7.2 reports the results.

Model	**Fluency**	**Simplicity**	**Adequacy**
YATS	4.03*	3.62*	3.92*
DMASS+DCSS	3.84*	3.70*	3.48*
BART+ACCESS	**4.41**	**4.02**	4.13
Our Model	4.30	3.99	**4.18**

Table 7.2: Results of human evaluation on 100 random sentences selected from TurkCorpus test set. Best results are marked in bold, and results marked with an '*' are significantly lower than our model according to paired t-test with $p<0.01$. The maximum value is 5, and the minimum is 1. Our model in use here is **T5-base+All Tokens**.

The results have shown that our model performs lower in fluency and about the same in simplicity and better in adequacy compared to BART+ACCESS. Based on our observation, there are two reasons that humans rated our model lower on fluency: (1) our model generates incorrect text format (without spaces) in some sentences (examples in Table 7.2). The problem can be easily spotted by humans, but it does not affect the automatic evaluation as EASSE uses a tokenizer that can split the whole sentence correctly. (2) Our model tends to produce longer sentences than BART+ACCESS, and in some cases, the subject is repeated

twice when the sentence is split into two (e.g., relative clause). Repetition is also considered one of the key features of simplification as it makes text easier to understand, but for native or fluent language speakers, repetition and the longer sentence make the fluency worse. Moreover, due to these problems, the evaluators also tend to lower the simplicity score as they consider it harder to read.

Sentence
So far the'celebrity'episodes have included Vic Reeves, Nancy Sorrell, and Gaby Roslin.
New South Wales'biggest city and capital is Sydney.

Figure 7.2: Examples of incorrect text format generated by our model.

7.5 Ablation Study

In this section, we investigate the contribution of each token and different T5 pre-trained models to the performance of the system. Table 7.3 reports the scores of models trained on WikiLarge and evaluated with TurkCorpus and ASSET test set. Table 7.4 shows all control token values used for all the models in Table 7.3 which are selected using the same process and tool as mentioned in Section 7.3.4.

Based on the results, the larger model (T5-base) performs better than the smaller one (T5-small) on both datasets (+3.06 on TurkCorpus, +4.3 on ASSET). It is due to the fact that a larger model has more information which could generate better and more coherent text. Moreover, when control tokens are added, the performance increases significantly. With only one token, WR performs best on TurkCorpus (+3.88 over T5-base) and L on ASSET (+7.43 over T5-base).

Using the pre-trained model alone does not gain much improvement; only when combined with control tokens, the results improve by a big

Model	ASSET			TurkCorpus		
	SARI↑	BLEU↑	FKGL↓	SARI↑	BLEU↑	FKGL↓
T5-small (No Tokens)	29.85	90.39	8.94	34.50	94.16	9.44
T5-small + All Tokens	39.12	86.08	6.99	40.83	85.12	6.78
T5-base (No Tokens)	34.15	88.97	8.94	37.56	90.96	8.81
T5-base:						
+ W	38.51	84.02	7.45	38.86	89.10	8.61
+ C	39.58	79.22	6.06	38.95	84.81	7.76
+ L	41.58	82.52	6.53	40.90	85.45	7.55
+ WR	41.40	76.75	5.85	41.44	85.46	7.67
+ DTD	40.08	81.94	6.56	39.18	87.60	7.81
T5-base:						
+ WR + L	42.85	80.38	4.47	41.75	83.90	7.42
+ C + WR + L	44.89	56.76	5.93	42.91	67.09	6.53
+ W + C + WR + L	44.65	58.52	5.52	43.03	68.11	5.96
+ C + WR + L + DTD	44.91	71.96	6.32	**43.31**	66.23	6.17
+ All Tokens	**45.04**	71.21	5.88	43.00	64.42	5.63

Table 7.3: Ablation study on different T5 models and different control token values. Each model is trained and evaluated independently. We report SARI, BLEU, and FKGL on TurkCorpus and ASSET test sets. Control token values corresponding to each model are listed in Table 7.4

margin (+3.06 and +9.28 for T5-small with and without tokens), and (+5.75 and +10.89 for T5-base with and without tokens).

7.5.1 Analysis on the effect of Word Ratio token (W)

Our goal of using the W control token is to make the model learn to generate shorter words, whereas C alone could help the model regulate the sentence length but not word length, so here we investigate how W and C control tokens affect the outputs.

Model	ASSET	TurkCorpus
T5-small (No Tokens)		
T5-small + All Tokens	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$	$W_{1.05}$ $C_{0.95}$ $WR_{0.85}$ $L_{0.85}$ $DTD_{0.85}$
T5-base (No Tokens)		
T5-base:		
+ W	$W_{0.75}$	$W_{0.85}$
+ C	$C_{0.5}$	$C_{0.75}$
+ L	$L_{0.75}$	$L_{0.85}$
+ WR	$WR_{0.25}$	$WR_{0.85}$
+ DTD	$DTD_{0.5}$	$DTD_{0.75}$
T5-base:		
+ WR + L	$W_{0.75}$ $L_{0.75}$	$W_{0.85}$ $L_{0.85}$
+ C + WR + L	$C_{0.95}$ $WR_{0.75}$ $L_{0.75}$	$C_{0.95}$ $WR_{0.85}$ $L_{0.85}$
+ W + C + WR + L	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$
+ C + WR + L + DTD	$C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$	$C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$
+ All Tokens	$W_{1.05}$ $C_{0.95}$ $WR_{0.75}$ $L_{0.75}$ $DTD_{0.75}$	$W_{1.05}$ $C_{0.95}$ $WR_{0.85}$ $L_{0.85}$ $DTD_{0.85}$

Table 7.4: These are the control token values used for the ablation study in Table 7.3. Each model is trained and evaluated independently. The values are selected using the hyperparameters search tool mentioned in Section 7.3.4.

Tokens	Model 1: **C_0.5** WR_0.75 L_0.75 DTD_0.75 Model 2: **W_1.0 C_0.5** WR_0.75 L_0.75 DTD_0.75
Source:	In order to accomplish their objective, surveyors use elements of geometry, engineering, trigonometry, **mathematics**, physics, and law.
Model 1:	In order to accomplish their objective, surveyors use geometry, engineering, and law.
Model 2:	In order to do this, surveyors use geometry, engineering, trigonometry, **math**, physics, and law.
Source:	The **municipality** has about 5700 inhabitants.
Model 1:	The municipality has 5700.
Model 2:	The **town** has about 5700.
Source:	A hunting dog refers to any dog who **assists** humans in hunting.
Model 1:	A hunting dog is any dog who hunts.
Model 2:	A hunting dog is a dog who **helps** humans in hunting.
Tokens	Model 1: **C_0.75** WR_0.75 L_0.75 DTD_0.75 Model 2: **W_1.0 C_0.75** WR_0.75 L_0.75 DTD_0.75
Source:	The park has become a traditional **location** for **mass demonstrations**.
Model 1:	The park has become a popular **place** for **demonstrations**.
Model 2:	The park has become a **place** for **people to show things**.
Source:	Frances was later **absorbed** by an **extratropical** cyclone on November 21.
Model 1:	Frances was later **taken** in by an extratropical cyclone.
Model 2:	Frances was later **taken** over by a cyclone on November 21.
Source:	There are claims that thousands of people were **impaled** at a single time.
Model 1:	There are claims that thousands of people were **killed**.
Model 2:	There are also stories that thousands of people were **killed** at a time.

Table 7.5: Examples showing the differences between the model with a number of words ratio versus the one without. Model 1 was trained with four tokens without W control token, and model 2 was trained with all five control tokens. All control token values used to generate the outputs are listed in the rows Tokens. We use bold to highlight the differences.

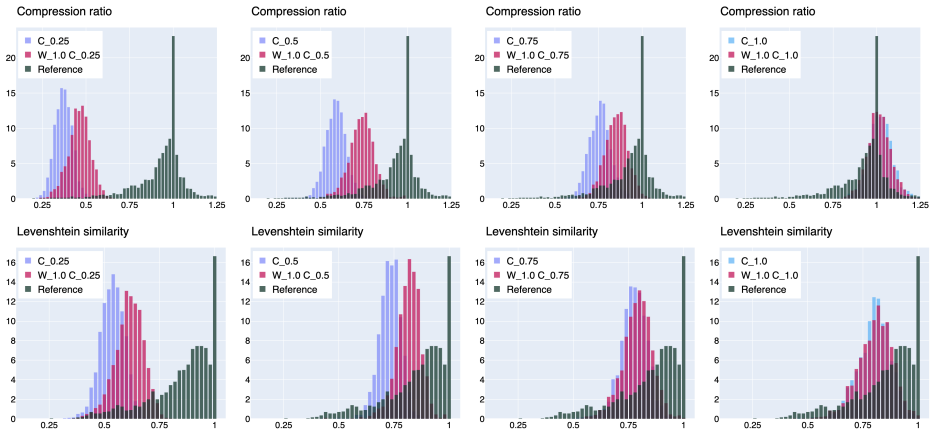


Figure 7.3: Influence of W and C control tokens on the simplification outputs. Red represents the outputs of the model trained with four tokens without W control token. Blue represents the outputs of the model trained with all five tokens. Green is the reference taken from TurkCorpus. The first row shows the compression ratio (number of chars ratio between system outputs and source sentences), and the second row is the Levenshtein similarity (words similarity between system outputs and source sentences) of each model. We plot the results of the 2000 validation sentences from TurkCorpus. Other control token values used here are set to 0.75, the example in Table 7.5.

For the model with the W token to work, it has to be incorporated with C, as W determines the number of words, and C limits the number of characters in the sentence. In our examples Table 7.5, we set W to 1.0, which means the number of words in the simplified sentence has to be similar to the original sentence, and C is set to 0.5 and 0.75, which means keeping the same amount of words but reduces 50% or 25% of characters.

Figure 7.3 shows the differences in density distribution (first row) and similarity (second row) between model 1 in red without W token, model 2 in blue with W tokens, and the one in green is the reference. The first column C is set to 0.25, the second column C=0.5, the third column C=0.75, the fourth C=1.0, and in all cases, W is set to 1.0. From the plots, we

can see that Model 1 does more compression than Model 2, which means Model 2 preserves more words than Model 1.

Table 7.5 shows some example sentences comparing models with C_0.75 and C_0.5. When C is set to 0.75, we do not see much difference between the two models, but when C is set to 0.5, the two models have differences in terms of sentence length and word length. For example, the word **mathematics** in example number one is replaced with the word **math** in model 2 (with W) and removed by model 1 (without W). Second example, the word **municipality** is replaced by the word **town** by model 2, and model 1 simply keeps the word and crops the sentence (the same problem with the third example). In addition, in the fourth example, the word **location** is replaced by both models with the word **place**, the phrase **mass demonstration** is reduced to **demonstration** by model 1, whereas model 2 changes to four shorter words **people to show things**.

There are many cases where model 1 and model 2 generate the same substitutions, but very often, model 1 tends to crop the end of the sentence or drops some words to fulfill the length constraint. Whereas model 2 tends to generate longer sentences than model 1, has less crop, and often replaces long complex words with shorter ones. Even though, based on the results from Table 7.1, adding the W control token does not significantly improve the SARI score and sometimes even lowers the score, it certainly holds its purpose.

7.6 Conclusion

This chapter proposed a method that leverages the technique of transfer learning by fine-tuning T5 for the controllable sentence simplification task. The experiments have shown good results of 43.31 SARI on the TurkCorpus evaluation set and 45.04 on the ASSET evaluation set, outperforming the current state-of-the-art models. Also, we have shown that adding the control token W is useful for generating substitutions with shorter word lengths. The only drawback of this approach is that it is slow to run, requiring resource-intensive hardware for training and inference.

Chapter 8

Sentence Simplification Capabilities of Transfer-Based Models

In Chapter 7, we presented the controllable sentence simplification for English. In this chapter, we explore the approach further by fine-tuning different models based on T5, mT5 (Xue et al., 2021), and mBart (Liu et al., 2020b) for English and Spanish. We also propose specifications for expert evaluation that are in accordance with well-established easy-to-read guidelines. We conduct expert evaluations of our new systems and the previous state-of-the-art systems for English and Spanish and discuss the strengths and weaknesses of each of them. Finally, we draw conclusions about the capabilities of the state-of-the-art sentence simplification systems and give some directions for future research.

