Automatic sentence simplification using controllable and unsupervised methods

Louis Martin

THÈSE DE DOCTORAT DE SORBONNE UNIVERSITÉ

École Doctorale N° 130
École Doctorale Informatique, Télécommunications et Électronique

Discipline : Informatique

Soutenue publiquement le 27/10/2021, par :

Louis MARTIN

Automatic Sentence Simplification using Controllable and Unsupervised Methods

Simplification automatique de phrases à l'aide de méthodes contrôlables et non supervisées

---

Devant le jury composé de :

Mirella LAPATA, Full Professor, University of Edinburgh	Rapporteure
Sara TONELLI, Head of Research Unit, Fondazione Bruno Kessler	Rapporteure
Thomas FRANÇOIS, Professeur des universités, UCLouvain	Examinateur
Marc LELARGE, Directeur de recherche, Inria	Examinateur
Benoît SAGOT, Directeur de recherche, Inria	Directeur de thèse
Éric DE LA CLERGERIE, Chargé de recherche, Inria	Co-encadrant
Antoine BORDES, Directeur de recherche, Facebook AI Research	Co-encadrant

Acknowledgements

I wish to thank the members of my thesis committee - Sara Tonelli, Mirella Lapata, Thomas François, and Marc Lelarge – for generously offering their time in reviewing this manuscript and evaluating my work in the defense to come. The rest of these acknowledgements will be in French.

Je remercie sincèrement Benoît Sagot ainsi qu'Éric de la Clergerie, mes directeurs de thèse. Je les remercie pour le temps qu'ils ont dédié à mon encadrement et la bienveillance dont ils ont fait preuve tout au long de ma thèse pour m'accompagner et m'aiguiller dans ma recherche. Leur pédagogie, leurs encouragements et le profond intérêt qu'ils ont porté à ma réussite ont formé un support exceptionnel pour surmonter toutes les épreuves de ces trois ans de thèse. Mention spéciale pour leur positivité et bonne humeur à toute épreuve, qui ont été un moteur inépuisable de motivation durant cette thèse.

Merci ensuite à Antoine Bordes, mon co-directeur de thèse à Facebook. Malgré son emploi du temps très chargé, Antoine s'est toujours consacré avec dévouement à mon encadrement et a su rentrer en profondeur sur chaque sujet. Il m'a appris à tirer le meilleur de moi-même et à sortir de ma zone de confort. Merci de m'avoir soutenu dans tous mes projets, même ceux qui sortaient du cadre de ma recherche, ainsi que d'avoir fait preuve de tolérance face mes plus folles lubies de sur-ingénierie.

Merci ensuite à tous mes collègues, notamment à Facebook : Angela, Rahma, Mathilde, Marieanne, Pauline, Sablayrolles, Defossez, Timothée, Guillaume, Léonard, Alexis Sensei, et bien évidemment M. Stock. Dédicace spéciale à l'incroyable équipe de restauration de Facebook Paris et notamment à Pini, Thibaut et Julien qui m'ont permis de prendre plusieurs kilos en dégustant leurs mets délicieux matin, midi, goûter et soir. Merci à Patricia, notre maman à tous à FAIR Paris, pour sa bienveillance et son soutien à toute épreuve. Merci à

Jérémy Rapin et Clotilde pour leur aide précieuse dans mon développement personnel et la préparation de ma future carrière. Je remercie chaudement l'équipe ALMAnaCH d'Inria Paris et notamment Clémentine, Benjamin et Pedro. Merci à mes co-auteurs et notamment Fernando que je n'ai rencontré qu'une fois au détour d'un poster, mais que j'ai l'impression de connaître comme un ami de longue date.

Merci à tous mes professeurs de lycée et de prépa, en particulier M. Imperor. Ils ont su transmettre leur passion pour les sciences, et sans eux cette thèse n'aurait probablement pas existé.

Je tiens enfin à remercier mes proches. Merci à Amandine pour ta présence et ton immense soutien au jour le jour, d'être restée éveillée jusqu'au bout de la nuit dans mes pires charrettes, de m'avoir forcé à réviser quand j'en avais besoin, d'avoir édité mes bibtex sans avoir de notions de latex, et d'avoir créé les plus belles des animations Powerpoint. Merci à mes frères Paul et Hugo, des piliers sans faille sur qui je sais que je pourrai toujours compter. Merci à mon père Bernard qui m'a inculqué la curiosité, le goût pour la science, le sérieux mais aussi l'humour et la tolérance. Et enfin merci à ma mère, Élisabeth, d'avoir toujours été un exemple d'empathie et d'ouverture vers les autres. Merci d'avoir toujours cru en moi et d'être ma première fan. Je lui dédie ce manuscrit.

Contents

Chapter 1

Introduction

Text simplification is the task of making a text easier to read and understand. This objective may be reached by reducing the lexical or syntactic complexity of the text while preserving the original meaning as much as possible.

Text simplification has a wide variety of useful societal applications, for example increasing accessibility for those with reading difficulties, such as people cognitive disabilities with aphasia [Carroll et al., 1998], dyslexia [Rello et al., 2013], or autism [Evans et al., 2014], but also for non-native speakers [Paetzold and Specia, 2016b], people with low literacy [Watanabe et al., 2009], children with reading difficulties [Gala et al., 2020], or deaf and hard-of-hearing adults [Alonzo et al., 2021].

While the number of people struggling with reading difficulties is important, automatic text simplification still faces many challenges preventing its application for greater public. Simplification models are still limited in the types of rewriting operations that they can perform. For instance they succeed at dropping some unimportant content or replacing complex words with simpler ones most of the time, but still struggle in rephrasing larger chunks of text, splitting sentences, or simplifying the sentence structure. Besides, text simplification is hard to define, in part due to the fact that there is not one unique type of simplification but many, varying depending on the target audience. A simplification that makes a text easier to read and understand for a non-native speaker will probably not be easy to understand for someone with cognitive disabilities. However current training datasets and systems do not take this specificity into account and consider text simplification

as a one-size-fits-all task. In addition to limiting the applications to being adapted only to a certain type of reading difficulties, the most simplification research focuses on the English language, leaving the vast majority of non-English speakers devoid of simplification tools. Existing good quality training datasets come from a restricted number of sources (e.g. Simple English Wikipedia or learning materials for children or second language learners), which either do not exist in other languages, or would require a substantial amount of work to reproduce. In this work we aim at tackling these challenges with the main goal of creating a tool for helping simplifying documents in French for people with cognitive disabilities.

We try to answer the following questions:

- How can we correctly evaluate simplification models given the wide diversity of simplification types?
- Can we make models flexible enough so that they adapt to each audience?
- Can we develop language-agnostic methods to create simplification systems?

1.1 Societal Impact and Challenges

Most of the info we receive on a daily basis is in textual form whereas it is in the form of emails, news articles, legal documents, and most of it is not easy to read and understand.

Information is hard to read Crucial information can be hard to read. For instance employment contracts or administrative documents are of paramount importance to individuals but are too often obscured with complicated legal or administrative language and very hard to understand for the layman. Even mainstream information sources such as news articles or encyclopedic articles can be written with long intricate sentences spanning multiple lines with specific vocabulary.

Reading disabilities are common In addition to texts that can be written in complicated language, many people suffer from reading difficulties. Around the world about 793 million

people struggle with low literacy alone according to UNESCO in 2011 [1]. Additionally, people can suffer from various disabilities impairing reading ease such as aphasia, dyslexia, autism, or deafness. Second language learners such as Chinese people learning French face reading challenges as well. The content they are facing is not written in their native language. For instance there are 6.3 million English Wikipedia articles for 370 million native speakers compared to "only" 1.7 million Spanish articles for 470 million native speakers (as of May 2021). And this is even more true for native speakers of low resource languages.

In these conditions, providing people with access to simpler texts is an important step towards inclusion and better accessibility. Given the scale of the demand for simplified text, this can only be achieved with the help of automatic assistive tools, that can produce a tentative simplification for a large amount of input documents. Still, the inherent complexity of the task and errors produced by automatic simplification models, make professional human post-editing indispensable. Editors can thus edit the proposed automatic simplifications to remove errors and reformulate sentences for the production of more accurate simple texts.

1.2 The Cap'FALC Project: Improving Accessibility of French Texts with Automatic Text Simplification

This thesis has been conducted within the Cap'FALC French accessibility project that we describe in this section. Improving accessibility with the help of automatic text simplification for languages other than English has received more and more attention with recent initiatives such as the Portuguese PorSimples project [Aluísio and Gasperin, 2010], the Spanish Simplext project [Saggion et al., 2011, 2015], the Belgian French AMesure project [François et al., 2020], or the French Alector project [Gala et al., 2020].

With a similar objective, the French Cap'FALC project has the ambitious goal of creating a tool to simplify complex documents with the FALC method, which is the equivalent to Easy-to-Read[2] for French[3], for people with cognitive disabilities.

[1] http://www.unesco.org/new/fr/member-states/single-view/news/8_september_international_literacy_day_793_million_adults/

[2] https://www.inclusion-europe.eu/easy-to-read/

[3] See Chapter 2 for a precise definition

FALC documents are in high demand People with cognitive disabilities have difficulties accessing important administrative or medical information specific to their situation, hindering their autonomy and integration. The demand for having simple FALC documents is strong, both from the readers and from the entities that want their documents to be accessible and understood (city halls, hospital, museums, private companies...). For instance during the COVID-19 pandemic, French citizens had to fill a complicated certificate to go out of their residence for grocery shopping or medical appointments. Its simplification in FALC greatly increased the autonomy of people with disabilities, as can be seen in Figure 1.1.

However there are not enough professional accredited editors for meeting the demand of FALC documents. FALC documents are currently mostly created in ESATs (*"établissement et service d'aide par le travail"*). ESATs propose adapted jobs for people with disabilities. Some ESATs have specialized workshops for the creation of FALC documents where persons are trained for the transcription of complicated documents into simpler FALC documents. The number of ESATs producing FALC documents amounts to about twenty in France, which is not enough to satisfy all the requests for FALC documents.

The Cap'FALC tool Cap'FALC aims at creating an open-source AI-augmented tool to assist professional editors in creating FALC documents in an easier fashion. It will do so by providing an easy-to-use interface that proposes candidate simplifications of an input document by using latest automatic simplification research presented in this thesis. It is important to note that the tool **will not replace professional editors** but rather assist them for easier and faster transcription in order to meet the growing demand.

Stakeholders Cap'FALC is a partnership of 5 actors:

- **UNAPEI**: French National association for people with cognitive disabilities and their families. UNAPEI groups more than 500 local associations and closely works with ESATs where simple transcriptions of documents are created. UNAPEI pilots the Cap'FALC project and makes the link between research and people with cognitive disabilities that benefit from FALC.