8.1 Introduction

In this work, we aim to extend the approach proposed in Chapter 7 to support both English and Spanish. This expansion is motivated by the need to cater to a wider audience and enhance the usability and applicability of the model. By incorporating support for multiple languages, we can

address the needs of users in either English or Spanish.

To evaluate the performance of the model, we employ a comprehensive evaluation methodology that goes beyond traditional automatic metrics. While automatic metrics provide a quick quantitative assessment of the model's performance, they may not capture the nuances and complexities of the simplified sentences. Therefore, we employ crowd-sourced human evaluation (see Section 8.3.3) and expert evaluation (see Section 8.3.4)) in addition to automatic metrics.

We make the following contributions:

- We propose new transformer-based sentence simplification systems for English and Spanish that, according to an extensive multi-facet evaluation, show state-of-the-art performances for both languages.[1]
- We conduct crowd-sourced human evaluations for English and Spanish.

8.2 Methodology

In this work, we use three transformer-based models:

- **mBart** (Liu et al., 2020b): a multilingual sequence-to-sequence model based on BART (Lewis et al., 2020), trained as a denoising auto-encoder, using random span masking and sentence shuffling on a subset of 25 languages from XLM-R dataset (Conneau et al., 2020).
- **T5** (Raffel et al., 2020): more details see Section 5.1.
- **mT5** (Xue et al., 2021): a multilingual model based on T5 trained on the multilingual colossal dataset (mC4), a dataset with over 100 languages also extracted from the public Common Crawl web scrape.

[1]The code and data is available at https://github.com/kimchengsheang/TS-AAAI_2022

The T5 and mT5 models are available in different sizes, depending on the number of attention modules and the number of parameters. Due to memory limitations and time constraints, we are able only to use T5-base for English. For Spanish, we use mT5-base and mT5-large. We implement the models using Huggingface Transformers library[2] (Wolf et al., 2020) with PyTorch[3] and Pytorch lightning.[4]

We adopt the four control tokens (C, L, WR, DTD) from the previous model described in Section 7.2.1. We set as the optimal values of the control tokens those values that lead to the highest SARI score on the validation datasets. The search for the optimal values of control tokens is done using Optuna for each language and dataset separately.

8.3 Experiments

In this section, we describe the training details, the datasets, automatic evaluation metrics, and expert evaluation guidelines.

8.3.1 Training Details

We train the three models following the same procedures and configurations as described in Chapter 7 Section 7.3.3. Except we train mBART and mT5 on our own computers due to the limited resources of Google Colab. We only change the batch size to adapt to the GPU memory. Our computer has an Intel core i9 8950HK, 32GB of memory, and an NVidia RTX 3090 GPU (24GB).

8.3.2 Datasets

For English, we use Wiki-Large (Zhang and Lapata, 2017) dataset for training. Wiki-Large is the largest and most commonly used dataset for

[2]https://huggingface.co/transformers/model_doc/t5.html
[3]https://pytorch.org
[4]https://pytorchlightning.ai

English sentence simplification. It contains 296,402 sentence pairs from automatically-aligned complex-simple sentence pairs from document-aligned English Wikipedia and Simple English Wikipedia articles. For validation and testing in English, we use two datasets: MTurk (Horn et al., 2014) and ASSET (Alva-Manchego et al., 2020a). Both datasets contain 2,000 samples for validation and 359 samples for testing. In the MTurk dataset, each sample contains an original sentence from English Wikipedia and eight simplifications of that sentence by eight Amazon Mechanical Turk workers. In ASSET, each sample contains an original sentence from English Wikipedia and ten manual simplifications. The original sentences are the same in both datasets.

For Spanish, we use automatically-aligned sentence pairs (Štajner et al., 2018) from the original and manually simplified Newsela corpus which comprises original news articles, manually simplified to several simpler levels by professional editors. The complex-simple sentence pairs were aligned using the CATS tool (Štajner et al., 2017, 2018) built especially for that purpose.[5] As the alignments of sentences between further-apart complexity levels are less reliable (Štajner et al., 2018), we only use the alignments between the original articles and the first level of simplification. The correctness of these alignments is estimated to be 96.1% for the recommended C3G sentence-level alignment (Štajner et al., 2018). From all aligned sentence pairs, we randomly selected 700 sentence pairs for validation and 350 for testing. The rest (7,414 sentence pairs) we use for training.

It is important to note that our English models are trained on a significantly larger dataset (296,402 sentence pairs) than the Spanish models (7,414 sentence pairs) and that the English simple sentences are significantly shorter than the Spanish simple sentences in the datasets used (Table 8.1).

[5]https://github.com/neosyon/SimpTextAlign

Dataset	Original	Simple
ES-train-Newsela	29.89	30.61
ES-dev-Newsela	30.56	30.75
ES-test-Newsela	30.64	30.48
EN-train-WikiLarge	25.68	18.86
EN-dev-MTurk	22.07	21.07
EN-test-MTurk	22.82	21.79
EN-dev-ASSET	21.93	19.27
EN-test-ASSET	22.60	18.89

Table 8.1: Average sentence length (in tokens) for different parts of the datasets.

8.3.3 Standard Evaluation

To compare our systems with a larger number of previously proposed systems, we use the SARI metric (Xu et al., 2016) implemented in EASSE (Alva-Manchego et al., 2019), a simplification evaluation library. SARI compares system outputs to the references and the source sentence by counting words that are added, deleted, and kept.

To compare our best systems with the best previous systems (according to SARI) with different architectures, we perform a crowd-sourced human evaluation of grammaticality (G), meaning preservation (M), and simplicity (S) on a 1–5 Likert scale, by five Amazon Mechanical Turk workers who are native speakers of the respective language (English or Spanish). We follow the same procedure as in other studies that perform this type of evaluation, e.g. (Martin et al., 2022). The annotators are first provided with the consent form and then the instructions and instances for evaluation. For each instance, they are provided with the original sentence and the three simplified versions. For each simplified version, they are asked to judge how much they agree (1–strongly disagree, 5–strongly agree) with the following statements (used to assess G, M, and S, respectively):

- The sentence is grammatically correct and well-formed.
- The sentence has the same meaning as the original one.
- The sentence is simpler than the original one.

For English, we adopt the same interface as in the previous study Section 7.4 and adapt the translation for Spanish.

8.3.4 Expert Evaluation

To better assess simplifications performed by different sentence simplification systems and their compliance with easy-to-read guidelines, we propose a novel expert evaluation and a detailed set of rules on how to judge whether the transformation made by the system results in a simpler form (Table 8.2). For each language, we ask two expert annotators to perform the assessment. We provide them with the above-mentioned set of rules and an online editing tool that highlights the differences between the original and simplified sentences.

The annotators are asked to count several types of lexical and syntactic transformations and judge their correctness. The transformation is correct if it satisfies all three conditions: (1) preserves the original meaning; (2) is grammatical; and (3) results in a simpler phrase/sentence(s) according to the evaluation guidelines provided. If conditions (1) and (3) are satisfied, but the transformation results in a small grammatical error (e.g., verb in plural instead of the singular form), the transformation is semi-correct and receives a 0.5 score (instead of 1 for a complete correct transformation). The annotators are instructed to separately count and evaluate phrase-level lexical transformations (everything beyond uni-grams on either source or target side), sentence splitting, reordering within a clause, removal and addition of information. For each pair of original-simplified sentences, the annotators are requested to assign a meaning preservation score and simplicity score on a 1–5 scale (Tables 8.4 and 8.5).

The annotators are requested to compare their results, reach a consensus, and provide us with their final joint result. Several examples with

correct and incorrect transformations are presented in Table 8.3. According to the guidelines for assigning simplicity and meaning preservation scores (Tables 8.4 and 8.5), the first sentence automatically simplified by the MUSS-sup system in Table 8.3 would get the score 3 for meaning preservation (as it lost the essential information that *Graham graduated with BA in anthropology*) and score 5 for simplicity (due to sentence splitting, lexical simplification, and less information to process).

Rule	Simpler form...
1	Uses active tense instead of passive
2	Uses the simplest form of the verb (simple present or past tense instead of conditionals or future)
3	Avoids hidden verbs (i.e. verbs converted into a noun)
4	Avoids abbreviations
5	Uses shorter and/or more commonly used words
6	Omits unnecessary words
7	Uses the same term consistently
8	Avoids legal, technical, or foreign jargon
9	Simplifies punctuation
10	Makes the sentence(s) shorter
11	Keeps subject, verb, and object close together
12	Avoids double negatives and exceptions to exceptions
13	Places the main idea before the exceptions and conditions
14	Covers only one main idea per sentence
15	Avoids figures of speech and metaphors
16	Uses number instead of the word

Table 8.2: Guidelines for expert annotation, based on the Plain Language guidelines (PlainLanguage, 2011), “Make it simple" guidelines (Freyhoff et al., 1998b), and “Am I making myself clear?" guidelines (Mencap, 2002).

Transformations	Original	Automatically Simplified	System
lexical, **splitting**, **addition** ('In 1943,')	Graham **attended** Wheaton College from 1939 to 1943**, when** he **graduated with a BA** in anthropology.	Graham **went** to Wheaton College from 1939 to 1943**. In 1943,** he **got a degree** in anthropology.	T5-base
lexical, **splitting** (*missing information*)	Graham **attended** Wheaton College from 1939 to 1943**, when** he graduated *with a BA in anthropology*.	Graham **went** to Wheaton College from 1939 to 1943**. H**e graduated from Wheaton College in 1943.	MUSS-sup
addition (','), **lexical-phrase reordering**, *lexical* (*missing information*)	In **1987** Wexler was **inducted into** the Rock and Roll Hall of Fame.	In **1987,** Wexler was **added to** the Rock and Roll Hall of Fame.	T5-base
	In 1987 *Wexler* was inducted into the Rock and Roll Hall of Fame.	*He* was inducted into the Rock and Roll Hall of Fame **in 1987**.	MUSS-sup

Table 8.3: Automatic English sentence simplification performed by our system (T5-base) versus the state of the art (MUSS-sup). Correct transformations are marked in bold, whereas incorrect transformations (lost or changed meaning) are marked in italics.

Score	Definition
1	Simplified sentence is meaningless.
2	Simplified sentence has a completely different meaning from the original.
3	Meaning has not been changed but some essential information is missing.
4	Meaning is almost the same; there are some minor differences that are not essential.
5	Meaning is fully kept (some nuances might have been lost due to deletion of non-essential information).

Table 8.4: Definition of meaning preservation scores in the expert evaluation.

Score	According to the rules in Table 8.2...
1	... original sentence is much easier to understand than the simplified one.
2	... original sentence is somewhat easier to understand than the simplified one.
3	... both sentences are equally easy/difficult to understand.
4	... simplified sentence is somewhat easier to understand than the original one.
5	... simplified sentence is much easier to understand than the original one.

Table 8.5: Definition of simplicity scores in the expert evaluation.

8.4 Results and Discussion

8.4.1 English Sentence Simplification

Standard Evaluation. We use SARI score to automatically compare our systems with previously proposed state-of-the-art sentence simplification systems with various architectures: the rule-based YATS system (Ferrés et al., 22), phrase-based MT (Wubben et al., 2012), encoder-decoder model (LSTM) with reinforcement learning Dress-LS (Zhang and Lapata, 2017), original-transformer-based model DMASS+DCSS (Zhao et al., 2018), the original transformer-based model with control tokens ACCESS (Martin et al., 2019), and transformer-based model (BART) with control tokens MUSS-sup (Martin et al., 2022). The ACCESS and MUSS-sup systems use the same four control tokens as our models (mBART, mT5-base, and T5-base) and the same training dataset. The only difference among those five systems (ACCESS, MUSS-sup, mBART, mT5-base, and T5-base) is the transformer model that is used. Our T5-base system achieves a higher SARI score than all previously proposed systems on both test sets (Table 8.6). Overall, the results show the superiority of transformer-based models with control tokens over all other approaches.

We further perform a human evaluation of grammaticality, meaning preservation, and simplicity, by five Amazon Mechanical Turk workers (all native English speakers) of several systems: our T5-base (as the best performing system), MUSS-sup (as the best performing previous system),

System	Type	ASSET	MTurk
YATS	rule-based	34.4	37.4
PBMT-R	phrase-based MT	34.6	38.0
Dress-LS	LSTM+reinfor.	36.6	37.0
DMASS+DCSS	transformer	36.7	39.9
ACCESS	transf.+control	40.1	41.4
MUSS-sup	BART+control	**43.6**	**42.6**
mBART (our)	mBART+control	40.4	41.4
mT5-base (our)	mT5+control	42.0	41.2
T5-base (our)	T5+control	**44.9**	**43.3**

Table 8.6: SARI scores for English sentence simplification on two test sets (ASSET and MTurk), each with 359 instances. Higher scores indicate better outputs.

and YATS (as the rule-based system).

The results are presented in Table 8.7. The output of MUSS-sup and our T5-base are rated similarly. Both systems produce simpler and more grammatical sentences than YATS.

System	G	M	S
YATS	$3.58^{*}_{\pm 0.14}$	$3.54_{\pm 0.14}$	$3.25^{*}_{\pm 0.13}$
MUSS-sup	$\mathbf{3.99}_{\pm 0.13}$	$3.54_{\pm 0.13}$	$3.66_{\pm 0.12}$
T5-base (our)	$3.91_{\pm 0.12}$	$\mathbf{3.58}_{\pm 0.13}$	$\mathbf{3.68}_{\pm 0.12}$

Table 8.7: Human evaluation scores (mean value with 95% confidence interval) for English on 50 randomly selected MTurk test instances. Higher scores indicate better outputs. Results marked with an '*' are significantly lower than the best ones (paired t-test; $p<0.01$).

Expert evaluation. The results of the expert evaluation for English, performed on the same instances used for the crowdsourced human evaluation, are presented in Table 8.8. T5-base outperforms MUSS-sup by

almost all metrics. The main issue found with MUSS-sup is the removal of essential parts which results in a lower overall meaning preservation score (M). Two instances that illustrate those phenomena were presented earlier in Table 8.3. The fewer number of lexical simplifications found in MUSS-sup and the higher percentage of errors among those led to a lower overall simplicity score (S). Among the additions made by MUSS-sup, only one was a hallucination: *"...on the steps of Michigan Union."* $\rightarrow$ *"...on the steps of Michigan Union University."*. The addition performed by MUSS-sup in one case led to a transformation of a sentence in the present tense into a hypothetical sentence. All other additions made by MUSS-sup were correct. They were necessary to preserve grammaticality during reordering and sentence splitting. Among the additions made by T5-base, we found only one case of hallucination.

System	Lexical-all		Lexical-phrase		Reorder		Split		Remove		Add		Same	M	S
	All	Corr.	All	Corr.	All	Corr.	All	Corr.	All	Corr.	All	Corr.			
T5-base	**80**	**91%**	**49**	**90%**	**20**	**70%**	**17**	94%	**22**	**59%**	**16**	**87%**	**2%**	**4.2**	**4.3**
MUSS-sup	69	77%	34	82%	6	50%	16	**100%**	16	41%	7	71%	**2%**	4.1	3.8

Table 8.8: Results of the expert analysis for English, done on 50 randomly selected instances from the MTurk test set, for two best-performing systems (both systems were analyzed for their output on the same 50 instances). The columns *Corr.* show the percentage of all cases of the respective category that were marked as correct. The column *Same* shows the percentage of sentences that were not changed by the system. Better scores in each category are presented in bold. Differences in M and S scores for the two systems are not statistically significant (Wilcoxon's sign rank test; $p< 0.01$).

Overall, we found that both systems perform a range of distinct simplification operations. For each of the 16 rules from Table 8.2, we found at least one example of a simplified sentence that is simpler than the original according to that rule in the output of each system. For example, we found two cases of passive to active voice conversion (e.g. *"Fives is a British sport believed to...* $\rightarrow$ *"Fives is a British sport. Many people think..."*) performed by T5-base, and one by MUSS-sup. All three were

correct.

When interpreting the results in Table 8.8, it is important to remember that the only difference between the architectures used in T5-base and MUSS-sup is the transformer model (T5-base vs. BART). Both systems are trained with the same Wiki-Large dataset and use the same four control tokens. Interestingly, we only found two instances for which both systems produced identical outputs.

8.4.2 Spanish Sentence Simplification

Standard Evaluation. For Spanish sentence simplification, we calculate SARI scores on the test set (350 instances) for the output of our three systems (mT5-base, mT5-large, and mBART), and the only two previously proposed fully-fledged systems: the rule-based system Simplext (Saggion et al., 2015), and the unsupervised MUSS system (Martin et al., 2022) which uses the combination of mBART with four control tokens (Table 8.9). The only difference between our mBART system and MUSS-unsup is that our system was trained with complex-simple sentence pairs from Spanish Newsela (7,414 sentence pairs), whereas the MUSS-unsup was trained with the web-mined paraphrases (996,609 sentence pairs). The mBART, mT5-large, and MUSS-unsup all achieve similar SARI scores, noticeably higher than those of the other two systems. Among them, the MUSS-unsup obtains the highest average scores for G, M, and S in the crowdsourced human evaluation (Table 8.10). However, the differences in G, M, and S scores between any pair of systems were not statistically significant.

Expert Evaluation. The results of the expert evaluation for Spanish are presented in Table 8.11. In comparison to the English T5-base system, the Spanish mT5-large system makes noticeably fewer lexical simplifications and sentence splittings and has a higher percentage of erroneous ones. Both phenomena are very likely the result of a much lower number of training instances for Spanish (7,414, as opposed to 296,402 for English) and the use of the multilingual instead of the monolingual transformer model.

System	Type	SARI
Simplext	rule-based	33.5
MUSS-unsup	mBART+control	**36.8**
mT5-base (our)	mT5+control	32.7
mT5-large (our)	mT5+control	**36.9**
mBART (our)	mBART+control	**37.1**

Table 8.9: Results of Spanish sentence simplification.

System	G	M	S
MUSS-unsup	$\mathbf{4.52}_{\pm 0.11}$	$\mathbf{3.96}_{\pm 0.17}$	$\mathbf{3.51}_{\pm 0.16}$
mT5-large (our)	$4.43_{\pm 0.13}$	$3.81_{\pm 0.18}$	$3.19_{\pm 0.18}$
mBART (our)	$4.38_{\pm 0.13}$	$3.86_{\pm 0.17}$	$3.19_{\pm 0.16}$

Table 8.10: Human evaluation scores (mean value with 95% confidence interval) for Spanish on 50 randomly selected test instances. Higher scores indicate better outputs. The differences in scores are not statistically significant (paired t-test; $p<0.01$) for any pair of systems.

According to the expert evaluation, MUSS-unsup performs more lexical simplifications than the other two Spanish sentence simplification models (especially mT5-large). However, those lexical transformations are found to be correct only in half of the cases (52%). The high percentage of errors made by MUSS-unsup resulted in noticeably lower average meaning preservation (M) and simplicity (S) scores. The most conservative system (mT5-large), which leaves 34% of the sentences unchanged, achieves the highest simplicity score among the three systems. Here is important to note that mBART and MUSS-unsup architectures differ only in the datasets they were trained with, their size and quality. MUSS-unsup was trained with a large number of web-mined paraphrases (996,609 sentence pairs), while mBART was trained with only 7,414 sentence pairs from a high-quality Newsela dataset. These results indicate that, in transformer-based sentence simplification with these four control tokens, the size and quality of the training set strongly influence the number of transformations and their variety.

System	Lexical-all		Lexical-phrase		Reorder		Split		Remove		Add		Same	M	S
	All	Corr.	All	Corr.	All	Corr.	All	Corr.	All	Corr.	All	Corr.			
mT5-large	20	40%	5	40%	1	0%	**2**	**50%**	26	31%	4	0%	34%	3.4	**3.6**
mBART	40	45%	12	42%	0	NA	**2**	**50%**	22	57%	**6**	**83%**	24%	**3.6**	3.3
MUSS-unsup	**53**	**52%**	**27**	**57%**	33	24%	3	0%	**23**	**59%**	1	0%	**2%**	3.2	3.1

Table 8.11: Results of the expert analysis for Spanish, done on 50 randomly selected instances from the test set for three systems (the same 50 instances for all three systems). The columns *Corr.* show the percentage of all cases of the respective category that were marked as correct. The column *Same* shows the percentage of sentences that were not changed by the system. Better scores in each category are presented in bold. Differences in M and S are not significantly different (Wilcoxon's sign rank test; $p < 0.01$) for any pair of systems.

8.5 Conclusion

Automatic sentence simplification is envisioned to play a significant role in making everyday texts more accessible for wider populations, thus ensuring their better social inclusion. In this study, we proposed several state-of-the-art sentence simplification systems for English and Spanish, using recently proposed transformer-based models coupled with a simplification control mechanism. We also proposed guidelines for expert human evaluation which takes into account recommendations for easy-to-read texts.

The extensive evaluation showed that the proposed systems perform state-of-the-art sentence simplification in both English and Spanish and that transformer-based systems with the chosen four-token control mechanism produce sentences that are simpler than the originals according to easy-to-read guidelines. All investigated transformer-based systems performed a wide range of simplification operations which led to simpler output according to easy-to-read guidelines. In English sentence simplification, the results of the expert evaluation indicate that the use of T5 leads to a higher number of simplification operations and a higher number of correct transformations than the use of mBART. In Spanish sentence simplification, the results of the expert evaluation indicated that the size and the quality of the training data have an influence on the correctness of some transformations.

Part V

Conclusions and Future Work

Chapter 9

Conclusions and Future Work

This thesis reported a comprehensive study and the systems we have proposed for automatic text simplification. We have presented different approaches for complex word identification, lexical simplification, and sentence simplification with the goal of creating a robust automatic text simplification system. The technologies we have developed could be helpful for creating applications to improve readability and comprehension for individuals with autism, dyslexia, aphasia, children, second-language learners, deaf and hard of hearing, and others with intellectual disabilities.

In Chapter 4, two experiments were conducted to identify complex words. The first experiment utilized CNN with a set of engineered features and non-contextualized word embeddings to identify complex words in English, Spanish, and German texts. The second experiment explored the use of additional features, including contextualized and non-contextualized word embeddings and different word frequency lists. We trained the model with three algorithms: CNN, CatBoost, and XGBoost. The experiments have shown that the CNN model works better than ensemble algorithms, CatBoost and XGBoost. Moreover, our CNN model performs comparable results to the state-of-the-art system for English and achieves state-of-the-art results for Spanish and German despite having very few features.

In Chapter 5 and 6, we presented two experiments on lexical simplifi-

cation, which led to two state-of-the-art systems, an English lexical simplification and a multilingual lexical simplification for English, Spanish, and Portuguese. Chapter 5 presented the first approach to monolingual controllable lexical simplification, and Chapter 6 presented more experiments on both monolingual and multilingual controllable lexical simplification based on T5 as well as controllable mechanism and masked language model candidates as additional input-level context. The models are controllable, meaning that the outputs can be altered based on the token values embedded into each input sentence to match our desire, which in our case, we generate the outputs that maximize the evaluation metrics.

In Chapter 7 and 8, we presented several state-of-the-art sentence simplification systems for English and Spanish, using Transformer-based models (T5, mT5, mBart) coupled with the simplification control mechanism. The approach is inspired by the knowledge-transfer idea, where the model has been trained on multiple tasks and then fine-tuned with specific data to perform a certain task. Therefore, we fine-tuned our systems with Transformer-based pre-trained models along with control tokens embedded into each input. The control tokens are intended to have control over different aspects of the outputs, such as sentence length, amount of paraphrasing, lexical complexity, and syntactic complexity. The extensive evaluation conducted based on automatic, crowd-sourced, and expert evaluations demonstrated that the proposed systems achieved state-of-the-art performance in sentence simplification for both English and Spanish. The Transformer-based systems, specifically those utilizing a four-token control mechanism, were able to generate sentences that were simpler than the original ones, as evaluated based on easy-to-read guidelines. These transformer-based systems performed a wide range of simplification operations, resulting in simpler output according to the guidelines. In English sentence simplification, the use of T5 led to a higher number of simplification operations and correct transformations compared to the use of mBART. In Spanish sentence simplification, the results of the expert evaluation indicated that the size and the quality of the training data have an influence on the correctness of some transformations.