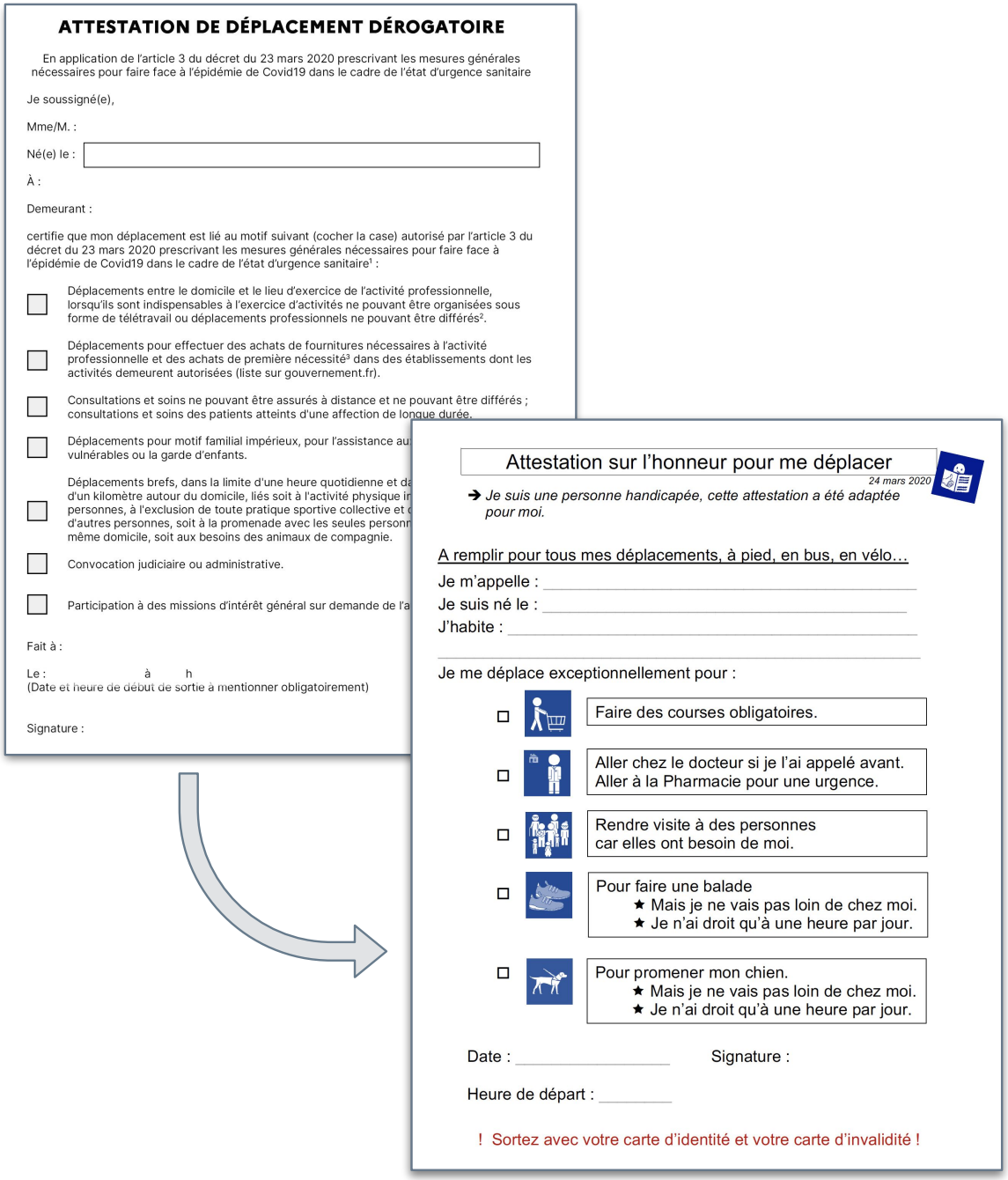


Figure 1.1: Example of a FALC document transcription. On the left the original certificate had to be used in France during the first COVID-19-related lockdown. On the right its FALC equivalent. The original document uses complicated language and verbose references to law articles, while the simplified version uses plain language, shorter sentences, larger font, and pictograms to increase readability. This FALC version was a life-saver for people with cognitive disabilities. It was created by the "Adapei du Doubs" association.

- **Inria**: National Institute for Research in Computer Science and Automation. Inria is one of the two research partners of the Cap'FALC project supervising this PhD thesis. This research was conducted in the ALMAnaCH team of Inria Paris, co-supervised by Benoît Sagot and Eric de la Clergerie.
- **Facebook AI Research (FAIR)**: Fundamental research lab of the Facebook company. FAIR is the second research partner supervising and funding this PhD thesis. This work was supervised by Antoine Bordes and used for the most part using the FAIR computing infrastructure.
- **French Secretary of State for people with cognitive disabilities**: Led by Minister Sophie Cluzel, the Secretary of state was instrumental in initiating the project, and supports the initiative.
- **Malakoff Humanis**: Non-profit social protection group. Malakoff Humanis helped finance the project and the development of the Cap'FALC tool.

1.3 Automatic Sentence Simplification

Source	The second **largest** city of Russia ~~**and one of the world's major cities**~~, St. Petersburg has played **a vital** role in Russian history.
Simplification	**St. Petersburg** is the second **biggest** city of Russia. St. Petersburg has played **an important** role in Russian history.

Table 1.1: **Example of sentence simplification** Differences between the two sentences are boldfaced.

Text simplification aims at making a text easier to read and understand for a target audience while preserving most of its meaning. We need to define three key notions here:

- What is a "text"? What is the granularity we want to work on?
- How do we define "easier to read and understand"? What is a definition of simplicity?
- What do we mean by "keeping most of its meaning"? How much of the original meaning do actually we want to keep?

We illustrate the task of automatic sentence simplification with an example in Table 1.1. This example features lexical simplification (*"largest"* becomes *"biggest"* and *"vital"* becomes *"important"*), sentence splitting (the original sentence is split in two), and the deletion of some unnecessary details (*"one of the world's major city"*).

Text Simplification at Different Granularities Here we can differentiate 3 levels of granularity for text simplification: **word-level** simplification (i.e. lexical simplification), **sentence-level** simplification, and **document-level** simplification. Most target applications will work at the document-level: one usually reads and tries to understand a whole document and more rarely a single sentence but never a single word. However working at a smaller granularity is easier and allows to make measurable research progress, this is why text simplification has historically been focused more on word-level simplification [Carroll et al., 1998, Devlin and Tait, 1998, Biran et al., 2011, Bott and Saggion, 2011a].

Then the field has transitioned to end-to-end sentence-level simplification [Zhu et al., 2010, Wubben et al., 2012, Zhang and Lapata, 2017, Zhao et al., 2018] allowing for more diverse type of simplification operations to be represented such as phrase dropping, substitution, reordering, or sentence splitting. Research is now transitioning to the ultimate step of document-level simplification [Alva-Manchego et al., 2019b] and consider novel operations such as coreference resolution, reordering ideas, summarizing content, or generating explanations.

In this work we focus on **Sentence Simplification**.

"Easier to read and understand" When people struggle to read a sentence, it can be due to various aspects: complicated words that the reader does not know, long sentences that are hard to keep in working memory, sentences with too many open dependency nodes, sentences with ambiguous meaning, or sentences with unclear logic. In order to make a complicated sentence easier to read, humans use a variety of simplification mechanisms to solve the aforementioned problems and we should expect Sentence Simplification systems to do the same. These rewriting operations include replacing complicated words with simpler ones (lexical simplification), reordering words or ideas, splitting long sentences in multiple

shorter sentences, resolving ambiguous coreferences, transforming passive sentences into active sentences, removing cluttering non-essential details.

While current Sentence Simplification simplification systems perform fairly well on light editing such as lexical simplification and content removal, only some achieve sentence splitting, passive active transformation or coreference correctly. Heavier rephrasing and rearranging ideas in a more logical order is even more rare.

"Keeping most of its meaning" In the previous example, we saw that simplification usually includes removing some content, but how much content exactly should we remove? Where do we draw the boundary? This depends on the application and target audience. For instance, simplifying texts for people with cognitive disabilities will strip most of the original text to focus only on the core ideas (see Section 1.2). Even for the same category of target audience, different levels of simplicity can be achieved by removing more or less content. Texts simplified for children with lower grade levels usually contain less words than the associated texts for higher grades in the NEWSELA corpus [Xu et al., 2015]. The more content you remove, the easier the text will be, highlighting an important **tradeoff** between **simplicity** and **meaning preservation**.

1.4 Thesis Structure

While text simplification is very important, research in automatic simplification still faces various challenges. We aim at tackling several of these challenges in this thesis, related to three major directions: evaluation, adaptability of models, and multilinguality.

Part I: Related Work In this part we survey the general simplification literature, and detail more specific and relevant related work in each chapter.

We first describe how simplification systems are trained and evaluated in Chapter 2. We give an overview of different guidelines for simple language: Basic English, Simple English Wikipedia, and FALC. We then describe the different training sets used in data driven Sentence Simplification. Finally we present the evaluation sets and automatic metrics

that are traditionally used.

In Chapter 3 we give an overview of different approaches for data-driven automatic Sentence Simplification.

Finally in Chapter 4 we present unsupervised approaches to Sentence Simplification that overcome the problem of lack of data, especially in languages other than English.

Part II: Evaluating Sentence Simplification Systems We highlighted that Sentence Simplification is hard to define: How do we define "simple text"? How to take into account the variety of possible simplifications? How much of the original meaning do we want to preserve? As a consequence, it is also hard to properly evaluate simplification systems. Current automatic metrics and evaluation data have various limitations.

In Chapter 5, we explore how different features of the simplification correlate with human judgements when no reference simplifications are available.

Then we present in Chapter 6 an effort to streamline evaluation of Sentence Simplification simplification with the EASSE library. We show how we gathered and normalized all standard simplification metrics and how we added word-level and quality estimation features for investigating Sentence Simplification systems.

Most automatic metrics rely on using reference simplifications. We show in Chapter 7 that current evaluation datasets are not diversified enough and do not match typical human simplifications. As a result we propose a new evaluation dataset, ASSET, that is more varied and deemed simpler than previous evaluation datasets.

Finally, in Chapter 8, we experiment with more recent neural-based evaluation metrics and discover that current automatic metrics have very low correlation with human judgements of system-generated simplifications, which might be linked to spurious correlations. These correlations completely vanish when trying to evaluate human-written simplifications, thus raising concern about automatically evaluating Sentence Simplification systems that close the gap with human performance.

Part III: Towards more Adaptable Simplification Systems Simplification cannot be defined in a unique manner: for a given source sentence, multiple simplification candidates

are acceptable. We argue that different audiences need different types of simplifications. As a result, we propose in Chapter 9 ACCESS, a model that can be adapted on demand to the type of simplifications needed. This is achieved by conditioning the model on simplification-related features at train time such as length, syntactic and lexical complexity, amount of rewriting. The model can then generate simplifications with given length or lexical complexity at test time.

Part IV: Extending Sentence Simplification to Other Languages Automatic text simplification has also suffered from a lack of high quality data to train strong systems. This has restrained the application of simplification systems mostly to English. Even in this relatively "high resource" language, data is automatically gathered using imperfect methods, resulting in models having flaws, such as not preserving the meaning (hallucinations), not being grammatical, or not simplifying enough. In Chapter 10, we propose MUSS, an approach to overcome the challenge of training data and show that we can train models in any language that improve fluency, meaning preservation, and simplicity. We do so by mining paraphrases from the web as training data, that we then use to train controllable simplification models based on the ACCESS method.

Given the focus of the Cap'FALC project on the French language, we focus more specifically on this language in Chapter 11. We train CamemBERT, the first masked language model in French. We then leverage our pretrained model to create a strong simplification model using the data that we collected in previous Chapter 10.

Part V: Conclusion and Perspectives We finally summarize and conclude on this thesis in Section 7.6. We highlight relevant areas of future work regarding simplification evaluation, transitioning to document-level simplification, generalizing our methods to other tasks, improving factual consistency, and applying our methods to FALC.

Part I

Related Work

Chapter 2

Evaluation of Text Simplification systems: Guidelines, Datasets and Metrics

2.1 Simplification Guidelines

Sentence Simplification is the task of rewriting a text in a simpler manner while keeping as much as the original meaning as possible. Such simplification can be accomplished in many different ways and levels of final simplicity. Multiple set of guidelines have been created in order to standardize the process of simplifying texts. Guidelines also provide a way to guarantee a minimum quality of output simplifications.

2.1.1 FALC

Such guidelines include the French **Facile À Lire et à Comprendre** method, abbreviated **FALC**. FALC is the French declination of the European **easy-to-read**. The FALC method details more than 100 rules and guidelines[1] to write texts that comply with the easy-to-read standard. Validated easy-to-read and FALC documents can be recognized with the associated logo displayed in Figure 2.1.

[1]English: `https://www.inspiredservices.org.uk/wp-content/uploads/EN_Information_for_all.pdf`

Figure 2.1: Logo of FALC.

FALC guidelines give advices on which words to use, on how to structure sentences, on how to order ideas in a whole document and also help the writer choose a pertinent font and page layout. Table 2.1 illustrates some FALC guidelines.

Word-level
Use easy to understand words that people will know well. Percentages (63%) and big numbers (1,758,625) are hard to understand. Use the same word to describe the same thing throughout your document. Use examples to explain things. Try to use examples that people will know from their everyday lives.
Sentence-level
Always keep your sentences short. Use positive sentences rather than negative ones where possible. Use active language rather than passive language where possible. Speak to people directly. Use words like "you" to do this.
Document-level
Always put your information in an order that is easy to understand and follow. Group all information about the same topic together. It is OK to repeat important information. It is OK to explain difficult words more than once.
Design and Layout
Never use a background that makes it difficult to read the text. Always use a font that is clear and easy to read. Never use italics.

Table 2.1: FALC guidelines

Research in Automatic Text Simplification has mostly focused on the word-level and sentence-level simplification aspects. Although it can be argued that true Text Simplification should take the whole document into account, research in this direction is fairly limited [Alva-

Manchego et al., 2020]. Furthermore, aspects pertaining to layout and design readability are not yet considered by the Automatic Text Simplification community, even though a some works exist in the field of Human-Computer Interactions [Alonzo et al., 2020].

2.1.2 Basic English

Basic English was created as an aid for second-language learners of English [Ogden, 1930]. As such it is a simplified subset of regular English and grammar restrictions and a small controlled vocabulary. For instance it requires to use a restricted vocabulary of only 850 basic words and only 18 verbs. It includes a simple grammar to modify the vocabulary for additional meaning such as "Nouns are formed with the endings -er (as in prisoner) or -ing (building)." or "Negatives can be formed with un- (unwise)."

2.1.3 Simple English Wikipedia

The guidelines from Simple English Wikipedia define another set of advices for writing simple text.[2] Simple English Wikipedia was created as an alternative to English Wikipedia where all encyclopedic articles are written in simple language, aimed at people who are learning English or children. These guidelines are inspired from Basic English but does not enforce the restricted Basic English vocabulary. For instance, when a word is complicated but cannot easily be replaced with a simpler words, the guidelines advise to explain the complex word instead in parentheses e.g. "blood, toil (hard work), tears, and sweat". Some guidelines of Simple English Wikipedia are illustrated in Table 2.2

2.1.4 Guidelines are not Unique

FALC, Basic English and SEW guidelines give different perspectives on how to write simple texts. They are aimed at different audiences and hence have differences and do not always agree. FALC is aimed at making texts accessible for people with cognitive disabilities who are often fluent in the language but have trouble understand long and intricate sentences.

[2] `https://simple.wikipedia.org/wiki/Wikipedia:How_to_write_Simple_English_pages`

Word-level
Write your words normally, as you would in speaking to ordinary people. Look for your words in the word lists. Try to use the simplest word list (such as Basic English). Look for a Basic English verb in past, present or future only. Always start by using simple sentences. Do not use idioms (one or more words that together mean something other than what they say). Do not use write in the second person. Good encyclopedia articles are never addressed to "you". Do not make statements about "you".
Sentence-level
Change to active voice. Example: change from "The bird was eaten by the cat." (passive voice) to "The cat ate the bird." Try to avoid compound sentences – those with embedded conjunctions (and, or, but, however, etc.) – when possible. Try not to use compound-complex sentences, with multiple independent and dependent clauses.

Table 2.2: Example of Simple English Wikipedia guidelines

Basic English and SEW are aimed at learners of English as a second language or children, who might struggle more with complicated vocabulary.

These guidelines have similar rules such as writing short sentences or using simpler words, but they also differ in a few ways. For instance FALC advises writers to directly address the user by using the second person “you”, to make people with disabilities more engaged with the text, whereas SEW explicitly discourage the use of the second person due to the Encyclopedic nature of the texts.

Even though SEW guidelines are inspired from Basic English, they also express concern over using such a restricted vocabulary of 850 words. For instance, as illustrated in the SEW guidelines, the sentence "I have nothing to offer but blood, toil, tears, and sweat" would be rewritten "I have nothing to offer but blood, hard work, drops from eyes and body water". Replacing "tears" with "drops from eyes" and "sweat" with "body water" makes the text less fluent and more difficult to understand.

We will see in Part III that simplification cannot be uniquely defined and that it is crucial for automatic Sentence Simplification systems to have a way to be adapted to the audience and context.

2.2 Training Datasets

Research in Sentence Simplification has focused around sentence-level data-driven methods inspired from machine translation, that we will describe in more details in Section 3. These methods require large amounts of parallel data in the form of complex sentences and their associated simple sentences. Finding hundred of thousands of parallel complex-simple sentence pairs is not an easy task as they do not naturally occur in the web in large quantities, except for a few sources that we will cover in this section.

2.2.1 Training with English Wikipedia and Simple English Wikipedia

The most prominent source of parallel simplification data that was used, relies on English Wikipedia[3] (EW) and Simple English Wikipedia[4] (SEW). As previously mentioned, SEW is a version of Wikipedia where contributors are explicitly asked to write in a simple language, with some rules inspired from Basic English. The vast majority of encyclopedic articles that appear in SEW also appear in EW, therefore providing a natural document-level alignment of complex-simple texts. Complex-Simple sentence pairs are then extracted from matching articles by automatically aligning sentences with similar meaning using term-based similarity heuristics.

Zhu et al. [2010] introduce PWKP, a dataset of 108k parallel complex-simple sentences extracted from English Wikipedia-Simple English Wikipedia (EW-SEW) using sentence-level TF-IDF similarity for sentence alignment. To allow for the sentence splitting operation to be represented in their dataset, they merge pairs where complex sentences are the same and simple sentences are adjacent, resulting in a 1-to-n mapping.

Woodsend and Lapata [2011] also align EW-SEW first by aligning paragraphs and then at the sentence-level using TF-IDF. They additionally use revision history of SEW to create complex-simple sentence pairs. The initial version of the sentence is used as the source and the edited as the target. They only use revisions using simplification-related keywords such as *simple*, *clarification*, *grammar*.

[3] `https://en.wikipedia.org/`
[4] `https://simple.wikipedia.org/`

Coster and Kauchak [2011a] similarly create a parallel sentence-level simplification dataset from EW-SEW. They first align every paragraph in the simple article with paragraphs from the complex article when the TF-IDF similarity is above a certain threshold. Then they find sentence-alignments using a dynamic programming approach based on [Barzilay and Elhadad, 2003], and computing the inter-sentence similarity also with TF-IDF. Their method allows for n-to-n alignments for sentences (with $n \leq 2$). The extraction and alignment process results in 137k aligned complex-simple sentence pairs. An automatic analysis of these aligned pairs finds that multiple rewriting operations are represented: 65% of pairs contain rewording, 47% deletions, 34% reorders, 31% merges of multiple complex words into one simple word and 27% of splits of complex words into multiple simple words. Kauchak [2013] further updated this dataset with more recent wikipedia data and improved text processing to create 167k aligned sentence pairs.

The sentence alignments from [Zhu et al., 2010, Kauchak, 2013, Woodsend and Lapata, 2011] were later combined into the **WIKILARGE** dataset [Zhang et al., 2017]. The resulting dataset combines 296,402 sentences. WIKILARGE has been used as the de facto standard of EW-SEW alignments in multiple subsequent works [Dong et al., 2019, Vu et al., 2018, Mallinson and Lapata, 2019, Kriz et al., 2019].

More recently Jiang et al. [2020] have introduced the WIKI-AUTO dataset extracted from EW-SEW with a better alignment method that uses neural-CRF models. The authors first align paragraphs of the same article in its complex and simple version. They do so computing pairwise sentence similarities using a BERT language model [Devlin et al., 2019] by averaging or taking the max of pairwise sentence similarities between each two pairs of paragraphs. Then paragraphs are aligned if they have a high semantic similarity and appear in similar positions in the document. Two complex paragraphs can also be merged if they are consecutive and have high semantic similarity with the same simple paragraph. Sentence alignment is then computed for each pair of two aligned paragraphs using the neural CRF approach. The CRF takes into account pairwise sentence similarities using the aforementioned finetuned BERT model but also alignment label transitions using a fully connected neural network based on 4 handcrafted features (e.g. if the alignement labels are consecutive). The CRF is trained on a set of manual alignments of complex-simple articles.

The resulting simplification dataset dubbed WIKI-AUTO contains 488,332 sentence pairs, an increase over the 296,402 sentence pairs from WIKILARGE that can be attributed to the new alignment method and the more recent dumps of wikipedia used. The authors show that models trained on their new data obtains better scores although not by a large margin for WIKI-AUTO vs. WIKILARGE.

Similar to [Woodsend and Lapata, 2011] described previously, other methods have used the EW edit-history to create complex-simple sentence pairs, deemed **WIKISPLIT** [Botha et al., 2018]. Their approach focused on extracting natural sentence splitting examples from wikipedia as an improvement to the previous artificially created and unnatural sentence splitting dataset **WEBSPLIT** [Narayan et al., 2017], that we detail in Chapter 3. A sentence split sample is composed of a complex sentence C aligned with two consecutive simple sentences deemed $S1$ and $S2$. They extract these sample from different temporal snapshots of EW by matching sentences where C and $S1$ start with the same trigram and C and $S2$ end with the same trigram. Misaligned pairs are then filtered out using the BLEU [Papineni et al., 2002] similarity metric when either BLEU(C, $S1$) or BLEU(C, $S2$) is lower than a certain threshold. As a result, they obtain 1 million sentence split samples with greatly improved diversity over WEBSPLIT.

2.2.2 Newsela

Using automatic alignments of EW-SEW has been shown to produce noisy training data with some alignments where the simple sentence is not simpler (33%) or not related to the complex sentence (17%) [Xu et al., 2015].

As a result the NEWSELA dataset was proposed [Xu et al., 2015]. NEWSELA is composed of 1,130 news articles which were re-written in 4 different levels of simplicity by professional editors from Newsela[5]. Various sentence-alignments of NEWSELA exist. The most widely used alignment was performed in [Zhang and Lapata, 2017], where the authors aligned the documents into 94k sentence pairs. More recently [Jiang et al., 2020] used a CRF model (same as WIKI-AUTO in previous section) to create NEWSELA-AUTO, an alignment of 394k sentence pairs that is claimed to improve the performance of models trained with it. In

[5] https://newsela.com

preliminary experiments of Chapter 10, we however obtained lower performance using this recent alignment than with the previous one.

NEWSELA also comes with Spanish news articles that were aligned at the sentence-level by [Aprosio et al., 2019]. Even though sentences were aligned using the CATS simplification alignment tool [Štajner et al., 2018], some alignment errors remain and automatic scores should be taken with a pinch of salt.

Such professional datasets are better in terms of quality but however come with restrictive licenses that hinder reproducibility and widespread usage.

Simplification data is however hard to find in large quantities especially for languages other than English. In Chapter 10 we show how one can mine data from raw web data to train state-of-the-art unsupervised simplification models in any language.

2.3 Multi-Reference Human Evaluation Datasets

In order to evaluate automatically generated simplifications, previous work has compared the generated simplification with high quality reference simplifications using automatic metrics. In this section, we present the high quality human evaluation sets that are traditionally used in Sentence Simplification. Note that test set splits of the previously mentioned training datasets are also used to evaluate Sentence Simplification systems by comparing the prediction with the associated reference simplification of the dataset. However doing so might be less reliable than with multi-reference human evaluation sets.

2.3.1 TURKCORPUS

Xu et al. [2016] have proposed TURKCORPUS, a dataset composed of 2359 complex sentences (2000 validation and 359 test) extracted from Wikipedia where, for each complex sentence, 8 reference simplifications where collected using Amazon Mechanical Turk. Most simplified sentences are however very similar to the complex sentence with only a few lexical simplifications or word deletions, i.e. they are not adapted to the evaluation of fully fledged Sentence Simplification systems performing sentence splitting and more complex rewrite operations.

2.3.2 HSPLIT

Focused solely on Sentence Splitting, the HSPLIT [Sulem et al., 2018a] evaluation set was created using the same 2359 complex sentences as TURKCORPUS and provides 4 human references per source sentence. Each reference was created by only operating sentence splitting on the original complex sentence. This is therefore a good dataset for the evaluation of sentence splitting but does not generalize to Sentence Simplification in general.

2.3.3 No General Purpose Evaluation Dataset

TURKCORPUS and HSPLIT are however too restricted in the type of simplification operation that they can evaluate. In Chapter 7 we propose ASSET, a new dataset with simplifications containing a more varied set of rewriting operations that is judged simpler and improves the correlation of automatic metrics with human judgement.

2.4 Automatic Evaluation Metrics

Systems are typically evaluated across 3 dimensions.

- **Meaning Preservation** Does the simplified sentence retain the original meaning?
- **Fluency** Is the simplified sentence fluent and without grammatical errors?
- **Simplicity** Is the simplified sentence simpler than the original sentence?

Those three criteria can however not always be maximized at the same time, with for instance Simplicity and Meaning Preservation being strongly inversely correlated [Schwarzer and Kauchak, 2018].

While these aspects should ideally be evaluated by humans at the end of the road [Štajner et al., 2016b, Xu et al., 2016, Sulem et al., 2018b], it requires costly annotations and trained experts for good quality evaluation. Multiple automatic evaluation metrics have been proposed and used as proxies to human judgements. We hereafter present the main automatic metrics used in Sentence Simplification.

2.4.1 BLEU

Sentence Simplification methods were traditionally evaluated with metrics borrowed from machine translation such as BLEU [Papineni et al., 2002]. BLEU compares the generated simplification with ground-truth human simplifications and can be used in a multi-reference setting. It first computes n-gram precisions of the generated text compared to ground truth references, for n-gram lengths from 1 to 4. Then those n-gram precisions are combined into a single score using a geometric mean.

BLEU was however shown to have poor correlation with human judgements of simplicity [Xu et al., 2016], but also meaning preservation and fluency especially when rewriting operations such as sentence splitting are involved [Sulem et al., 2018b].

2.4.2 SARI

Xu et al. [2016] proposed SARI a new evaluation metric for text simplification that correlates better with humans ratings. SARI takes advantage of the fact that Text Simplification is a monolingual rewriting task and instead of comparing the automatic simplification only to references, it also uses the source sentence for a better analysis of rewriting performed. SARI compares the predicted simplification with both the source *and* the target references. It is an average of F1 scores for three n-gram operations: additions, keeps and deletions. For each operation, these scores are then averaged for all n-gram orders (from 1 to 4) to get the overall F1 score.

$$ope \in [add, keep, del]$$

$$f_{ope}(n) = \frac{2 \times p_{ope}(n) \times r_{ope}(n)}{p_{ope}(n) + r_{ope}(n)}$$

$$F_{ope} = \frac{1}{k} \sum_{n=[1,..,k]} f_{ope}(n)$$

$$SARI = \frac{F_{add} + F_{keep} + F_{del}}{3}$$

SARI has become the de facto metric for Sentence Simplification. We will however see

in Chapters 7 and 8 that it can have low correlations with human ratings of simplifications.