Our research started with the following questions, and now it is time

to revisit them:

- **RQ1** Is it possible to employ a deep learning method with word embeddings and engineered features to accurately identify lexical sources of complexity in sentences?

  In Chapter 4, we addressed the research question **RQ1** by presenting a complex word identification model that utilizes a CNN along with word embeddings and engineered features. Our approach was found to be highly effective, achieving results that are comparable to state-of-the-art models in English and even outperforming state-of-the-art models in Spanish and German, despite using a significantly smaller number of features. This aspect of our model makes it highly adaptable to other languages, as it can be easily modified to accommodate the specific features of different languages.

- **RQ2** Can we build an adaptive lexical simplification?

  The research question **RQ2** is addressed in Chapter 5 and Chapter 6 in which we proposed controllable lexical simplification models, both monolingual and multilingual. These models have the ability to adjust the simplifications according to the control token values embedded in each input, resulting in outputs that are tailored to suit various target audiences. However, in our case, we selected the control token values to generate the outputs that maximize the evaluation metrics.

- **RQ3** Can we build an adaptive sentence simplification?

  Chapter 7 and Chapter 8 address the research question **RQ3** by presenting various controllable sentence simplification models for English and Spanish. These models generate outputs based on the control tokens embedded in each input sentence, which can be customized to suit different target audiences. However, since we did not have the chance to evaluate the outputs with the target users, we mainly selected the control tokens values that maximize the

evaluation metric. Moreover, in Section 7.5, we did some analysis showing the effects of control tokens confirming that they do have control over the simplification outputs.

- **RQ4** Can transfer-learning methods be used to improve the performance of text simplification?

  To address the research question **RQ4**, we have clear evidence demonstrating that transfer learning significantly enhances the performance of sentence simplification. In Chapter 7, Section 7.3.6, we compared our model, which utilizes four tokens, to ACCESS (Martin et al., 2019), a sequence-to-sequence model (Transformer) trained from scratch. Our model, on the other hand, is fine-tuned using the T5 pre-trained model, and both models share a similar underlying architecture and are trained on the same data and control tokens. Our model performed significantly better than ACCESS in the ASSET dataset (+4.78) and TurkCorpus dataset (+1.93) on the SARI metric.

Regarding the limitations, we have highlighted the needs and challenges of target audiences, and our lexical and sentence simplification systems are capable of adjusting outputs to better serve these groups. However, we were not able to conduct an evaluation with these specific target users for two primary reasons. Firstly, to evaluate with a specific user group, we would require in-depth knowledge of their characteristics and requirements to generate outputs tailored to their needs, which would necessitate the involvement of experts or additional studies in the field. Secondly, we do not have access to these end users to carry out the evaluation. It is worth noting that the human evaluations conducted in Chapter 7 and Chapter 8 were conducted on readers without any reading impairments. Therefore, future studies should prioritize the evaluation of the proposed methods with target audiences who have specific needs and challenges in order to better understand the effectiveness of our models in serving these groups.

In conclusion, based on all the studies we have presented, we can say that we have achieved the objectives we set at the beginning, and the rest

will be left for future research.

Future Work

Despite the advancements in natural language processing, and machine learning, the current results in text simplification have yet to reach a satisfactory level that can be integrated into practical applications. The complexity of the task, the need for accurate preservation of information, and the lack of reliable evaluation metrics and datasets contribute to the ongoing challenges in text simplification research. Therefore, there is a potential for future research as the following:

Automatic Evaluation Metrics: One crucial aspect of text simplification is the development of automatic evaluation metrics that can effectively measure the quality and readability of simplified texts. Current evaluation metrics, such as BLEU and SARI, have limitations in capturing the semantic and structural changes introduced by simplification. Future work can focus on developing more sophisticated evaluation metrics that take into account the semantic and structural changes as well as specific linguistic characteristics of simplified texts.

Datasets: The availability of high-quality datasets is essential for training and evaluating text simplification models. Existing datasets based on Wikipedia data have been widely used but lack in quality and may not cover the full range of linguistic variations and complexities present in real-world texts. Newsela dataset offers higher quality with different levels of simplifications but is not open, which limits the possibility for research. Future work can involve the creation of new datasets that encompass a wider range of text genres, domains, and languages. These datasets should also include annotations for different levels of simplification, allowing for more fine-grained analysis and evaluation.

Accuracy: Improving the accuracy of text simplification models is another important area for future work. Current models often struggle with preserving the meaning and coherence of the original text while simplifying it. Future research can focus on developing more advanced

models that can better capture the semantic and syntactic structures and changes of the input text and generate simplified versions that are both accurate and readable. This can involve exploring new neural network architectures, incorporating external knowledge sources, and leveraging techniques such as transfer learning, multi-task learning, large language models, and instruction-based learning.

Domain-specific Simplification: Text simplification techniques can be further developed to cater to specific domains, such as medical texts, legal documents, or scientific literature. These domains often contain specialized terminology and complex concepts that can pose challenges for readers with limited domain knowledge. Future work can focus on developing domain-specific simplification models and datasets that are tailored to the unique characteristics and requirements of these domains.

Multilingual Simplification: While much of the existing research on text simplification has focused on English, there is a need for text simplification techniques in other languages as well. Future work can explore text simplification for different languages, taking into account the specific linguistic features and challenges of each language. This can involve leveraging parallel corpora, machine translation techniques (e.g., back translation), creating multilingual datasets, and investigating cross-lingual transfer learning techniques.

User-centered Evaluation: In addition to automatic evaluation metrics, future work can also focus on the user-centered evaluation of text simplification systems. This can involve conducting user studies to assess the readability, understandability, and usability of simplified texts by different user groups, such as individuals with low literacy, language learners, or people with cognitive impairments. This could involve incorporating user feedback and preferences into the text simplification process to ensure that the generated simplified text is tailored to the individual user's needs.

Bibliography

A. El-Zraigat, I. (2012). Assessing Special Needs of Students with Hearing Impairment in Jordan and Its Relation to Some Variables. *International Education Studies*, 6(2).

AbuRa'ed, A. and Saggion, H. (2018). LaSTUS/TALN at Complex Word Identification (CWI) 2018 Shared Task. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 159–165, New Orleans, Louisiana. Association for Computational Linguistics.

Adoyo, P. O. and Maina, E. N. (2019). *Practices and Challenges in Deaf Education in Kenya*, pages 73–86. Oxford University Press.

Adrian, J. E., Clemente, R. A., Villanueva, L., and Rieffe, C. (2005). Parent–child picture-book reading, mothers' mental state language and children's theory of mind. *Journal of Child Language*, 32(3):673–686.

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2019). FLAIR: An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In Teredesai, A., Kumar, V., Li, Y., Rosales, R., Terzi, E., and Karypis, G.,

editors, *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pages 2623–2631. ACM.

Al-Dawsari, H. M. and Hendley, R. (2022). Understanding the Needs of Arab Learners with Dyslexia for Adaptive Systems. In *35th International BCS Human-Computer Interaction Conference*.

Al-Shidhani, T. A. and Arora, V. (2012). Understanding Dyslexia in Children Through Human Development Theories. *Sultan Qaboos University Medical Journal*, 12(3):286–294.

Al-Thanyyan, S. S. and Azmi, A. M. (2021). Automated Text Simplification: A Survey. *ACM Computing Surveys*, 54(2):1–36.

Aleksandrova, D. and Brochu Dufour, O. (2022). RCML at TSAR-2022 shared task: Lexical simplification with modular substitution candidate ranking. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 259–263, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Alfter, D. and Pilán, I. (2018). SB@GU at the Complex Word Identification 2018 Shared Task. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 315–321, New Orleans, Louisiana. Association for Computational Linguistics.

Alsraisri, N., Albakheet, H., Alsajjan, N., and Aldaajani, N. (2020). Blended Learning Approach for Deaf or Hard of Hearing Students: Investigating university teachers' views. *Revista Amazonia Investiga*, 9(32):36–44.

Alva-Manchego, F., Bingel, J., Paetzold, G., Scarton, C., and Specia, L. (2017). Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint*

Conference on Natural Language Processing (Volume 1: Long Papers), pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Alva-Manchego, F., Martin, L., Bordes, A., Scarton, C., Sagot, B., and Specia, L. (2020a). ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Alva-Manchego, F., Martin, L., Scarton, C., and Specia, L. (2019). EASSE: Easier Automatic Sentence Simplification Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

Alva-Manchego, F., Scarton, C., and Specia, L. (2020b). Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187.

Anderson, H. L., DeAndrea-Lazarus, I., and Featherstone, Z. (2021). Leveraging the Perspectives of Deaf Trainees to Better Care for Vulnerable Communities. *Academic Medicine*, 96(6):783–784.

Ardila, A. (2010). Aphasia revisited: A reply to Buckingham, Kertesz, and Marshall. *Aphasiology*, 24(3):413–422.

Aroyehun, S. T., Angel, J., Pérez Alvarez, D. A., and Gelbukh, A. (2018). Complex Word Identification: Convolutional Neural Network vs. Feature Engineering. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 322–327, New Orleans, Louisiana. Association for Computational Linguistics.

Aumiller, D. and Gertz, M. (2022). UniHD at TSAR-2022 shared task: Is compute all we need for lexical simplification? In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 251–258, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Bahdanau, D., Cho, K., and Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Barbu, E., Martín-Valdivia, M. T., Martínez-Cámara, E., and Ureña-López, L. A. (2015). Language technologies applied to document simplification for helping autistic people. *Expert Systems with Applications*, 42(12):5076–5086.

Bernhard, D., Viron, L. D., Moriceau, V., and Tannier, X. (2012). Question Generation for French: Collating Parsers and Paraphrasing Questions. *Dialogue & Discourse*, 3(2):43–74.

Bilbili, S. (2013). Treatment of Children with Autism Spectrum Disorder in Vlora. *Mediterranean Journal of Social Sciences*.

Biran, O., Brody, S., and Elhadad, N. (2011). Putting it simply: A context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, volume 2 of *ACL*, pages 496–501, Portland, Oregon, USA. Association for Computational Linguistics.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2016). Enriching word vectors with subword information. arXiv 2016. *arXiv preprint arXiv:1607.04606*.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. In *Transactions of the Association for Computational Linguistics*, volume 5, pages 135–146, Cambridge, MA. MIT Press.

Borst, A., Gaudinat, A., Boyer, C., and Grabar, N. (2008). Lexically based distinction of readability levels of health documents. In *MIE 2008*.

Bott, S., Saggion, H., and Mille, S. (2012). Text simplification tools for Spanish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, LREC, pages 1665–1671, Istanbul, Turkey. European Language Resources Association (ELRA).

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M.-F., and Lin, H.-T., editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, Virtual*.

Butnaru, A. M. and Ionescu, R. T. (2018). UnibucKernel : A kernel-based learning method for complex word identification. *Proceedings of the Thirteenth Workshop on Innovative Use ofNLP for Building Educational Applications*, pages 175–183.

Cardellino, C. (2016). Spanish Billion Words Corpus and Embeddings.

Carroll, J., Minnen, G., Canning, Y., Devlin, S., and Tait, J. (1998). Practical simplification of English newspaper text to assist aphasic readers.

In *Proceedings of the AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10. Association for the Advancement of Artificial Intelligence.

Carroll, J., Minnen, G., Pearce, D., Canning, Y., Devlin, S., and Tait, J. (1999). Simplifying text for language-impaired readers. In *In Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 269–270, Bergen, Norway. Association for Computational Linguistics.

Cartwright, K. B. (2007). The Contribution of Graphophonological-Semantic Flexibility to Reading Comprehension in College Students: Implications for a Less Simple View of Reading. *Journal of Literacy Research*, 39(2):173–193.

Caute, A., Cruice, M., Friede, A., Galliers, J., Dickinson, T., Green, R., and Woolf, C. (2015). Rekindling the love of books – a pilot project exploring whether e-readers help people to read again after a stroke. *Aphasiology*, pages 1–30.

Chandrasekar, R., Doran, C., and Srinivas, B. (1996). Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*, volume 2, page 1041, Copenhagen, Denmark. Association for Computational Linguistics.

Chandrasekar, R. and Srinivas, B. (1997). Automatic induction of rules for text simplification. *Knowledge-Based Systems*, 10(3):183–190.

Chen, P., Rochford, J., Kennedy, D. N., Djamasbi, S., Fay, P., and Scott, W. (2017). Automatic Text Simplification for People with Intellectual Disabilities. In *Artificial Intelligence Science and Technology*, pages 725–731, Shanghai,China. WORLD SCIENTIFIC.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Chersoni, E. and Hsu, Y.-Y. (2022). PolyU-CBS at TSAR-2022 shared task: A simple, rank-based method for complex word substitution in two steps. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 225–230, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Cho, K., Van Merrienboer, B., Bahdanau, D., and Bengio, Y. (2014a). On the Properties of Neural Machine Translation: Encoder–Decoder Approaches. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Cho, K., Van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014b). 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, Doha, Qatar. Association for Computational Linguistics.

Cistola, G., Farrús, M., and Meulen, I. (2021). Aphasia and acquired reading impairments: What are the high-tech alternatives to compensate for reading deficits? *International Journal of Language & Communication Disorders*, 56(1):161–173.

Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., and Stoyanov, V. (2020). Unsupervised Cross-lingual Representation Learning at Scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Cornoldi, C., Rivella, C., Montesano, L., and Toffalini, E. (2022). Difficulties of Young Adults With Dyslexia in Reading and Writing Numbers. *Journal of Learning Disabilities*, 55(4):338–348.

Coster, W. and Kauchak, D. (2011a). Learning to simplify sentences using Wikipedia. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1–9, Portland, Oregon. Association for Computational Linguistics.

Coster, W. and Kauchak, D. (2011b). Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Portland, Oregon, USA: Association for Computational Linguistics*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Cox-Magno, N., Ross, P., Dimino, K., and Wilson, A. (2018). Metacognitive Reading Strategy and Emerging Reading Comprehension in Students With Intellectual Disabilities. *Journal of Educational Research and Practice*, 8(1).

Crossley, S. A., Allen, D., and McNamara, D. S. (2011). Text simplification and comprehensible input: A case for an intuitive approach. *Language Teaching Research*, 16(1):89–108.

Cuetos, F., Glez-Nosti, M., Barbón, A., and Brysbaert, M. (2012). SUBTLEX-ESP: Spanish word frequencies based on film subtitles. *Psicológica*, 33(2):133–143.

Cumbicus-Pineda, O. M., Gonzalez-Dios, I., and Soroa, A. (2021). A Syntax-Aware Edit-based System for Text Simplification. In *Proceedings of the Conference Recent Advances in Natural Language Processing - Deep Learning for Natural Language Processing Methods and Applications*, pages 324–334, Varna, Bulgaria. INCOMA Ltd.

Dalilan, Sartika, E., and Lestari, D. I. (2021). The Practices and Obstacles of English Language Teaching in Intellectual Disability Classroom: A Case Study at Special School (SLB) in Palembang. *PANYONARA: Journal of English Education*, 3(1):1–18.

De Belder, J., Deschacht, K., and Moens, M.-F. (2010). Lexical simplification. In *Proceedings of Itec2010 : 1st International Conference on Interdisciplinary Research on Technology, Education and Communication*, Kortrijk, Belgium.

De Belder, J. and Moens, M.-F. (2012). A Dataset for the Evaluation of Lexical Simplification. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., and Gelbukh, A., editors, *Computational Linguistics and Intelligent Text Processing*, volume 7182, pages 426–437. Springer Berlin Heidelberg, Berlin, Heidelberg.

De Hertog, D. and Tack, A. (2018). Deep Learning Architecture for Complex Word Identification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 328–334, New Orleans, Louisiana. Association for Computational Linguistics.

Dębska, A., Zawadzka, A., Uniejewska, S., and Słoma, D. (2020). Psychopathology of mental and behavioral disorders in people with intellectual disability. *Journal of Education, Health and Sport*, 10(9):422–430.

Deepti S (2016). Casual Reading Habits and Interpersonal Reactivity: A Correlational Study. *International Journal of Indian Psychology*, 3(2).

Degraeuwe, J. and Saggion, H. (2022). Lexical simplification in foreign language learning: Creating pedagogically suitable simplified example sentences. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 98–110, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Deng, Y. and Wang, Q. (2019). Influence of Starting Age on Second Language Acquisition. In *Proceedings of the 2019 International Confer-*

ence on Contemporary Education and Society Development (ICCESD 2019), Jinan, China. Atlantis Press.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Devlin, S. and Tait, J. (1998). The Use of a Psycholinguistic Database in the Simplification of Text for Aphasic Readers. In *Linguistic Databases*, pages 161–173. CSLI.

Dong, Y., Li, Z., Rezagholizadeh, M., and Cheung, J. C. K. (2019). EditNTS: An Neural Programmer-Interpreter Model for Sentence Simplification through Explicit Editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Dorogush, A. V., Ershov, V., and Gulin, A. (2018). CatBoost: Gradient boosting with categorical features support.

Drndarević, B. and Saggion, H. (2012). Towards automatic lexical simplification in Spanish: An empirical study. In *Proceedings of the First Workshop on Predicting and Improving Text Readability for Target Reader Populations*, PITR, pages 8–16, Montréal, Canada. Association for Computational Linguistics.

Edunov, S., Ott, M., Auli, M., and Grangier, D. (2018). Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

Elias, L., Venu, M., Anand, P. K. V., and Rahul H (2023). Management of "Wernick's Aphasia" through Ayurveda- A Case Study. *International Research Journal of Ayurveda & Yoga*, 06(04):49–55.

Evans, R., Orasan, C., and Dornescu, I. (2014). An evaluation of syntactic simplification rules for people with autism. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, PITR, pages 131–140, Gothenburg, Sweden. Association for Computational Linguistics.

Evans, R. J. (2011). Comparing methods for the syntactic simplification of sentences in information extraction. *Literary and Linguistic Computing*, 26(4):371–388.

Fabbretti, D., Volterra, V., and Pontecorvo, C. (1998). Written language abilities in deaf Italians. *The Journal of Deaf Studies and Deaf Education*, 3(3):231–244.

Fan, A., Grangier, D., and Auli, M. (2018). Controllable Abstractive Summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Fazeli, S., Naghibolhosseini, M., and Bahrami, F. (2008). An Adaptive Neuro-Fuzzy Inference System for Diagnosis of Aphasia. In *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*, pages 535–538, Shanghai, China. IEEE.

Feng, Y., Zhang, S., Zhang, A., Wang, D., and Abel, A. (2017). Memory-augmented Neural Machine Translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1390–1399, Copenhagen, Denmark. Association for Computational Linguistics.

Ferrés, D., Marimon, M., Saggion, H., and AbuRa'ed, A. (22). YATS: Yet Another Text Simplifier. In *Proceedings of the 21st International Con-*

ference on Applications of Natural Language to Information Systems, pages 335–342.

Ferrés, D., Saggion, H., and Gómez Guinovart, X. (2017). An Adaptable Lexical Simplification Architecture for Major Ibero-Romance Languages. In *Proceedings of the First Workshop on Building Linguistically Generalizable NLP Systems*, pages 40–47, Copenhagen, Denmark. Association for Computational Linguistics.

Ficler, J. and Goldberg, Y. (2017). Controlling Linguistic Style Aspects in Neural Language Generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, Copenhagen, Denmark. Association for Computational Linguistics.

Fleiss, J. L. (1971). Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Freyhoff, G., Hess, G., Kerr, L., Tronbacke, B., and Van Der Veken, K. (1998a). Make it simple.

Freyhoff, G., Hess, G., Kerr, L., Tronbacke, B., and Van Der Veken, K. (1998b). *Make It Simple, European Guidelines for the Production of Easy-toRead Information for People with Learning Disability*. ILSMH European Association, Brussels.

Gala, N., François, T., Bernhard, D., and Fairon, C. (2014). A model to predict lexical complexity and to grade words (Un modèle pour prédire la complexité lexicale et graduer les mots) [in French]. In *Proceedings of TALN 2014*, pages 91–102, Marseille, France.

Gala, N. and Ziegler, J. (2016). Reducing lexical complexity as a tool to increase text accessibility for children with dyslexia. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 59–66, Osaka, Japan. The COLING 2016 Organizing Committee.

Ganitkevitch, J., Van Durme, B., and Callison-Burch, C. (2013). PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Glavaš, G. and Štajner, S. (2015). Simplifying Lexical Simplification: Do We Need Simplified Corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China. Association for Computational Linguistics.

Gonzalez, Y. and Camacho-Vega, D.-O. (2021). THE IMPORTANCE OF WELL-BEING AND SATISFACTION WITH LIFE IN READING-COMPREHENSION AND MATHEMATICAL ABILITIES IN DEAF AND HEARING INDIVIDUALS. *ANTHROPOLOGICAL RESEARCHES AND STUDIES*, 1(11):117–128.

Gooding, S. and Kochmar, E. (2018). CAMB at CWI Shared Task 2018: Complex Word Identification with Ensemble-Based Voting. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 184–194, New Orleans, Louisiana. Association for Computational Linguistics.

Gordon, J. K. (2008). Measuring the lexical semantics of picture description in aphasia. *Aphasiology*, 22(7-8):839–852.

Gorman, B. (2009). Cross-Linguistic Universals in Reading Acquisition with Applications to English-Language Learners with Reading Disabilities. *Seminars in Speech and Language*, 30(04):246–260.

Grabar, N., Claveau, V., and Dalloux, C. (2018). CAS: French corpus with clinical cases. In *LOUHI 2018*, pages 1–12, Bruxelles, Belgique.

Gutiérrez-Fandiño, A., Armengol-Estapé, J., Pàmies, M., Llop-Palao, J., Silveira-Ocampo, J., Carrino, C. P., Gonzalez-Agirre, A., Armentano-Oller, C., Rodriguez-Penagos, C., and Villegas, M. (2022). MarIA: Spanish Language Models. *Procesamiento del Lenguaje Natural*, abs/2107.07253:39–60.

Haffari, G., Roy, M., and Sarkar, A. (2009). Active learning for statistical phrase-based machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on - NAACL '09*, page 415, Boulder, Colorado. Association for Computational Linguistics.

Halepoto, D. M. and Al-Ayadhi, L. Y. (2014). In Search of Autism Biomarkers: Possible Autism Bio-Markers Discovery at Autism Research and Treatment Center, King Saud University, KSA. *International Journal of Bioscience, Biochemistry and Bioinformatics*, 4(3):175–179.

Harris, J. C. (2005). *Understanding and Evaluating Emotional and Behavioral Impairment*, pages 140–187. Oxford University Press.

Hartmann, N. and Santos, L. B. (2018). NILC at CWI 2018: Exploring Feature Engineering and Feature Learning. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 335–340, New Orleans, Louisiana. Association for Computational Linguistics.

Herschensohn, J. (2007). *Language Development and Age*. Cambridge University Press, 1 edition.

Hmeljak Sangawa, K. (2016). An Analysis of Simplification Strategies in a Reading Textbook of Japanese as a Foreign Language. *Acta Linguistica Asiatica*, 6(1):9–33.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Honnibal, M. and Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://spacy.io/.

Horn, C., Manduca, C., and Kauchak, D. (2014). Learning a Lexical Simplifier Using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland. Association for Computational Linguistics.

Howard, J. and Ruder, S. (2018). Universal Language Model Fine-tuning for Text Classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Howell, J. J. and Luckner, J. L. (2003). Helping One Deaf Student Develop Content Literacy Skills: An Action Research Report. *Communication Disorders Quarterly*, 25(1):23–27.

Hwang, W., Hajishirzi, H., Ostendorf, M., and Wu, W. (2015). Aligning Sentences from Standard Wikipedia to Simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 211–217, Denver, Colorado. Association for Computational Linguistics.