2.4.3 SAMSA

Without relying on references, SAMSA [Sulem et al., 2018a] evaluates the structural simplicity of a simplification. It makes strong assumptions on how sentences should be simplified: (1) each output sentence should contain a single semantic event (as described by UCCA [Abend and Rappoport, 2013] Scenes) (2) all semantic events should be kept between the source and simplification. A system that will perform only strong sentence splitting, will obtain the highest scores. SAMSA first creates a semantic parse of the source sentence and the simplification using the UCCA representation. Then it aligns tokens together and scores the simplification to penalize sentences that contain multiple UCCA scenes, that dropped scenes, or where a single scene is incorrectly split into multiple sentences.

SAMSA was not used very much in practice since its introduction probably due to two reasons. First SAMSA's strong assumptions on the simplification task make it unable to correctly evaluate simplifications where lexical simplification is more important than sentence splitting. Second SAMSA's approach and implementation make it very slow and cumbersome to run, which might hinder practical use.

2.4.4 FKGL & FRE

The Flesch-Kincaid Grade Level (FKGL) and Flesch Reading Ease (FRE) [Flesch, 1948, Kincaid et al., 1975] are two metrics aimed at measuring the readability of an input text.

Both measures are linear combinations of two features: average number of words per sentence, and average number of syllables per word. The first feature is a simple but strong proxy for structural simplicity. Shorter sentences are easier to understand and have less intricated syntax. We will show in Chapter 5 that this feature is one of the best predictor of sentence simplicity. The second feature is the average number of syllables per word which accounts for lexical complexity. Indeed longer words are less frequent (Zipf's Law for word frequencies), and word frequencies have been found to be a strong indicator of lexical complexity [Paetzold and Specia, 2016a].

These two features are then fitted to predict overall document complexity in English using a linear regression which gives the following two formulas:

$$FKGL = 0.39\frac{\text{total words}}{\text{total sentences}} + 11.8\frac{\text{total syllables}}{\text{total words}} - 15.59$$

$$FRE = 206.835 - 1.015\frac{\text{total words}}{\text{total sentences}} - 84.6\frac{\text{total syllables}}{\text{total words}}$$

Note that lower FKGL indicate simpler texts, while it is the opposite for FRE.

FKGL and FRE however have limits. They were established a long time ago as an army standard on a set of domain-specific documents in English. This implies that they might not apply to all type of documents and they would not be adapted for languages other than English. Furthermore, FKGL and FRE are document-level metrics and should be used as such. Using them to evaluate sentence-level readability as is common in the literature might not be optimal. Still, FKGL is one of the best predictor of simplicity according to our experiments in Chapter 8.

Chapter 3

Data-driven Sentence Simplification

Earlier Text Simplification methods have divided the problem into subtasks and approached each one off them independently such as lexical simplification [Carroll et al., 1998, De Belder and Moens, 2010, Specia et al., 2012, Biran et al., 2011] and syntactic simplification [Chandrasekar and Srinivas, 1997, Carroll et al., 1998, Siddharthan, 2006, Brouwers et al., 2014]. However more recently, data-driven methods inspired from machine translation have used a more holistic approach Text Simplification by using statistical or neural models to encompass multiple text rewriting operation in an end-to-end model. Most research has focused on Text Simplification at the sentence level, i.e. Sentence Simplification. We will focus on those methods in this chapter.

3.1 English Simplification Systems

The majority of research in Sentence Simplification has been focused on the English language, especially because of the availability of training and evaluation corpora in this language such as EW-SEW and NEWSELA (section 2.2).

Phrase-Based Sentence Simplification Statistical MT (SMT) methods such as Phrase-Based MT (PBMT) have been used on parallel complex-simple corpora such as EW-SEW to create simplification systems.

Coster and Kauchak [2011b] use PBMT [Koehn et al., 2007] to train an Sentence Simplification system (MosesDel) on 137k sentence pairs extracted from EW-SEW. They enhance the model with a phrasal deletion component to improve the model on this particular type of simplification operations. Sentence Simplification systems often operate too little modifications on the original sentence if trained without specific inductive biases.

Wubben et al. [2012] also modify PBMT for Sentence Simplification using EW-SEW data, but instead of adding a deletion component, they rerank the hypothesis based on a Levenshtein distance dissimilarity metric to force the model into making enough modifications. They show that this dissimilarity incentive improves the quality of simplifications. However the model still performs relatively few modifications on the original sentence and it does not handle sentence splitting.

Tree-based and Syntax-based Sentence Simplification Simplifying a sentence requires performing various structural rewriting operations such as sentence splitting, passive to active transformations, or phrase reordering. In order to capture those type of transformations, previous work has performed simplification by relying on the parse tree representations of sentences.

The Tree-based simplification model (TSM) [Zhu et al., 2010], is the first statistical model that handles splitting, dropping, reordering and substitution, thus covering lexical and syntactic simplification in the same model. It operates on the parse tree of the sentence, and the authors implement each operation independently with a set of task-specific rules and features.

Woodsend and Lapata [2011] learn Quasi-Synchronous Grammar rewrite rules using EW-SEW and the SEW revision history. The algorithm uses Integer Linear Programming to find the set of tree rewriting operations that produces the best simplified sentence that satisfies grammaticality and coherence constraints.

Bach et al. [2011] use a parse tree decomposition of the original sentence to generate simple sentences based on the subject-verb-object structure. They find the best candidates by ranking using various hand-crafted lexical and syntactic features.

Xu et al. [2016] propose a syntactic-based MT model augmented with paraphrases

extracted from the external paraphrase database PPDB [Ganitkevitch et al., 2013, Pavlick et al., 2015]. In PPDB, paraphrase rules are associated with 33 features such as translation probabilities, word-for-word lexical translation probabilities. Xu et al. [2016] incorporate 9 additional simplification-specific features such as length in characters and in words, number of syllables, or proportion of common English words. These features are then combined into a weighted sum to score paraphrase rules for simplification. The weights are fitted to maximize performance on the validation set of TURKCORPUS using metrics such as SARI or BLEU.

Semantic Methods Narayan and Gardent [2014] combine deep semantics with a phrase-based MT model in a system called **Hybrid**. They first produce a semantic Discourse Representation Structure to the complex sentence, then modify this representation using a probabilistic sentence splitting and deletion model to produce a set of simpler sentences. These simpler sentences are further simplified using a phrase-based MT system to account for substitution and reordering.

Neural Approaches The first neural approaches to Sentence Simplification are recent [Nisioi et al., 2017, Zhang and Lapata, 2017] compared to how widespread they have been in other tasks. Similarly to previous statistical approaches, they get inspiration from MT and adapt models for the task of sentence simplification.

Nisioi et al. [2017] are the first to train a basic neural sequence-to-sequence model with a simple two-layer LSTM with attention. Their model, called NTS, is trained on EW-SEW. Although they show that their model is the first that can jointly perform lexical simplification and content reduction, it still suffers from making very few changes to the input sentence. They resort to the method of always selecting the second beam hypothesis of the beam search instead of the first one to have the model make more modifications. This goes in the same line as the hypothesis reranking of [Wubben et al., 2012], although it is less intuitive to arbitrarily select an hypothesis given its rank in the beam search.

On the other hand, Zhang and Lapata [2017] use reinforcement learning to adapt a neural sequence-to-sequence specifically on the task of Sentence Simplification. They do so by

training using REINFORCE [Williams, 1992] on rewards computed for simplicity using the SARI metric, meaning preservation (neural semantic encoder), and fluency (language model). While this method can bring improved performance, it can also lead to reward hacking on SARI which has been shown to be imperfect [Sulem et al., 2018b].

Controllable Models For a given sentence, various simplifications can be acceptable, and they often vary simplicity-meaning trade-off. In order to account for this range of acceptable simplifications, Scarton and Specia [2018] and Nishihara et al. [2019] used controllable generation mechanisms. They showed that adding control tokens at the beginning of sentences can improve the performance of Seq2Seq models for Sentence Simplification. Plain text control tokens were used to encode attributes such as the target school grade-level (i.e. understanding level) and the type of simplification operation applied between the source and the ground truth simplification (identical, elaboration, one-to-many, many-to-one).

Methods using External knowledge Good quality simplification data is hard to find in sufficiently large quantities for a system to be able to model all simplification types. Approaches have relied on using auxiliary databases to augment the capacity of their models. As previously mentioned Xu et al. [2016] used PPDB [Ganitkevitch et al., 2013, Pavlick et al., 2015] to integrate paraphrasing in their syntax-based simplification system. A version of PPDB, dedicated to simplification was later proposed [Pavlick and Callison-Burch, 2016]. Simple PPDB is a subset of PPDB only containing simplification rules. Paraphrase rules are classified as simplifications using a supervised lexical simplification scorer. The authors created the labelled training data of the classifier by asking humans from Amazon Mechanical Turk to judge wether paraphrases are simplifications are not. Simple PPDB was later used to augment simplification models [Zhao et al., 2018]. Their model, DMASS-DCSS is augmented with these simplification rules using two mechanisms. Deep Critic Sentence Simplification (DCSS) adds a new training loss that fosters use of these simplification rules and also reweights the decoding probabilities to favor simplification rules. The Deep Memory Augmented Sentence Simplification (DMASS) component augments the neural model with a dynamic memory to record multiple key-value pairs for each rules in PPDB.

3.2 Simplification Systems for Other Languages

Simplification research has mostly been conducted in English. No SEW equivalents exist in other languages, preventing the training of data-driven simplification models. In this section we cover the different approaches to simplification in non-English languages

Brazilian Portuguese The PorSimples project aimed at proposing Sentence Simplification systems to assist authors in creating simple texts and to help people read web content. Aluísio et al. [2008] study the linguistic phenomena that make texts complex or simple. Six simple corpora and one corpus of complex texts are analyzed along the following criteria: size of sentences and words, number of relative clauses, appositions, subordinate and coordinate conjunctions, main and subordinate clause ordering, and number of simple words. They extract a set of simplification rules in Portuguese that serve as a base for a rule-based system.

Spanish Simplext is a similar project for Spanish Sentence Simplification [Bott and Saggion, 2011b, Saggion et al., 2015]. In [Bott and Saggion, 2011b], the authors release a corpus composed of 200 news articles that were manually simplified by trained experts for people with learning disabilities. They analyse the different types of simplification operations performed in manual simplification. They categorize the transformations in 4 categories: changes, insertions, deletions, and splitting. Due to the lack of large enough training corpora, they propose a modular system that combines rule-based lexical and syntactic simplification [Saggion et al., 2015].

Štajner et al. [2015b] latter use the data from the Simplext project to train statistical phrase-based MT models that perform equally well to the modular system of [Saggion et al., 2015] although it uses little training data. They observe that predictions from models trained on "lightly" edited simplifications are more grammatical although less simple than generations from models trained on "heavily" edited simplifications.

Italian Barlacchi and Tonelli [2013] present the first Sentence Simplification system for Italian, performing rule-based simplification in two steps: anaphora resolution and sentence-level syntactic simplification aimed at children with reading difficulties. The sentence-level

simplification identifies and retains only factual events based on tense and mood information, and expresses them in the present for better readability.

Brunato et al. [2015] present a resource composed of two simplification corpora for Italian. The first one is composed of 32 short novels that were manually simplified for children. This results in about 1000 aligned simplification samples, with around 4% of samples containing sentence splitting. The second corpus contains 24 documents that were independently simplified by teachers for L2 students with a B2 level in Italian. Only 68% of documents were aligned at the sentence level for this second heterogeneous corpus.

Tonelli et al. [2017] introduce a lexical simplification tool for Italian that supports phrases instead of single tokens, and create a benchmark for Italian lexical simplification.

Tonelli et al. [2016] leverage the Italian Wikipedia edit history to create a simplification corpus. This is similar to previous work in English [Woodsend and Lapata, 2011, Botha et al., 2018]. They select Wikipedia edits marked as "simplified" and further annotate them to identify the type of simplification perform. After manually filtering out bad simplifications, the remaining 345 sentence pairs are gathered to form the SIMPITIKI corpus.

French Brouwers et al. [2014] propose a rule-based syntactic simplification method designed after analyzing two corpora of differing complexity. These rules are applied on the syntax tree of the original sentence and then an Integer Linear Programming algorithm selects the best transformations to be applied.

A corpus of aligned complex-simple texts were proposed recently in the ALECTOR corpus [Gala et al., 2020]. ALECTOR is a collection of 79 tales, stories, and scientific texts that were simplified at the document-level. These documents were extracted from French pupils textbooks.