Ingram, R. U., Halai, A. D., Pobric, G., Sajjadi, S., Patterson, K., and Lambon Ralph, M. A. (2020). Graded, multidimensional intra- and intergroup variations in primary progressive aphasia and post-stroke aphasia. *Brain*, 143(10):3121–3135.

Inui, K., Fujita, A., Takahashi, T., Iida, R., and Iwakura, T. (2003). Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing* -, volume 16, pages 9–16, Sapporo, Japan. Association for Computational Linguistics.

Irdamurni, I., Kasiyati, K., Zulmiyetri, Z., and Taufan, J. (2018). The Influence of Mentoring on Teachers Performance in Reading Instruction for Dyslexia Children. In *Proceedings of the International Conference of Mental Health, Neuroscience, and Cyber-psychology - Icometh-NCP 2018*, pages 58–61, Padang. Fakultas Ilmu Pendidikan.

Ivo Paclt and Anna Strunecka (2010). Autism Spectrum Disorders: Clinical Aspects. In Strunecka, A., L. Blaylock, R., A. Hyman, M., and Paclt, I., editors, *Cellular and Molecular Biology of Autism Spectrum Disorders*, pages 1–16. BENTHAM SCIENCE PUBLISHERS.

Javourey-Drevet, L., Dufau, S., François, T., Gala, N., Ginestié, J., and Ziegler, J. C. (2022). Simplification of literary and scientific texts to improve reading fluency and comprehension in beginning readers of French. *Applied Psycholinguistics*, 43(2):485–512.

Johnson, C. P., Myers, S. M., and and the Council on Children With Disabilities (2007). Identification and Evaluation of Children With Autism Spectrum Disorders. *Pediatrics*, 120(5):1183–1215.

Jonnalagadda, S. and Gonzalez, G. (2010). BioSimplify: An open source sentence simplification engine to improve recall in automatic biomedical information extraction. *AMIA 2010 Symposium Proceedings*, 2010:351–5.

Kajiwara, T. and Komachi, M. (2018). Complex Word Identification Based on Frequency in a Learner Corpus. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 195–199, New Orleans, Louisiana. Association for Computational Linguistics.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Kariuk, O. and Karamshuk, D. (2020). CUT: Controllable Unsupervised Text Simplification. volume abs/2012.01936 of *ArXiv Preprint*.

Kauchak, D. (2013). Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL, pages 1537–1546, Sofia, Bulgaria. Association for Computational Linguistics.

Kikuchi, Y., Neubig, G., Sasano, R., Takamura, H., and Okumura, M. (2016). Controlling Output Length in Neural Encoder-Decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, Austin, Texas. Association for Computational Linguistics.

Kim, Y.-S. G. and Wagner, R. K. (2015). Text (Oral) Reading Fluency as a Construct in Reading Development: An Investigation of Its Mediating Role for Children From Grades 1 to 4. *Scientific Studies of Reading*, 19(3):224–242.

Kincaid, J. P., Fishburne, R. P., Rogers, R. L., and Chissom, B. S. (1975). Derivation of New Readability Formulas (Automated Readability Index, Fog Count and Flesch Reading Ease Formula) for Navy Enlisted Personnel. Technical report, National Technical Information Service, Springfield, Virginia 22151.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Krashen, S. (1989). We acquire vocabulary and spelling by reading: Additional evidence for the input hypothesis. *The modern language journal*, 73(4):440–464.

Kriz, R., Sedoc, J., Apidianaki, M., Zheng, C., Kumar, G., Miltsakaki, E., and Callison-Burch, C. (2019). Complexity-Weighted Loss and

Diverse Reranking for Sentence Simplification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3137–3147, Minneapolis, Minnesota. Association for Computational Linguistics.

Kumar, D., Mou, L., Golab, L., and Vechtomova, O. (2020). Iterative Edit-Based Unsupervised Sentence Simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

Kumar, M., Agrawal, M. A., and MCh, Mc. (2017). Pure motor aphasia: An uncommon presentation of left middle cerebral artery territory infarct. *Narayana Medical Journal*, 6(2):30.

Kurinna, ., Iliichuk, L., Mamicheva, O., Nesterenko, T., Romanova, I., and Pinchuk, Y. (2022). Formation of Rhetorical Competence in University Applicants as a Necessary Factor for Successful Professional Activity. *Revista Romaneasca pentru Educatie Multidimensionala*, 14(4):230–242.

Kushalnagar, P., Smith, S., Hopper, M., Ryan, C., Rinkevich, M., and Kushalnagar, R. S. (2016). Making Cancer Health Text on the Internet Easier to Read for Deaf People Who Use American Sign Language. *Journal of Cancer Education*, 33(1):134–140.

Kusmiatun, A. and Liliani, E. (2020). Indonesia–Thailand Culture Similarities and Their Contributions in BIPA Learning. In *Proceedings of the International Conference on Educational Research and Innovation (ICERI 2019)*, Yogyakarta, Indonesia. Atlantis Press.

Lal, P. and Rüger, S. (2002). Extract-based Summarization with Simplification. In *Proceedings of the DUC 2002 Workshop on Text Summarization*.

Laurent, D., Nègre, S., and Séguéla, P. (2009). L'analyseur syntaxique Cordial dans Passage. In *Traitement Automatique Des Langues Naturelles (TALN)*.

LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.

Lederberg, A. R., Schick, B., and Spencer, P. E. (2013). Language and literacy development of deaf and hard-of-hearing children: Successes and challenges. *Developmental Psychology*, 49(1):15–30.

Levenshtein, V. (1965). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet physics. Doklady*, 10:707–710.

Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., and Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Li, S., Liu, Z.-Q., and Chan, A. B. (2015). Heterogeneous Multi-task Learning for Human Pose Estimation with Deep Convolutional Neural Network. *International Journal of Computer Vision*, 113(1):19–36.

Li, X., Wiechmann, D., Qiao, Y., and Kerz, E. (2022). MANTIS at TSAR-2022 shared task: Improved unsupervised lexical simplification with pretrained encoders. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 243–250, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Licardo, M., Volčanjk, N., and Haramija, D. (2021). Differences in Communication Skills among Elementary Students with Mild Intellectual Disabilities after Using Easy-to-Read Texts. *The New Educational Review*, 64(2):236–246.

Lin, T., Wang, Y., Liu, X., and Qiu, X. (2022). A survey of transformers. *AI Open*, 3:111–132.

Liu, X., Duh, K., Liu, L., and Gao, J. (2020a). Very Deep Transformers for Neural Machine Translation.

Liu, Y., Gu, J., Goyal, N., Li, X., Edunov, S., Ghazvininejad, M., Lewis, M., and Zettlemoyer, L. (2020b). Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv preprint*, abs/1907.11692.

Loshchilov, I. and Hutter, F. (2019). Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Lu, J., Zhang, D., Zhang, J., and Zhang, P. (2022). Flat Multi-modal Interaction Transformer for Named Entity Recognition. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2055–2064, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Maddela, M., Alva-Manchego, F., and Xu, W. (2021). Controllable Text Simplification with Explicit Paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553, Online. Association for Computational Linguistics.

Malebese, M. L., Tlali, M. F., and Mahlomaholo, S. (2019). A socially inclusive teaching strategy for fourth grade English (second) language learners in a South African school. *South African Journal of Childhood Education*, 9(1).

Marcu, D. and Wong, W. (2002). A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - EMNLP '02*, volume 10, pages 133–139, Not Known. Association for Computational Linguistics.

Martin, L., de la Clergerie, É., Sagot, B., and Bordes, A. (2019). Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Martin, L., Fan, A., de la Clergerie, É., Bordes, A., and Sagot, B. (2020a). Multilingual Unsupervised Sentence Simplification. *ArXiv preprint*, abs/2005.00352.

Martin, L., Fan, A., de la Clergerie, É., Bordes, A., and Sagot, B. (2022). MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.

Martin, L., Muller, B., Ortiz Suárez, P. J., Dupont, Y., Romary, L., de la Clergerie, É., Seddah, D., and Sagot, B. (2020b). CamemBERT: A tasty french language model. *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Matausch, K. and Peböck, B. (2010). Easyweb - A study how people with specific learning difficulties can be supported on using the internet. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6179 LNCS(PART 1):641–648.

Mencap (2002). *Am I Making Myself Clear? Mencap's Guidelines for Accessible Writing*. London.

Mikhailova, N. F., Fattakhova, M. E., Mironova, M. A., and Vyacheslavova, E. V. (2019). Psychological Adaptation Of Deaf And Hard-Of-Hearing Students. In *Psychology of Subculture: Phenomenology and Contemporary Tendencies of Development*, pages 398–405.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013). Efficient estimation of word representations in vector space. In Bengio, Y. and LeCun, Y., editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*.

Miller, G. A. (1994). WordNet: A Lexical Database for English. In *Human Language Technology: Proceedings of a Workshop Held at Plainsboro, New Jersey, March 8-11, 1994*.

Millichap, J. G. and Millichap, J. J. (2014). AAP Genetics Diagnostic Approach to Intellectual Disability or Global Developmental Delay. *Pediatric Neurology Briefs*, 28(10):79.

Mohammed, I. J. (2020). Learner Differences in Second Language Acquisition. *Journal of Tikrit University for Humanities*, 27(8):43–55.

Mohammed, S. A., Rajashekar, S., Giri Ravindran, S., Kakarla, M., Ausaja Gambo, M., Yousri Salama, M., Haidar Ismail, N., Tavalla, P., Uppal, P., and Hamid, P. (2022). Does Vaccination Increase the Risk of Autism Spectrum Disorder? *Cureus*.

Narayan, S. and Gardent, C. (2016). Unsupervised Sentence Simplification Using Deep Semantics. In *Proceedings of the 9th International Natural Language Generation Conference*, pages 111–120, Edinburgh, UK. Association for Computational Linguistics.

Nikita, N. and Rajpoot, P. (2022). teamPN at TSAR-2022 shared task: Lexical simplification using multi-level and modular approach. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 239–242, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Nishihara, D., Kajiwara, T., and Arase, Y. (2019). Controllable Text Simplification with Lexical Constraint Loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

Nisioi, S., Štajner, S., Ponzetto, S. P., and Dinu, L. P. (2017). Exploring Neural Text Simplification Models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Norbury, C. and Nation, K. (2011). Understanding Variability in Reading Comprehension in Adolescents With Autism Spectrum Disorders: Interactions With Language Status and Decoding Skill. *Scientific Studies of Reading*, 15(3):191–210.

North, K., Dmonte, A., Ranasinghe, T., and Zampieri, M. (2022). GMU-WLV at TSAR-2022 shared task: Evaluating lexical simplification models. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 264–270, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

North, K., Ranasinghe, T., Shardlow, M., and Zampieri, M. (2023). Deep Learning Approaches to Lexical Simplification: A Survey.

Och, F. J. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, 29(1):19–51.

Och, F. J. and Ney, H. (2004). The Alignment Template Approach to Statistical Machine Translation. *Computational Linguistics*, 30(4):417–449.

Olajide, S. B. (2010). Linking Reading and Writing in An English-As-A-Second-Language (ESL) Classroom for National Reorientation and Reconsruction. *International Education Studies*, 3(3).

Omelianchuk, K., Raheja, V., and Skurzhanskyi, O. (2021). Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

Orăsan, C., Evans, R., and Dornescu, I. (2013). Text simplification for people with autistic spectrum disorders. *Towards multilingual europe 2020: A romanian perspective*.

Paetzold and Specia (2016a). BenchLS: A Reliable Dataset for Lexical Simplification.

Paetzold and Specia (2016b). NNSeval: Evaluating Lexical Simplification for Non-Natives.

Paetzold, G. and Specia, L. (2015). LEXenstein: A Framework for Lexical Simplification. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 85–90, Beijing, China. Association for Computational Linguistics and The Asian Federation of Natural Language Processing.

Paetzold, G. and Specia, L. (2016c). Benchmarking lexical simplification systems. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3074–3080, Portorož, Slovenia. European Language Resources Association (ELRA).

Paetzold, G. and Specia, L. (2016d). SemEval 2016 Task 11: Complex Word Identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Paetzold, G. H. and Specia, L. (2016e). Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI*

Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA, volume 30, pages 3761–3767. AAAI Press.