Japanese Goto et al. [2015] release a corpus of news articles associated with their simplified versions produced by teachers of Japanese as a second language. The dataset is composed of 10,651 automatically aligned and 2735 manually aligned sentence pairs that can be used for evaluation.

Other approaches have used unsupervised MT to train Japanese Sentence Simplification

systems [Katsuta and Yamamoto, 2019], we detail such methods in Section 4.

Multilingual Methods Some works have proposed methods working in multiple languages. Part of the SIMPATICO project, Scarton et al. [2017] introduce MUSST, a multilingual rule-based syntactic simplification tool in English, Italian, and Spanish. MUSST identifies and implements common simplification rules in the 3 languages such as splitting conjoint clauses, relative clauses and appositive phrases, and changing sentences from passive to active voice.

3.3 Other Related Text Rewriting Tasks

Sentence Simplification has many similarities but also stark differences with other text rewriting tasks. In this section we give an overview of similar tasks and how they differ from Sentence Simplification

3.3.1 Subtasks of Sentence Simplification

Lexical Simplification Replacing complicated words with simpler ones is core to text simplification and can be isolated from other types of operations that one would find in a fully-fledged Sentence Simplification system. Lexical simplification is usually conducted in two stages: first complex words that need simplification have to be located, this is Complex Word Identification [Paetzold and Specia, 2016a], and then for a given complex word, a simpler synonym has to be produced. Complex word identification can either be performed by thresholding a lexical complexity measure [Bott et al., 2012], using a lexicon of complex words [Watanabe et al., 2009], or that evaluate potential simplifications for each word and discard the simplification if it does not make the overall sentence simpler. For the second step of associating simpler synonyms to complex words, Yatskar et al. [2010] and Biran et al. [2011] compared words from EW and SEW to constitute a set of simplification rules. In a similar direction, Pavlick et al. [2015] have used the large paraphrase database PPDB to identify paraphrases which are lexical simplifications, creating Simple PPDB. These lexical simplification components have been successfully combined with syntactic simplification

methods for general Sentence Simplification [Zhu et al., 2010, Coster and Kauchak, 2011b, Kauchak, 2013], or integrated in deep neural networks [Zhao et al., 2018].

Sentence Splitting Narayan et al. [2017] introduce the Split-and-Rephrase task, with the goal of learning and evaluating the sentence splitting operation. Sentence splitting is a core component of text simplification that is often left out by automatic systems and evaluation benchmarks such as TURKCORPUS. To this end, the authors introduce the WEBSPLIT *synthetic* dataset of 1M samples, each sample is a complex sentence associated with multiple shorter sentences. Aharoni and Goldberg [2018] improve the splitting between training set and validation/test sets of the dataset. Indeed, some simple sentences appeared both in the training and validation/test sets due to overlap between underlying entities that were used to generate the sentences. However the synthetic nature of the dataset produced some unnatural linguistic expressions over only a small vocabulary [Botha et al., 2018]. This is why Botha et al. [2018] introduce WIKISPLIT, a sentence splitting dataset created using the Wikipedia edit history composed of 1M samples. Niklaus et al. [2019] improve even further by introducing the concept of *minimality*, where each complex sentence should be broken down in a set of minimal propositions. They observe that WIKISPLIT examples are always composed of 1 single sentence split per complex sentence, resulting in sometimes too long simple sentences that could be split even further. To this end MINWIKISPLIT automatically splits long simple sentences from WIKISPLIT using hand-written transformation rules. Sentence splitting evaluation can be performed using the SAMSA metric [Sulem et al., 2018a] on the HSPLIT human evaluation dataset [Sulem et al., 2018b].

In our work we try to integrate sentence splitting operations in our sentence simplification systems by proposing a new evaluation dataset encompassing both typical sentence simplification operations such as lexical simplification and compression, but also sentence splitting (Chapter 7). We also propose to explicitly model the sentence splitting aspect by conditioning simplification models on syntactic complexity controllable generation mechanisms (Chapter 9).

Sentence Compression Sentence Compression consists in shortening an input sentence while keeping its general meaning. It is also one of the core component of Sentence Simplification but it also constitute a standalone task. It was first proposed for summarization purposes by [Jing, 2000], where the goal was to find phrases that could be removed from the source sentence to make it more concise. This formulation of finding which words or phrases could be removed was explored using dynamic programming using heuristics [Turner and Charniak, 2005], and using Integer Linear Programming with various constraints carefully designed for sentence compression [Clarke and Lapata, 2008]. However, only considering word deletion does not fully encompass the variety of rewrite operations that humans would perform. Abstractive sentence compression additionally consider other operations such as substitution, reordering, or insertion thus making the task even more similar to Sentence Simplification. [Cohn and Lapata, 2013] propose a new corpus for abstractive sentence compression and use tree transduction approach to abstractive sentence compression. Latest approaches to sentence compression use neural sequence-to-sequence models. For instance, sentence compression can be achieved using length control by feeding the network a length countdown scalar [Fevry and Phang, 2018]. Length control has also been conducted by using a length vector and multilingual pivoting to overcome the lack of training data [Mallinson et al., 2018]. Sentence Simplification is very similar to sentence compression in the sense that it often reduces the length of input sentences, but it also includes additional operations such as lexical simplification or sentence splitting. The generated text can however sometimes be longer when sentence splitting or explanation are involved. In Chapter 9, we augment length control ideas for the task of Sentence Simplification to achieve controllable Sentence Simplification systems.

3.3.2 Other Tasks

Machine Translation Most methods used in Machine Translation (MT) are also adapted to Sentence Simplification such as Statistical MT [Koehn et al., 2007] adapted in [Wubben et al., 2012, Xu et al., 2016], sequence-to-sequence models [Sutskever et al., 2014, Vaswani et al., 2017] used in [Nisioi et al., 2017, Zhang and Lapata, 2017], or unsupervised MT

[Lample et al., 2018a, Lample and Conneau, 2019, Artetxe et al., 2018]. Indeed, both tasks are sequence to sequence rewriting task where the output has to express the original meaning, be fluent, but is in another language or level of complexity. Sentence Simplification has the particularity of being a monolingual task, therefore the source can be used for evaluation such as with SARI [Xu et al., 2016], unlike BLEU [Papineni et al., 2002] for MT which only consider the target language. This makes it also harder to learn good simplification models because keeping the source completely unchanged is a strong baseline that models tend to fall into unless specific inductive biases are baked into the models [Wubben et al., 2012].

Paraphrasing On the other hand, paraphrasing is also a monolingual text rewriting task that aims at expressing the original meaning but with another wording. Sentence Simplification can be considered as a specific type of paraphrasing where the paraphrase has to be easier to read and understand. Similar to Sentence Simplification, parallel paraphrases are hard to find in large quantities. Previous research has aligned sentences from various parallel corpora [Barzilay and Lee, 2003] with multiple objective functions [Liu et al., 2020a]. A large body of work has used bilingual pivoting to create paraphrase data. Bilingual pivoting consists in using a bilingual parallel MT dataset, say English-French, and translating the French side back to English. This translated English sentence forms a paraphrase of the original English sentence. This method has been used to create large databases of word-level paraphrases [Pavlick et al., 2015], lexical simplifications [Pavlick and Callison-Burch, 2016, Kriz et al., 2018], or sentence-level paraphrase corpora [Wieting and Gimpel, 2018].

Summarization Sentence Simplification also shares similarities with summarization. While Sentence Simplification is only studied at the sentence-level, summarization operates at the document-level. Summaries are often easier to read and use shorter sentences, but this is not a requirement. Contrary to Sentence Simplification, summarization discards a good portion of the original meaning and details, whereas Sentence Simplification generally keeps the most of the original information. However when research transition to fully-fledged document-level Text Simplification, summarization will certainly play an important role in the overall simplification process. As an example, FALC (Chapter 2) combines

heavy text summarization with very simple wording of the output text. Two stream of summarization methods have been proposed: extractive and abstractive summarization. Extractive summarization consists in selecting and extracting a few sentences as is from the original document to form a summary [Kupiec et al., 1995, Paice, 1990, Saggion and Poibeau, 2013]. Extractive methods are easier to implement and reach strong performance. For instance just extracting the first few sentences of a news article usually constitutes a strong baseline. Abstractive summarization on the other hand uses text generation methods to generate brand new sentences that will form the summary [Rush et al., 2015, Chopra et al., 2016, Nallapati et al., 2016]. Abstractive summarization however often suffers from factual consistency errors [Kryscinski et al., 2020] which also happens in Sentence Simplification. Sentence Simplification methods are more similar to abstractive summarization methods, because they need to rewrite at least some of the input text, since only extracting original content cannot handle all simplification operations. Still, most tokens are usually copied from the source, which lead recent Sentence Simplification approaches [Guo et al., 2018] to combine it with what the hybrid extractive-abstractive Pointer-copy model introduced for summarization [See et al., 2017].

Chapter 4

Unsupervised Simplification

Simplification data is hard to find in English, and even more so in other languages. Furthermore, in English, available data such as EW-SEW can be noisy with low quality alignments where sentences are not related or the simple side is not simpler than the source [Xu et al., 2015]. Multiple works have successfully proposed Sentence Simplification systems that do not need labelled simplification data.

4.1 Method inspired from Unsupervised Machine Translation

Various approaches have reused methods from Unsupervised Machine Translation (MT) [Lample et al., 2018a,b, Artetxe et al., 2018] to perform unsupervised Sentence Simplification. In the unsupervised MT setting, we want to learn a model that translates from one language to the other given two distinct monolingual corpora, one in each language. This is why the prevailing approach to unsupervised Sentence Simplification first splits a monolingual corpora into sets of complex and simple sentences using readability metrics. In unsupervised MT models are often initialized using dictionary alignment, which is not necessary in Sentence Simplification because it is a monolingual task, hence most of the vocabulary is naturally shared. [Surya et al., 2019] split EW into two monolingual sets using the Flesch Reading Ease (FRE) readability metric [Flesch, 1948] and use auto-encoding

to train an Sentence Simplification model in an unsupervised manner. A shared encoder and two decoders (one for complex text and one for simple text) are trained using a reconstruction loss and an adversarial loss. Given a sentence from either the complex corpus or simple corpus, the encoder creates a latent representation, which the associated decoder needs to convert back into the original text (reconstruction loss). The authors further train the model with denoising by perturbing the input sentence with word shuffling for instance. In order to make the latent representation shared for complex and simple original sentences, a discriminator is trained in an adversarial manner to distinguish latent representations of complex sentences and latent representations of simple sentences. The model is trained to confuse the discriminator and thus create shared latent representations. At test time, the shared encoder encodes a complex sentence, and the simple decoder generates a simplification by using the shared latent representation. Zhao et al. [2020] reuse the same split of EW into two disjoint sets, but instead train their model with back-translation. Back-translation consists in creating synthetic aligned pairs by training a "complexification" model that will take a simple sentence as input and generate an associated complex sentence. This generated complex sentence is then fed as input to the simplification model that learns to predict the original simple sentence. Denoising is also used for better performance. The authors also optimize simplification specific rewards related to fluency, relevance, and complexity using reinforcement learning (policy gradients). Aprosio et al. [2019] also use back-translation to train models in Italian and Spanish with a very small high quality labelled dataset and a large unaligned simple dataset for semi-supervised Sentence Simplification.

4.2 Other Methods

Kajiwara and Komachi [2018] emulate the alignment methods traditionally used with EW-SEW, but without using SEW. They split EW in two disjoint sets using FRE and then aligned similar sentences from the complex and the simple set using alignment between word embeddings. This pseudo-corpus was then used to train Sentence Simplification systems with good performance in English. Other unsupervised approaches iteratively edit the sentence until a certain criterion is reached [Kumar et al., 2020]. They first generate

various candidate simplification operations on the parse tree of the input sentence and then select the operation that scored the best using quality estimation features of fluency, meaning preservation, and simplicity. Given the availability of labelled Sentence Simplification data in English and MT data from English to various languages,Mallinson et al. [2020] train an encoder-decoder model at multi-tasking between Sentence Simplification and MT. Using task-specific layers and language-specific layers, they can at test time perform cross-lingual simplification without using any cross-lingual simplification labelled data. machine translation data to adapt English simplification models for other languages [Mallinson et al., 2020].

4.3 Discussion

The performance of unsupervised methods are generally below their supervised counterparts. In Chapter 10, we propose an unsupervised method that bridges the performance gap with supervised method and removes the need for deciding in advance how complex and simple sentences should be separated, but instead trains directly on paraphrases mined from raw corpora.

Part II

Evaluating Sentence Simplification Systems

Chapter 5

Evaluating a Simplification when no References are Available

In this chapter and the following, we study how Sentence Simplification is evaluated, highlight shortcomings of current evaluation methods and propose new contributions. We first study evaluation metrics when no reference is available in this chapter. Then, in Chapter 6, we will propose a new library regrouping traditional evaluation metrics and quality estimation features (EASSE). In Chapter 7, we propose a new evaluation dataset for Sentence Simplification with more varied rewriting operations. Finally in Chapter 8 we highlight shortcomings of current evaluation methods and adapt recent neural evaluation metrics to the task of Sentence Simplification.