Paetzold, G. H. and Specia, L. (2017). A Survey on Lexical Simplification. *Journal of Artificial Intelligence Research*, 60(1):549–593.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Pavlick, E. and Callison-Burch, C. (2016). Simple PPDB: A Paraphrase Database for Simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany. Association for Computational Linguistics.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Phytanza, D. T. P., BurhaeiN, E., Sukoco, and Ghautama, W. S. (2018). LIFE SKILL DIMENSION BASED ON UNIFIED SPORTS SOCCER PROGRAM IN PHYSICAL EDUCATION OF INTELLECTUAL DISABILITY. *Yaşam Becerileri Psikoloji Dergisi*, 2(4):199–205.

PlainLanguage (2011). Federal plain language guidelines.

Popović, M. (2018). Complex Word Identification Using Character n-grams. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 341–348, New Orleans, Louisiana. Association for Computational Linguistics.

Qiang, J., Li, Y., Zhu, Y., Yuan, Y., and Wu, X. (2020). Lexical simplification with pretrained encoders. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pages 8649—8656. AAAI Press.

Qiang, J., Li, Y., Zhu, Y., Yuan, Y., and Wu, X. (2021). LSBert: Lexical Simplification Based on BERT. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3064–3076.

Quinlan, P. (1992). *The Oxford Psycholinguistic Database*. Oxford University Press.

Rabia, M., Mubarak, N., Tallat, H., and Nasir, W. (2017). A Study on Study Habits and Academic Performance of Students. *International Journal of Asian Social Science*, 7(10):891–897.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. (2020). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Reimers, N. and Gurevych, I. (2020). Making Monolingual Sentence Embeddings Multilingual using Knowledge Distillation.

Rello, L., Baeza-Yates, R., Bott, S., and Saggion, H. (2013a). Simplify or help?: Text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility - W4A '13*, page 1, Rio de Janeiro, Brazil. ACM Press.

Rello, L., Baeza-Yates, R. A., Dempere-Marco, L., and Saggion, H. (2013b). Frequent Words Improve Readability and Short Words Improve Understandability for People with Dyslexia. In *Proceedings of the International Conference on Human-Computer Interaction (Part IV)*, INTERACT, pages 203–219, Cape Town, South Africa.

Rello, L., Bautista, S., Baeza-Yates, R., Gervás, P., Hervás, R., and Saggion, H. (2013c). One Half or 50%? An Eye-Tracking Study of Number Representation Readability. In Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Kotzé, P., Marsden, G., Lindgaard, G., Wesson, J., and Winckler, M., editors, *Human-Computer Interaction – INTERACT 2013*, volume 8120, pages 229–245. Springer Berlin Heidelberg, Berlin, Heidelberg.

Rets, I. and Jekaterina Rogaten (2020). To simplify or not? Facilitating English L2 users' comprehension and processing of open educational resources in English using text simplification. *Journal of Computer Assisted Learning*, 37(3):705–717.

Ristić, I., Popović, D., and Milovanović, B. (2021). Indicators of the Wider Social Context and Academic Performance of the Deaf and Hard of Hearing Students. *International Journal of Cognitive Research in Science, Engineering and Education (IJCRSEE)*, 9(2):265–274.

Rivero-Contreras, M., Engelhardt, P. E., and Saldaña, D. (2021). An experimental eye-tracking study of text adaptation for readers with dyslexia: Effects of visual support and word frequency. *Annals of Dyslexia*, 71(1):170–187.

Roitsch, J. and Watson, S. (2019). An Overview of Dyslexia: Definition, Characteristics, Assessment, Identification, and Intervention. *Science Journal of Education*, 7(4):81.

Roksandić, I., Pavković, I., and Kovačević, J. (2018). THE CHARACTERISTICS OF BEHAVIOR OF DEAF AND HARD-OF-HEARING LEARNERS IN DIFFERENT TYPES OF SCHOOL ENVIRONMENT. *Journal Human Research in Rehabilitation*, 8(1):27–34.

Romberg, F., Shaywitz, B. A., and Shaywitz, S. E. (2016). How Should Medical Schools Respond to Students with Dyslexia? *AMA journal of ethics*, 18(10):975–985.

Saggion, H. (2017). *Automatic Text Simplification*, volume 10 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers, University of Toronto.

Saggion, H., Gómez-Martínez, E., Etayo, E., Anula, A., and Bourg, L. (2011). Text Simplification in Simplext: Making Text More Accessible. *Revista de la Sociedad Española para el Procesamiento del Lenguaje Natural*, 47:3504–3510.

Saggion, H., Štajner, S., Bott, S., Mille, S., Rello, L., and Drndarevic, B. (2015). Making It Simplext: Implementation and Evaluation of a Text Simplification System for Spanish. *ACM Transactions on Accessible Computing*, 6(4):1–36.

Saggion, H., Štajner, S., Ferrés, D., Sheang, K. C., Shardlow, M., North, K., and Zampieri, M. (2022). Findings of the TSAR-2022 shared task on multilingual lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 271–283, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Salton, G. (1991). Developments in automatic text retrieval. *Science (New York, N.Y.)*, 253:974–979.

Scarton, C. and Specia, L. (2018). Learning Simplifications for Specific Target Audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Scherer, N., Smythe, T., Hussein, R., Wapling, L., Hameed, S., Eaton, J., Kabaja, N., Kakuma, R., and Polack, S. (2023). Communication, inclusion and psychological wellbeing among deaf and hard of hearing children: A qualitative study in the Gaza Strip. *PLOS Global Public Health*, 3(6):e0001635.

Seneviratne, S., Daskalaki, E., and Suominen, H. (2022). CILS at TSAR-2022 shared task: Investigating the applicability of lexical substitution methods for lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 207–212, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Shardlow, M. (2013). A comparison of techniques to automatically identify complex words. In *51st Annual Meeting of the Association for Computational Linguistics Proceedings of the Student Research Workshop*, pages 103–109, Sofia, Bulgaria. Association for Computational Linguistics.

Shardlow, M. (2014). Out in the open: Finding and categorising errors in the lexical simplification pipeline. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, LREC, pages 1583–1590, Reykjavik, Iceland. European Language Resources Association (ELRA).

Shardlow, M., Evans, R., Paetzold, G. H., and Zampieri, M. (2021). SemEval-2021 task 1: Lexical complexity prediction. In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 1–16, Online. Association for Computational Linguistics.

Shaywitz, S. E., Escobar, M. D., Shaywitz, B. A., Fletcher, J. M., and Makuch, R. (1992). Evidence That Dyslexia May Represent the Lower Tail of a Normal Distribution of Reading Ability. *New England Journal of Medicine*, 326(3):145–150.

Sheang, K. C. and Saggion, H. (2021). Controllable sentence simplification with a unified text-to-text transfer transformer. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Shurr, J. and Taber-Doughty, T. (2017). The Picture Plus Discussion Intervention: Text Access for High School Students with Moderate Intellectual Disability. *Focus on Autism and Other Developmental Disabilities*, 32(3):198–208.

Siddharthan, A. (2002). An architecture for a text simplification system. In *Language Engineering Conference, 2002. Proceedings*, pages 64–71, Hyderabad, India. IEEE Comput. Soc.

Siddharthan, A. (2006). Syntactic Simplification and Text Cohesion. *Research on Language and Computation*, 4(1):77–109.

Siddharthan, A. (2010). Complex Lexico-syntactic Reformulation of Sentences Using Typed Dependency Representations. In *Proceedings of the 6th International Natural Language Generation Conference*. Association for Computational Linguistics.

Siddharthan, A. (2011). Text Simplification using Typed Dependencies: A Comparision of the Robustness of Different Generation Strategies. In *Proceedings of the 13th European Workshop on Natural Language*

Generation, pages 2–11, Nancy, France. Association for Computational Linguistics.

Siddharthan, A. (2014). A survey of research on text simplification. *ITL - International Journal of Applied Linguistics*, 165(2):259–298.

Siddharthan, A., Nenkova, A., and McKeown, K. (2004). Syntactic simplification for improving content selection in multi-document summarization. In *Proceedings of the 20th International Conference on Computational Linguistics - COLING '04*, pages 896–902, Geneva, Switzerland. Association for Computational Linguistics.

Sivasubramanian, P. (2020). Emotional Intelligence in Individuals With Intellectual disability:. In Gopalan, R. T., editor, *Advances in Medical Diagnosis, Treatment, and Care*, pages 236–249. IGI Global.

Specia, L. (2010). Translating from Complex to Simplified Sentences. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, volume 7243 of *PROPOR*, pages 30–39, Porto Alegre, RS, Brazil. Springer Berlin Heidelberg.

Stajner, S. (2021). Automatic Text Simplification for Social Good: Progress and Challenges. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652, Online. Association for Computational Linguistics.

Štajner, S., Ferrés, D., Shardlow, M., North, K., Zampieri, M., and Saggion, H. (2022a). Lexical simplification benchmarks for English, Portuguese, and Spanish. *Frontiers in Artificial Intelligence*, 5:991242.

Štajner, S., Franco-Salvador, M., Ponzetto, S. P., Rosso, P., and Stuckenschmidt, H. (2017). Sentence Alignment Methods for Improving Text Simplification Systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 97–102, Vancouver, Canada. Association for Computational Linguistics.

Štajner, S., Franco-Salvador, M., Rosso, P., and Ponzetto, S. P. (2018). CATS: A Tool for Customized Alignment of Text Simplification Corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3895–3903, Miyazaki, Japan. European Language Resources Association (ELRA).

Štajner, S., Mitkov, R., and Saggion, H. (2014). One Step Closer to Automatic Evaluation of Text Simplification Systems. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR) at EACL*, pages 1–10, Gothenburg, Sweden. Association for Computational Linguistics.

Štajner, S. and Popovic, M. (2016). Can text simplification help machine translation? In *In Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, volume 4, pages 230–242.

Štajner, S., Sheang, K. C., and Saggion, H. (2022b). Sentence Simplification Capabilities of Transfer-Based Models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 12172–12180, British Columbia, Canada (Virtual).

Stanovich, K. E. (2009). Matthew Effects in Reading: Some Consequences of Individual Differences in the Acquisition of Literacy. *Journal of Education*, 189(1-2):23–55.

Sucksmith, E., Roth, I., and Hoekstra, R. A. (2011). Autistic Traits Below the Clinical Threshold: Re-examining the Broader Autism Phenotype in the 21st Century. *Neuropsychology Review*, 21(4):360–389.

Sulem, E., Abend, O., and Rappoport, A. (2018). BLEU is Not Suitable for the Evaluation of Text Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Sun, K. K. and Kemp, C. (2006). The Acquisition of Phonological Awareness and its Relationship to Reading in Individuals with Intellectual Disabilities. *Australasian Journal of Special Education*, 30(1):86–99.

Surya, S., Mishra, A., Laha, A., Jain, P., and Sankaranarayanan, K. (2019). Unsupervised Neural Text Simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Tanprasert, T. and Kauchak, D. (2021). Flesch-Kincaid is Not a Text Simplification Evaluation Metric. In *Proceedings of the 1st Workshop on Natural Language Generation, Evaluation, and Metrics (GEM 2021)*, pages 1–14, Online. Association for Computational Linguistics.

Ushio, A. and Camacho-Collados, J. (2021). T-NER: An All-Round Python Library for Transformer-based Named Entity Recognition. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 53–62, Online. Association for Computational Linguistics.

Vale, A. P., Fernandes, C., and Cardoso, S. (2022). Word reading skills in autism spectrum disorder: A systematic review. *Frontiers in Psychology*, 13.

Vasilakopoulou, S. and Tzvetkova-Arsova, M. (2021). AN APPROACH TO INTELLECTUAL DISABILITY. *Business Management, Economics and Social Sciences*, 1(1):86.

Vásquez-Rodríguez, L., Nguyen, N., Shardlow, M., and Ananiadou, S. (2022). UoM&MMU at TSAR-2022 shared task: Prompt learning for

lexical simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 218–224, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Vásquez-Rodríguez, L., Shardlow, M., Przybyła, P., and Ananiadou, S. (2021). Investigating Text Simplification Evaluation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 876–882, Online. Association for Computational Linguistics.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, \., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Vickrey, D. and Koller, D. (2008). Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, volume 344–352, pages 344–352, Columbus, Ohio. Association for Computational Linguistics.

Volkmar, F. R., Lord, C., Bailey, A., Schultz, R. T., and Klin, A. (2004). Autism and pervasive developmental disorders. *Journal of Child Psychology and Psychiatry*, 45(1):135–170.

Vu, T., Hu, B., Munkhdalai, T., and Yu, H. (2018). Sentence Simplification with Memory-Augmented Neural Networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 79–85, New Orleans, Louisiana. Association for Computational Linguistics.

Wani, N., Mathias, S., Gajjam, J. A., and Bhattacharyya, P. (2018). The whole is greater than the sum of its parts : Towards the effectiveness of voting ensemble classifiers for complex word identification. *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 200–205.

Watanabe, S. and Iwasaki, N. (2009). The acquisition of japanese modality during study abroad. *Japanese modality*, pages 231–258.

Whistely, P., Mathias, S., and Poornima, G. (2022). PresiUniv at TSAR-2022 shared task: Generation and ranking of simplification substitutes of complex words in multiple languages. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 213–217, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Wilkens, R., Alfter, D., Cardon, R., Gribomont, I., Bibal, A., Patrick, W., De marneffe, M.-C., and François, T. (2022). CENTAL at TSAR-2022 shared task: How does context impact BERT-Generated substitutions for lexical simplification? In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 231–238, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). HuggingFace's transformers: State-of-the-art natural language processing. *CoRR*, abs/1910.03771.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Woodsend, K. and Lapata, M. (2011). Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language*

Processing, EMNLP '11, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Wubben, S., van den Bosch, A., and Krahmer, E. (2012). Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1 of *ACL*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Wurst, D., Jones, D., and Luckner, J. (2005). Promoting Literacy Development with Students Who are Deaf, Hard-of-Hearing, and Hearing. *TEACHING Exceptional Children*, 37(5):56–62.

Xu, W., Callison-Burch, C., and Napoles, C. (2015). Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

Xu, W., Napoles, C., Pavlick, E., Chen, Q., and Callison-Burch, C. (2016). Optimizing Statistical Machine Translation for Text Simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Xue, L., Constant, N., Roberts, A., Kale, M., Al-Rfou, R., Siddhant, A., Barua, A., and Raffel, C. (2021). mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Yamada, I., Asai, A., Shindo, H., Takeda, H., and Matsumoto, Y. (2020). LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6442–6454, Online. Association for Computational Linguistics.

Yaneva, V., Temnikova, I., and Mitkov, R. (2015). Accessible Texts for Autism: An Eye-Tracking Study. In *Proceedings of the 17th International ACM SIGACCESS Conference on Computers & Accessibility - ASSETS '15*, ASSETS '15, pages 49–57, New York, NY, USA. Association for Computing Machinery.

Yang, Y. and Chen, L. (2018). A Literature Review on Individual Differences in Second Language Acquisition. *International Journal of Linguistics*, 10(6):72.

Yang, Y. and Wu, W. (2017). Study on Internal Factors Influencing Chinese College Students' Willingness to Communicate in English Based on Logistic Regression Analysis Model. In *Proceedings of 3rd International Symposium on Social Science (ISSS 2017)*, Dalian City, China. Atlantis Press.

Yatskar, M., Pang, B., Danescu-Niculescu-Mizil, C., and Lee, L. (2010). For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *NAACL HLT 2010 - Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Proceedings of the Main Conference*, ACL, pages 365–368, Los Angeles, California. Association for Computational Linguistics.

Yimam, S. M., Biemann, C., Malmasi, S., Paetzold, G., Specia, L., Štajner, S., Tack, A., and Zampieri, M. (2018). A Report on the Complex Word Identification Shared Task 2018. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 66–78, New Orleans, Louisiana. Association for Computational Linguistics.

Yimam, S. M., Štajner, S., Riedl, M., and Biemann, C. (2017a). CWIG3G2 - complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short*

Papers), pages 401–407, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Yimam, S. M., Štajner, S., Riedl, M., and Biemann, C. (2017b). Multilingual and cross-lingual ComplexWord identification. In *RANLP 2017 - Recent Advances in Natural Language Processing Meet Deep Learning*, pages 813–822. Incoma Ltd. Shoumen, Bulgaria.

Yunus, H. and Ahmad, N. A. (2022). Understanding The Definition and Characteristics of Dyslexia. *Malaysian Journal of Social Sciences and Humanities (MJSSH)*, 7(5):e001353.

Zeng, Q. T., Kim, E., Crowell, J., and Tse, T. (2005). A text corpora-based estimation of the familiarity of health terminology. In *ISBMDA 2006*, pages 184–92.

Zhang, X. and Lapata, M. (2017). Sentence Simplification with Deep Reinforcement Learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Zhao, S., Meng, R., He, D., Saptono, A., and Parmanto, B. (2018). Integrating Transformer and Paraphrase Rules for Sentence Simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Zhao, W. X., Zhou, K., Li, J., Tang, T., Wang, X., Hou, Y., Min, Y., Zhang, B., Zhang, J., Dong, Z., Du, Y., Yang, C., Chen, Y., Chen, Z., Jiang, J., Ren, R., Li, Y., Tang, X., Liu, Z., Liu, P., Nie, J.-Y., and Wen, J.-R. (2023). A Survey of Large Language Models.

Zhao, Y., Chen, L., Chen, Z., and Yu, K. (2020). Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, the Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, the Tenth AAAI Symposium on*

Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, pages 9668–9675. AAAI Press.

Zhu, Z., Bernhard, D., and Gurevych, I. (2010). A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, volume 24 of *COLING*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

Żulewska-Wrzosek, J. (2021). Problems of patients with aphasia in a psychosocial context. *Praca Socjalna*, 36(1):121–137.

Part I

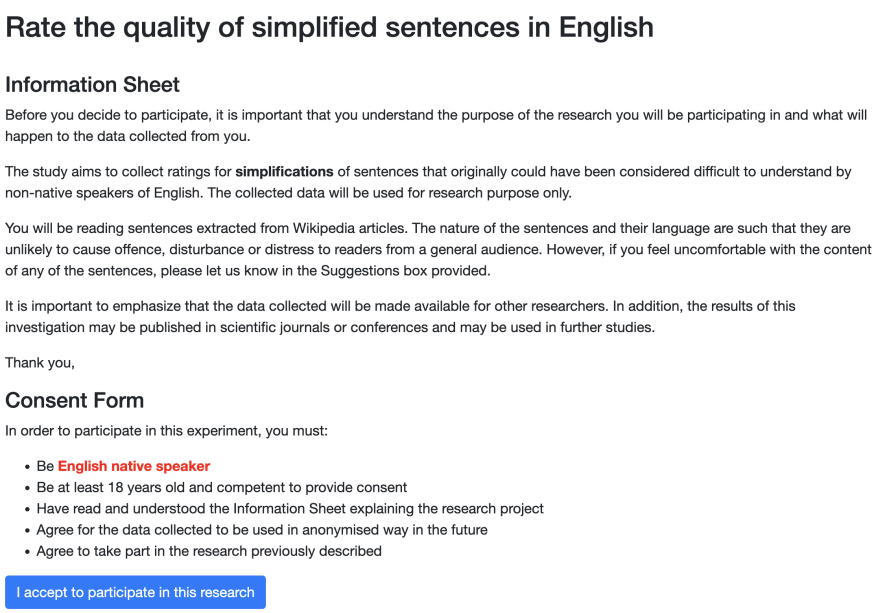
Appendix

Appendix A

Sentence Simplification Human Evaluation

Our interface is based on the one proposed by Kriz et al. (2019), and the consent form is based on Alva-Manchego et al. (2020a).

A.1 Consent Form



Rate the quality of simplified sentences in English

Information Sheet

Before you decide to participate, it is important that you understand the purpose of the research you will be participating in and what will happen to the data collected from you.

The study aims to collect ratings for **simplifications** of sentences that originally could have been considered difficult to understand by non-native speakers of English. The collected data will be used for research purpose only.

You will be reading sentences extracted from Wikipedia articles. The nature of the sentences and their language are such that they are unlikely to cause offence, disturbance or distress to readers from a general audience. However, if you feel uncomfortable with the content of any of the sentences, please let us know in the Suggestions box provided.

It is important to emphasize that the data collected will be made available for other researchers. In addition, the results of this investigation may be published in scientific journals or conferences and may be used in further studies.

Thank you,

Consent Form

In order to participate in this experiment, you must:

- Be **English native speaker**
- Be at least 18 years old and competent to provide consent
- Have read and understood the Information Sheet explaining the research project
- Agree for the data collected to be used in anonymised way in the future
- Agree to take part in the research previously described

I accept to participate in this research

Figure A.1: Human evaluation consent form.

A.2 Instructions

Please Note

- You have to be an **English Native Speaker**.
- You have to complete the ratings for all sentences. **All fields are required.**

Instructions

In this task, you will be given 5 source sentences and each source sentence has 4 simplified sentences from different systems. The goal is to judge each simplified sentence using 1-5 rating scale. You need to read each source sentence and its simplified sentences then give your opinions on three aspects:

- **Fluency** (or Grammaticality): is it grammatically correct and well-formed?
- **Simplicity**: is it simpler than the original sentence?
- **Adequacy** (or Meaning preservation): does it preserve meaning of the original sentence?

Use the sliders to indicate how much you agree with the statements (1 = Strongly disagree, 5 = Strongly agree).

Some clarifications:

- It is valid for the Simplified version of an Original sentence to be composed of more than one sentence. Splitting a complex and long sentence into several smaller ones helps readability sometimes. However, it is up to you to judge if the splitting actually made the sentence easier to read/understand or not.
- Different systems may produce the same simplified sentences, please judge accordingly.
- Fluency should be judged looking solely at the Simplified sentence. In your rating, mainly consider the grammatical and/or spelling errors, but also 'how well' (or natural) the sentence reads.
- Adequacy (or meaning preservation) and Simplicity should be judged looking at both the Original and Simplified versions. Judge whether or not the changes made preserved the Original meaning or not, and if they made it easier to understand, respectively. What if Original and Simplified are exactly the same? As the question in the form states, we ask you to judge if Simplified is "easier to understand" than Original. This implies that changes should have been made.
- It is very likely that Simplified does no have all the details that Original presented. When scoring Adequacy, it is up to you to judge the impact those changes had in the meaning of the sentence.
- Judging the quality of a simplification is subjective. Each person has their own opinion on what is fluent, adequate or simple. That is why we are collecting a big number of judgments, so that we can study the agreement/disagreement of the ratings. This is also why we do not provide you with examples: it is a way to prevent our own judgement biases to affect your personal judgments.

Please read the instructions carefully.
Thank you!

Figure A.2: Human evaluation instructions.

A.3 Examples

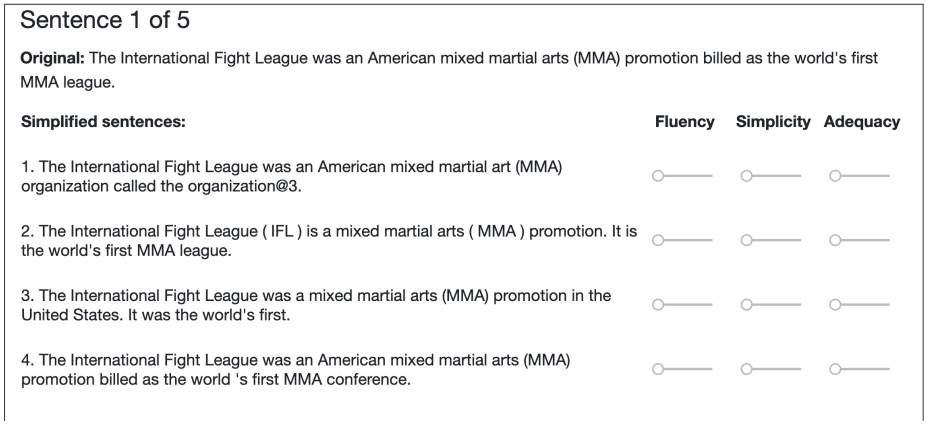


Figure A.3: Human evaluation examples.