One of the main challenges in Sentence Simplification is finding an adequate automatic evaluation metric, which is necessary to avoid the time-consuming human evaluation. Any Sentence Simplification evaluation metric should take into account three properties expected from the output of a Sentence Simplification system, namely:

- **Grammaticality**: how grammatically correct is the Sentence Simplification system output?

- **Meaning preservation**: how well is the meaning of the source sentence preserved in the Sentence Simplification system output?

- **Simplicity**: how simple is the Sentence Simplification system output?[1]

As previously mentioned and as in the majority of research, we limit the scope of our work to a sentence-level problem, whereby one sentence is transformed into a simpler version containing one or more sentences.

Sentence Simplification, seen as a sentence-level problem, is often viewed as a monolingual variant of (sentence-level) MT. The standard approach to automatic Sentence Simplification evaluation is therefore to view the task as a translation problem and to use machine translation (MT) evaluation metrics such as BLEU [Papineni et al., 2002]. However, MT evaluation metrics rely on the existence of parallel corpora of source sentences and manually produced reference translations, which are available on a large scale for many language pairs [Tiedemann, 2012a]. Sentence Simplification datasets are less numerous and smaller. Moreover, they are often automatically extracted from comparable corpora rather than strictly parallel corpora, which results in noisier reference data. For example, the PWKP dataset [Zhu et al., 2010] consists of 100,000 sentences from the English Wikipedia automatically aligned with sentences from the Simple English Wikipedia based on term-based similarity metrics. It has been shown by Xu et al. [2015] that many of PWKP's "simplified" sentences are in fact not simpler or even not related to their corresponding source sentence. Even if better quality corpora such as Newsela do exist [Xu et al., 2015], they are costly to create, often of limited size, and not necessarily open-access.

This creates a challenge for the use of reference-based MT metrics for Sentence Simplification evaluation. However, Sentence Simplification has the advantage of being a monolingual translation-like task, the source being in the same language as the output. This allows for new, non-conventional ways to use MT evaluation metrics, namely by using them to compare the output of a Sentence Simplification system with the source sentence, thus avoiding the need for reference data. However, such an evaluation method can only capture at most two of the three above-mentioned dimensions, namely meaning preservation and, to a lesser extent, grammaticality.

Previous works on reference-less Sentence Simplification evaluation include Štajner

[1]There is no unique way to define the notion of *simplicity* in this context. Previous works often rely on the intuition of human annotators to evaluate the level of simplicity of a Sentence Simplification system output.

et al. [2014], who compare the behavior of six different MT metrics when used between the source sentence and the corresponding simplified output. They evaluate these metrics with respect to meaning preservation and grammaticality. We extend their work in two directions. Firstly, we extend the comparison to include the degree of simplicity achieved by the system. Secondly, we compare additional features, including those used by Štajner et al. [2016a], both individually, as elementary metrics, and within multi-feature metrics. To our knowledge, no previous work has provided as thorough a comparison across such a wide range and combination of features for the reference-less evaluation of Sentence Simplification.[2]

First we review available text simplification evaluation methods and traditional quality estimation features. We then present the QATS shared task and the associated dataset, which we use for our experiments. Finally we compare all methods in a reference-less setting and analyze the results.

5.1 Related Work

5.1.1 Existing evaluation methods

Using MT metrics to compare the output and a reference Sentence Simplification can be considered as a monolingual translation task. As a result, MT metrics such as BLEU [Papineni et al., 2002], which compare the output of an MT system to a reference translation, have been extensively used for Sentence Simplification [Narayan and Gardent, 2014, Štajner et al., 2015a, Xu et al., 2016]. Other successful MT metrics include TER [Snover et al., 2009], ROUGE [Lin, 2004] and METEOR [Banerjee and Lavie, 2005], but they have not gained much traction in the Sentence Simplification literature.

These metrics rely on good quality references, something which is often not available in Sentence Simplification, as discussed by Xu et al. [2015]. Moreover, Štajner et al. [2015a] and Sulem et al. [2018b] showed that using BLEU to compare the system output with a reference is not a good way to perform Sentence Simplification evaluation, even when good

[2]This chapter is an adapted version of [Martin et al., 2018].

quality references are available. This is especially true when the Sentence Simplification system produces more than one sentence for a single source sentence.

Using MT metrics to compare the output and the source sentence As mentioned in the Introduction, the fact that Sentence Simplification is a monolingual task means that MT metrics can also be used to compare a system output with its corresponding source sentence, thus avoiding the need for reference data. Following this idea, Štajner et al. [2014] found encouraging correlations between 6 widely used MT metrics and human assessments of grammaticality and meaning preservation. However MT metrics are not relevant for the evaluation of simplicity, which is why they did not take this dimension into account. Xu et al. [2016] also explored the idea of comparing the Sentence Simplification system output with its corresponding source sentence, but their metric, SARI, also requires to compare the output with a reference. In fact, this metric is designed to take advantage of more than one reference. It can be applied when only one reference is available for each source sentence, but its results are better when multiple references are available.

Attempts to perform Quality Estimation on the output of Sentence Simplification systems, without using references, include the 2016 Quality Assessment for Text Simplification (QATS) shared task [Štajner et al., 2016b], to which we shall come back in section 5.2. Sulem et al. [2018a] introduce another approach, named SAMSA. The idea is to evaluate the structural simplicity of a Sentence Simplification system output given the corresponding source sentence. SAMSA is maximized when the simplified text is a sequence of short and simple sentences, each accounting for one semantic event in the original sentence. It relies on an in-depth analysis of the source sentence and the corresponding output, based on a semantic parser and a word aligner. A drawback of this approach is that good quality semantic parsers are only available for a handful of languages. The intuition that sentence splitting is an important sub-task for producing simplified text motivated Narayan et al. [2017] to organize the *Split and Rephrase* shared task, which was dedicated to this problem.

Other metrics One can also estimate the quality of a Sentence Simplification system output based on simple features extracted from it.

For instance, the QuEst framework for quality estimation in MT gives a number of useful baseline features for evaluating an output sentence [Specia et al., 2013]. These features range from simple statistics, such as the number of words in the sentence, to more sophisticated features, such as the probability of the sentence according to a language model. Several teams who participated in the QATS shared task used metrics based on this framework, namely SMH [Štajner et al., 2016a], UoLGP [Rios and Sharoff, 2015] and UoW [Béchara et al., 2015].

Readability metrics such as Flesch-Kincaid Grade Level (FKGL) and Flesch Reading Ease (FRE) [Kincaid et al., 1975] have been extensively used for evaluating simplicity. These two metrics, which were shown experimentally to give good results, are linear combinations of the number of words per sentence and the number of syllables per word, using carefully adjusted weights. See Chapter 2 for more details.

5.2 Benchmarking Existing Metrics

Our goal is to compare a large number of ways to perform Sentence Simplification evaluation without a reference. To this end, we use the dataset provided in the QATS shared task. We first compare the behavior of elementary metrics, which range from commonly used metrics such as BLEU to basic metrics based on a single low-level feature such as sentence length. We then compare the effect of aggregating these elementary metrics into more complex ones and compare our results with the state of the art, based on the QATS shared task data and results.

5.2.1 The QATS shared task

The data from the QATS shared task [Štajner et al., 2016b] consists of a collection of 631 pairs of english sentences composed of a source sentence extracted from an online corpus and a simplified version thereof, which can contain one or more sentences. This collection is split into a training set (505 sentence pairs) and a test set (126 sentence pairs). Simplified versions were produced automatically using one of several Sentence Simplification systems trained by the shared task organizers. Human annotators labelled each sentence pair using

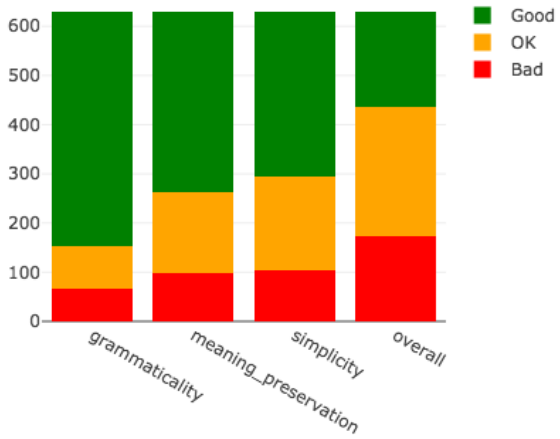


Figure 5.1: Label repartition on the QATS Shared task

one of the three labels *Good*, *OK* and *Bad* on each of the three dimensions: grammaticality, meaning preservation and simplicity[3]. An overall quality label was then automatically assigned to each sentence pair based on its three manually assigned labels using a method detailed in [Štajner et al., 2016b]. Distribution of the labels and examples are presented in Figure 5.1 and Table 5.1.

The goal of the shared task is, for each sentence in the test set, to either produce a label (*Good*, *OK*, *Bad*) or a raw score estimating the overall quality of the simplification for each of the three dimensions. Raw score predictions are evaluated using the Pearson correlation with the ground truth labels, while actual label prediction are evaluated using the weighted F1-score. The shared task is described in further details on the QATS website[4].

5.2.2 Considered Features

In our experiments, we compared about 60 elementary metrics. BLEU and FKGL are detailed in Chapter 2.

- MT metrics
  - BLEU, ROUGE, METEOR, TERp

[3]We were not able to find detailed information about the annotation process. In particular, we do not know whether each sentence was annotated only once or whether multiple annotations were produced, followed by an adjudication step.

[4]`http://qats2016.github.io/shared.html`

Version	Sentence	Aspect				Modification
		G	M	S	O	
Original	All three were arrested in the Toome area **and** have been taken to the Serious Crime Suite at Antrim police station.	good	good	good	good	syntactic
Simple	All three were arrested in the Toome area. **All three** have been taken to the Serious Crime Suite at Antrim police station.					
Original	For years the former Bosnia Serb army commander Ratko Mladic had evaded capture **and was one of the world's most wanted men, but his time on the run finally ended last year when he was arrested near Belgrade.**	good	bad	ok	bad	content reduction
Simple	For years the former Bosnia Serb army commander Ratko Mladic had evaded capture.					
Original	Madrid was occupied by French troops during the Napoleonic Wars, and Napoleon's brother Joseph was **installed** on the throne.	good	good	good	good	lexical
Simple	Madrid was occupied by French troops during the Napoleonic Wars, and Napoleon's brother Joseph was **put** on the throne.					
Original	Keeping articles with potential **encourages** editors, especially unregistered users, to be bold and improve the article to allow it to evolve over time.	bad	bad	ok	bad	dropping
Simple	Keeping articles with potential editors, especially unregistered users, to be bold and improve the article to allow it to evolve over time.					

Table 5.1: Examples from the training dataset of QATS. Differences between the original and the simplified version are presented in bold. This table is adapted from Štajner et al. [2016b].

  - Variants of BLEU: BLEU_1gram, BLEU_2gram, BLEU_3gram, BLEU_4gram and seven smoothing methods[5] from NLTK [Bird and Loper, 2004].
  - Intermediate components of TERp inspired by [Štajner et al., 2016a]: e.g. number of insertions, deletions, shifts...

- Readability metrics and other sentence-level features: FKGL and FRE, numbers of words, characters, syllables...

- Metrics based on the baseline QUEST features (17 features) [Specia et al., 2013], such as statistics on the number of words, word lengths, language model probability and n-gram frequency.

- Metrics based on other features: frequency table position, concreteness as extracted from Brysbaert et al.'s 2014 list, language model probability of words using a convolutional sequence to sequence model from [Gehring et al., 2017], comparison methods using pre-trained fastText word embeddings [Mikolov et al., 2018] or Skip-thought sentence embeddings [Kiros et al., 2015].

Table 5.2 lists 30 of the elementary metrics that we compared, which are those that we found to correlate the most with human judgments on one or more of the three dimensions (grammaticality, meaning preservation, simplicity).

5.2.3 Experimental setup

Evaluation of elementary metrics We rank all features by comparing their behavior with human judgments on the training set. We first compute for each elementary metric the Pearson correlation between its results and the manually assigned labels for each of the three dimensions. We then rank our elementary metrics according to the absolute value of the Pearson correlation.[6]

[5] `https://www.nltk.org/api/nltk.translate.html#nltk.translate.bleu_score.SmoothingFunction`

[6] The code is available on Github at `https://github.com/facebookresearch/text-simplification-evaluation`

Short name	Description
NBSourcePunct	Number of punctuation tokens in source (QUEST)
NBSourceWords	Number of source words (QUEST)
NBOutputPunct	Number of punctuation tokens in output (QUEST)
TypeTokenRatio	Type token ratio (QUEST)
TERp_Del	Number of deletions (TERp component)
TERp_NumEr	Number of total errors (TERpt component)
TERp_Sub	Number of substitutions (TERp component)
TERp	TERp MT metric
BLEU_1gram	BLEU MT metric with unigrams only
BLEU_2gram	BLEU MT metric up to bigrams
BLEU_3gram	BLEU MT metric up to trigrams
BLEU_4gram	BLEU MT metric up to 4-grams
METEOR	METEOR MT metric
ROUGE	ROUGE summarization metric
BLEUSmoothed	BLEU MT metric with smoothing (method 7 from nltk)
AvgCosineSim	Cosine similarity between source and output pre-trained word embeddings
NBOutputChars	Number of characters in the output
NBOutputCharsPerSent	Average number of characters per sentence in the output
NBOutputSyllables	Number of syllables in the output
NBOutputSyllablesPerSent	Average number of syllables per sentence in the output
NBOutputWords	Number of words in the output
NBOutputWordsPerSent	Average number of words per sentence in the output
AvgLMProbsOutput	Average log-probabilities of output words (Language Model)
MinLMProbsOutput	Minimum log-probability of output words (Language Model)
MaxPosInFreqTable	Maximum position of output words in the frequency table
AvgConcreteness	Average word concreteness Brysbaert et al.'s 2014 concreteness list
OutputFKGL	Flesch-Kincaid Grade Level
OutputFRE	Flesch Reading Ease
WordsInCommon	Percentage of words in common between source and Output

Table 5.2: Brief description of 30 of our most relevant elementary metrics

Training and evaluation of a combined metric We use our elementary metrics as features to train classifiers on the training set, and evaluate their performance on the test set. We therefore scale them and reduce the dimensionality with a 25-component PCA[7], then train several regression algorithms[8] and classification algorithms[9] using scikit-learn [Pedregosa et al., 2011]. For each dimension, we keep the two models performing best on the test set and add them in the leaderboard of the QATS shared task (Table 5.4), naming them with the name of the regression algorithm they were built with.

[7]We used PCA instead of feature selection because it performed better on the validation set. The number of component was tuned on the validation set as well.

[8]Regressors: Linear regression, Lasso, Ridge, Linear SVR (SVM regressor), Adaboost regressor, Gradient boosting regressor and Random forest regressor.

[9]Classifiers: Logistic regression, MLP classifier (with L2 penalty), SVC (linear SVM classifier), k-nearest neighbors classifier (k=3), Adaboost classifier, Gradient boosting classifier and Random forest classifier.

Grammaticality			Meaning Preservation			Simplicity		
Short name	Train ↓	Test	Short name	Train ↓	Test	Short name	Train ↓	Test
Best QATS team		0.48	*Best QATS team*		0.59	*Best QATS team*		0.38
METEOR	0.36	0.39	BLEUSmoothed	0.59	0.52	NBOutputCharsPerSent	-0.52	-0.45
BLEUSmoothed	0.33	0.34	BLEU_3gram	0.57	0.52	NBOutputSyllablesPerSent	-0.52	-0.49
BLEU_4gram	0.32	0.34	METEOR	0.57	0.58	NBOutputWordsPerSent	-0.51	-0.39
BLEU_3gram	0.31	0.34	BLEU_2gram	0.57	0.52	NBOutputChars	-0.48	-0.37
TERp_NumEr	-0.30	-0.31	BLEU_4gram	0.57	0.51	NBOutputWords	-0.47	-0.29
BLEU_2gram	0.30	0.34	WordsInCommon	0.55	0.50	NBOutputSyllables	-0.46	-0.42
TERp	-0.30	-0.32	BLEU_1gram	0.55	0.52	NBOutputPunt	-0.42	-0.31
ROUGE	0.29	0.29	ROUGE	0.55	0.47	NBSourceWords	-0.38	-0.21
AvgLMProbsOutput	0.28	0.34	TERp	-0.54	-0.48	outputFKGL	-0.36	-0.37
BLEU_1gram	0.27	0.33	TERp_NumEr	-0.53	-0.49	NBSourcePunct	-0.34	-0.18
WordsInCommon	0.27	0.30	TERp_Del	-0.50	-0.52	TypeTokenRatio	-0.22	-0.04
TERp_Del	-0.27	-0.35	AvgCosineSim	0.44	0.34	AvgConcreteness	0.21	0.32
NBSourceWords	-0.25	-0.07	AvgLMProbsOutput	0.39	0.36	MaxPosInFreqTable	-0.18	0.03
AvgCosineSim	0.23	0.25	AvgConcreteness	-0.28	-0.06	MinLMProbsOutput	0.17	0.15
MinLMProbsOutput	0.11	-0.07	NBSourceWords	-0.28	-0.13	OutputFRE	0.16	0.27

Table 5.3: Pearson correlation with human judgments of elementary metrics ranked by absolute value on training set (15 best metrics for each dimension).

5.3 Results

5.3.1 Comparing elementary metrics

Figure 5.3 ranks all elementary metrics given their absolute Pearson correlation on each of the three dimensions.

Grammaticality N-gram based MT metrics have the highest correlation with human grammaticality judgments. METEOR seems to be the best, probably because of its robustness to synonymy, followed by smoothed BLEU (BLEUSmoothed in 5.2). This indicates that relevant grammaticality information can be derived from the source sentence. We were expecting that information contained in a language model would help achieving better results (*AvgLMProbsOutput*), but MT metrics correlate better with human judgments. We deduce that the grammaticality information contained in the source is more specific and more helpful for evaluation than what is learned by the language model.

Meaning preservation It is not surprising that meaning preservation is best evaluated using MT metrics that compare the source sentence to the output sentence, with in particular smoothed BLEU, BLEU_3gram and METEOR. Very simple features such as the percentage

of words in common between source and output also rank high. Surprisingly, word embedding comparison methods do not perform as well for meaning preservation, even when using word alignment.

Simplicity Methods that give the best results are the most straightforward for assessing simplicity, namely word, character and syllable counts in the output, averaged over the number of output sentences. These simple features even outperform the traditional, more complex metrics FKGL and FRE. As could be expected, we find that metrics with the highest correlation to human simplicity judgments only take the output into account. Exceptions are the *NBSourceWords* and *NBSourcePunct* features. Indeed, if the source sentence has a lot of words and punctuation, and is therefore likely to be particularly complex, then the output will most likely be less simple as well. We also expected word concreteness ratings and position in the frequency table to be good indicators of simplicity, but it does not seem to be the case here. Structural simplicity might simply be more important than such more sophisticated components of the human intuition of simple text.

Discussion Even if counting the number of words or comparing n-grams are good proxies for the simplification quality, they are still very superficial features and might miss some deeper and more complex information. Moreover the fact that grammaticality and meaning preservation are best evaluated using n-gram-based comparison metrics might bias the Sentence Simplification models towards copying the source sentence and applying fewer modifications.

Syntactic parsing or language modeling might capture more insightful grammatical information and allow for more flexibility in the simplification model. Regarding meaning preservation, semantic analysis or paraphrase detection models would also be good candidates for a deeper analysis.

Warning note We should be careful when interpreting these results as the QATS dataset is relatively small. We compute confidence intervals on our results, and find them to be non-negligible, yet without putting our general observations into question. For instance, METEOR, which performs best on grammaticality, has a 95% confidence interval of $0.36 \pm$

0.08 on the training set. These results are therefore preliminary and should be validated on other datasets.

5.3.2 Combination of all features with trained models

We also combine all elementary metrics and train an evaluation models for each of the three dimensions. Table 5.4a presents our two best regressors in validation for each of the dimensions and Table 5.4b for classifiers.

Pearson correlation for regressors (raw scoring) Combining the features does not bring a clear advantage over the elementary metrics METEOR and NBOutputSyllablesPerSent. Indeed our best models score respectively on grammaticality, meaning preservation and simplicity: 0.33 (Lasso), 0.58 (Ridge) and 0.49 (Ridge) versus 0.39 (METEOR), 0.58 (METEOR) and 0.49 (NBOutputSyllablesPerSent).

It is surprising to us that the aggregation of multiple elementary features would score worse than the features themselves. However, we observe a strong discrepancy between the scores obtained on the train and test set, as illustrated by Table 5.3. We also observed very large confidence intervals in terms of Pearson correlation. For instance our lasso model scores 0.33 ± 0.17 on the test set for grammaticality. This should observe caution when interpreting Pearson scores on QATS.

F1-score for classifiers (assigning labels) On the classification task, our models seem to score best for meaning preservation, simplicity and overall, and third for grammaticality. This seems to confirm the importance of considering a large ensemble of elementary features including length-based metrics to evaluate simplicity.

5.4 Discussion

Finding accurate ways to evaluate Sentence Simplification without the need for reference data is a key challenge, both for exploring new approaches and for optimizing current models, in particular those relying on unsupervised, often MT-inspired models.

Grammaticality	Meaning Preservation	Simplicity	Overall
0.482 OSVCML1	0.588 IIT-Meteor	0.487 **Ridge**	0.423 **Ridge**
0.384 METEOR	0.585 OSVCML	0.456 **LinearSVR**	0.423 **LinearRegression**
0.344 BLEU	0.575 **Ridge**	0.382 OSVCML1	0.343 OSVCML2
0.340 OSVCML	0.573 OSVCML2	0.376 OSVCML2	0.334 OSVCML
0.327 **Lasso**	0.555 **Lasso**	0.339 OSVCML	0.232 SimpleNets-RNN2
0.323 TER	0.533 BLEU	0.320 SimpleNets-MLP	0.230 OSVCML1
0.308 SimpleNets-MLP	0.527 METEOR	0.307 SimpleNets-RNN3	0.205 UoLGP-emb
0.308 WER	0.513 TER	0.240 SimpleNets-RNN2	0.198 SimpleNets-MLP
0.256 UoLGP-emb	0.495 WER	0.123 UoLGP-combo	0.196 METEOR
0.256 UoLGP-combo	0.482 OSVCML1	0.120 UoLGP-emb	0.189 UoLGP-combo
0.208 UoLGP-quest	0.465 SimpleNets-MLP	0.086 UoLGP-quest	0.144 UoLGP-quest
0.118 **GradientBoostingRegressor**	0.285 UoLGP-quest	0.052 IIT-S	0.130 TER
0.064 SimpleNets-RNN3	0.262 SimpleNets-RNN2	-0.169 METEOR	0.112 SimpleNets-RNN3
0.056 SimpleNets-RNN2	0.262 SimpleNets-RNN3	-0.242 TER	0.111 WER
	0.250 UoLGP-combo	-0.260 WER	0.107 BLEU
	0.188 UoLGP-emb	-0.267 BLEU	

(a) Pearson correlation for regressors (raw scoring)

Grammaticality	Meaning Preservation	Simplicity	Overall
71.84 SMH-RandForest	70.14 **SVC**	61.60 **SVC**	49.61 **LogisticRegression**
71.64 SMH-IBk	68.07 SMH-Logistic	56.95 **AdaBoostClassifier**	48.57 SMH-RandForest-b
70.43 **LogisticRegression**	65.60 MS-RandForest	56.42 SMH-RandForest-b	48.20 UoW
69.96 SMH-RandForest-b	64.40 SMH-RandForest	53.02 SMH-RandForest	47.54 SMH-Logistic
69.09 BLEU	63.74 TER	51.12 SMH-IBk	46.06 SimpleNets-RNN2
68.82 SimpleNets-MLP	63.54 SimpleNets-MLP	49.96 SimpleNets-RNN3	45.71 **AdaBoostClassifier**
68.36 TER	62.82 BLEU	49.81 SimpleNets-MLP	44.50 SMH-RandForest
67.60 **GradientBoosting**	62.72 MT-baseline	48.31 MT-baseline	40.94 METEOR
67.53 MS-RandForest	62.69 IIT-Meteor	47.84 MS-IBk-b	40.75 SimpleNets-RNN3
67.50 IIT-LM	61.71 MS-IBk-b	47.82 MS-RandForest	39.85 MS-RandForest
66.79 WER	61.50 MS-IBk	47.47 SimpleNets-RNN2	39.80 DeepIndiBow
66.75 MS-RandForest-b	60.38 **GradientBoosting**	43.46 IIT-S	39.30 IIT-Metrics
65.89 DeepIndiBow	60.12 METEOR	42.57 DeepIndiBow	38.27 MS-IBk
65.89 DeepBow	59.69 SMH-RandForest-b	40.92 UoW	38.16 MS-IBk-b
65.89 MT-baseline	59.06 WER	39.68 Majority-class	38.03 DeepBow
65.89 Majority-class	58.83 UoW	38.10 MS-IBk	37.49 MT-baseline
65.72 METEOR	51.29 SimpleNets-RNN2	35.58 DeepBow	34.08 TER
65.50 SimpleNets-RNN2	51.00 CLaC-RF	34.88 CLaC-RF-0.5	34.06 CLaC-0.5
65.11 SimpleNets-RNN3	46.64 SimpleNets-RNN3	34.66 CLaC-RF-0.6	33.69 SimpleNets-MLP
64.39 CLaC-RF-Perp	46.30 DeepBow	34.48 WER	33.04 IIT-Default
62.00 MS-IBk	42.53 DeepIndiBow	34.30 CLaC-RF-0.7	32.92 BLEU
46.32 UoW	42.51 Majority-class	33.52 TER	32.88 CLaC-0.7
		33.34 METEOR	32.20 CLaC-0.6
		33.00 BLEU	31.28 WER
			26.53 Majority-class

(b) Weighted F1 Score for classifiers (assign the label Good, OK or Bad)

Table 5.4: QATS leaderboard. Results in **bold** are our additions to the original leaderboard. We only select the two models that rank highest during cross-validation.

We explore multiple reference-less quality evaluation methods for automatic Sentence Simplification systems, based on data from the 2016 QATS shared task. We rely on the three key dimensions of the quality of a Sentence Simplification system: grammaticality, meaning preservation and simplicity.

Our results show that grammaticality and meaning preservation are best assessed using n-gram-based MT metrics evaluated between the output and the source sentence. In particular, METEOR and smoothed BLEU achieve the highest correlation with human judgments. These approaches even outperform metrics that make an extensive use of external data, such as language models. This shows that a lot of useful information can be obtained from the source sentence itself.

Regarding simplicity, we observe that counting the number of characters, syllables and words provides the best results. In other words, given the currently available metrics, the length of a sentence seems to remain the best available proxy for its simplicity. We reuse this finding in Chapter 9 to create controllable models conditioned on length.

However, given the small size of the QATS dataset and the high variance observed in our experiments, these results must be taken with a pinch of salt and will need to be confirmed on a larger dataset. Creating a larger annotated dataset as well as averaging multiple human annotations for each pair of sentences would help reducing the variance of the experiments and confirming our findings.

Finally, it remains to be understood how we can optimize the trade-off between grammaticality, meaning preservation and simplicity, in order to build the best possible comprehensive Sentence Simplification metric in terms of correlation with human judgments. Unsurprisingly, optimizing one of these dimensions often leads to lower results on other dimensions [Schwarzer and Kauchak, 2018]. For instance, the best way to guarantee grammaticality and meaning preservation is to leave the source sentence unchanged, thus resulting in no simplification at all. Improving Sentence Simplification systems will require better global Sentence Simplification evaluation metrics. This is especially true when considering that Sentence Simplification is in fact a multiply defined task, as there are many different ways of simplifying a text, depending on the different categories of people and applications at whom Sentence Simplification is aimed. In an attempt to solve this problem, we introduce

in Chapter 7 ASSET, a new Sentence Simplification benchmark featuring varied types of simplification operations.

Chapter 6

EASSE: A Tool for Evaluation Simplification Systems

In the previous chapter we explored how to estimate the quality of a generated simplification without using any references. However a few simplification datasets exists with gold references that can be used for single or multi-reference evaluation. It is common practice to use machine translation (MT) metrics (e.g. BLEU [Papineni et al., 2002]), simplicity metrics (e.g. SARI [Xu et al., 2016]), and readability metrics (e.g. FKGL [Kincaid et al., 1975]).

Most of these metrics are available in individual code repositories, with particular software requirements that sometimes differ even in programming language (e.g. corpus-level SARI is implemented in Java, whilst sentence-level SARI is available in both Java and Python). Other metrics (e.g. SAMSA [Sulem et al., 2018a]) suffer from insufficient documentation or require executing multiple scripts with hard-coded paths, which prevents researchers from using them.

We introduce EASSE (Easier Automatic Sentence Simplification Evaluation), a Python package that provides access to popular automatic metrics in Sentence Simplification evaluation and ready-to-use public datasets through a simple command-line interface.[1] With this tool, we make the following contributions: (1) we provide popular automatic metrics in a single software package, (2) we supplement these metrics with word-level transformation analysis and reference-less Quality Estimation (QE) features, (3) we provide straightforward

[1]This chapter is an adapted version of [Alva-Manchego et al., 2019a].

access to commonly used evaluation datasets, and (4) we generate a comprehensive HTML report for quantitative and qualitative evaluation of a Sentence Simplification system. We believe this package will facilitate evaluation and improve reproducibility of results in Sentence Simplification. EASSE is available at `https://github.com/feralvam/easse`.

6.1 Package Overview

6.1.1 Automatic Corpus-level Metrics

Although human judgements on grammaticality, meaning preservation and simplicity are considered the most reliable method for evaluating a Sentence Simplification system's output [Štajner et al., 2016b], it is common practice to use automatic metrics. They are useful for either assessing systems at development stage, to compare different architectures, for model selection, or as part of a training policy. EASSE implementation works as a wrapper for the most common evaluation metrics in Sentence Simplification. This section serves as a brief reminder and lays out implementation details for each metrics. We describe these evalution metrics in more details in Chapter 2.

BLEU is a precision-oriented metric that relies on the proportion of n-gram matches between a system's output and reference(s). Previous work [Xu et al., 2016] has shown that BLEU correlates fairly well with human judgements of grammaticality and meaning preservation. EASSE uses SACREBLEU [Post, 2018][2] to calculate BLEU. This package was designed to standardise the process by which BLEU is calculated: it only expects a detokenised system's output and the name of a test set. Furthermore, it ensures that the same pre-processing steps are used for the system's output and reference sentences.

SARI measures how the simplicity of a sentence was improved based on the words added, deleted and kept by a system. The metric compares the system's output to multiple simplification references and the original sentence. SARI has shown positive correlation

[2] `https://github.com/mjpost/sacreBLEU`

with human judgements of simplicity gain. We re-implement SARI's corpus-level version in Python (it was originally available in Java).
Although Xu et al. [2016] indicate that only precision should be considered for the deletion operation, we follow the Java implementation that uses F1 score for all operations in corpus-level SARI (see Chapter 2 for the exact formula).

SAMSA measures structural simplicity (i.e. sentence splitting). This is in contrast to SARI, which is designed to evaluate simplifications involving paraphrasing. EASSE refactors the original SAMSA implementation[3] with some modifications: (1) an internal call to the TUPA parser [Hershcovich et al., 2017], which generates the semantic annotations for each original sentence; (2) a modified version of the monolingual word aligner [Sultan et al., 2014] that is compatible with Python 3, and uses Stanford CoreNLP [Manning et al., 2014][4] through their official Python interface; and (3) a single function call to get a SAMSA score instead of running a series of scripts.

FKGL Readability metrics, such as Flesch-Kincaid Grade Level (FKGL), are commonly reported as measures of simplicity. They however only rely on average sentence lengths and number of syllables per word, so short sentences would get good scores even if they are ungrammatical, or do not preserve meaning [Wubben et al., 2012]. Therefore, these scores should be interpreted with caution. EASSE re-implements FKGL by porting publicly available scripts[5] to Python 3 and fixing some edge case inconsistencies (e.g. newlines incorrectly counted as words or bugs with memoization).

6.1.2 Word-level Analysis and QE Features

Word-level Transformation Analysis EASSE includes algorithms to determine which specific text transformations a Sentence Simplification system performs more effectively. This is done based on word-level alignment and analysis.

[3] https://github.com/eliorsulem/SAMSA
[4] https://stanfordnlp.github.io/stanfordnlp/corenlp_client.html
[5] https://github.com/mmautner/readability

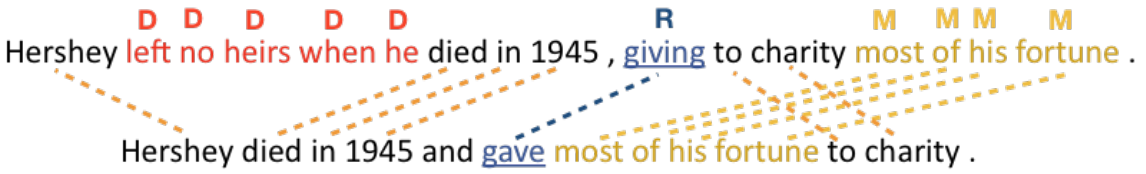


Figure 6.1: Example of automatic transformation annotations based on word alignments between an original (top) and a simplified (bottom) sentence. Unaligned words are DELETE. Words that are aligned to a different form are REPLACE. Aligned words without an explicit label are COPY. A word whose relative index in the original sentence changes in the simplified one is considered a MOVE.

Since there is no available simplification dataset with manual annotations of the transformations performed, we re-use the annotation algorithms from MASSAlign [Paetzold et al., 2017]. Given a pair of sentences (e.g. original and system's output), the algorithms use word alignments to identify deletions, movements, replacements and copies (see Fig. 6.1). This process is prone to some errors: when compared to manual labels produced by four annotators in 100 original-simplified pairs, the automatic algorithms achieved a micro-averaged F1 score of 0.61 [Alva-Manchego et al., 2017].

We generate two sets of automatic word-level annotations: (1) between the original sentences and their reference simplifications, and (2) between the original sentences and their automatic simplifications produced by a Sentence Simplification system. Considering (1) as reference labels, we calculate the F1 score of each transformation in (2) to estimate their correctness. When more than one reference simplification exists, we calculate the per-transformation F1 scores of the output against each reference, and then keep the highest one as the sentence-level score. The corpus-level scores are the average of sentence-level scores.

Quality Estimation Features Traditional automatic metrics used for Sentence Simplification rely on the existence and quality of references, and are often not enough to analyse the complex process of simplification. QE leverages both the source sentence and the output simplification to provide additional information on specific behaviours of simplification systems which are not reflected in metrics such as SARI. EASSE uses QE features from

Chapter 5[6]. The QE features currently available are: the compression ratio of the simplification with respect to its source sentence, its Levenshtein similarity, the average number of sentence splits performed by the system, the proportion of exact matches (i.e. original sentences left untouched), average proportion of added words, deleted words, and lexical complexity score[7].

6.1.3 Access to Test Datasets

EASSE provides access to three publicly available datasets for automatic Sentence Simplification evaluation (Table 6.1): PWKP [Zhu et al., 2010], TurkCorpus [Xu et al., 2016], and HSplit [Sulem et al., 2018b]. All of them consist of the data from the original datasets, which are sentences extracted from English Wikipedia (EW) articles. EASSE can also evaluate system's outputs in other custom datasets provided by the user.

PWKP Zhu et al. [2010] automatically aligned sentences in 65,133 EW articles to their corresponding versions in Simple EW (SEW). Since the latter is aimed at English learners, its articles are expected to contain fewer words and simpler grammar structures than those in their EW counterpart. The test set split of PWKP contains 100 sentences, with 1-to-1 and 1-to-N alignments (resp. 93 and 7 instances). The latter correspond to instances of sentence splitting. Since this dataset has only one reference for each original sentence, it is not ideal for calculating automatic metrics that rely on multiple references, such as SARI.

TurkCorpus Xu et al. [2016] asked crowdworkers to simplify 2,359 original sentences extracted from PWKP to collect multiple simplification references for each one. This dataset was then randomly split into tuning (2,000 instances) and test (359 instances) sets. The test set only contains 1-to-1 alignments, mostly with instances of paraphrasing and deletion. Each original sentence in TurkCorpus has 8 simplified references. As such, it is better suited for computing SARI and multi-reference BLEU scores.

[6] `https://github.com/facebookresearch/text-simplification-evaluation`

[7] The lexical complexity score of a simplified sentence is computed by taking the log-ranks of each word in the frequency table. The ranks are then aggregated by taking their third quartile.

Test Dataset	Instances	Alignment Type	References
PWKP	93	1-to-1	1
	7	1-to-N	1
TurkCorpus	359	1-to-1	8
HSplit	359	1-to-N	4

Table 6.1: Test datasets available in EASSE. An instance corresponds to a source sentence with one or more possible references. Each reference can be composed of one or more sentences.

HSplit Sulem et al. [2018b] recognized that existing EW-based datasets did not contain sufficient instances of sentence splitting. As such, they collected four reference simplifications of this transformation for all 359 original sentences in the TurkCorpus test set. Even though SAMSA's computation does not require access to references, this dataset can be used to compute an upper bound on the expected performance of Sentence Simplification systems that model this type of structural simplification.

6.1.4 HTML Report Generation

EASSE wraps all the aforementioned analyses in a simple comprehensive HTML report that can be generated with a single command. This report compares the system's output with human reference(s) using simplification metrics and QE features. It also plots the distribution of compression ratios or Levenshtein similarities between sources and simplifications over the test set. Moreover, the analysis is broken down by source sentence length in order to get insights on how the model handles short source sentence versus longer source sentences, e.g. *does the model keep short sentences unmodified more often than long sentences?* This report further facilitates qualitative analysis of systems' outputs by displaying source sentences with their respective simplifications. The modifications performed by the model are highlighted for faster and easier analysis. For visualisation, EASSE samples simplification instances to cover different behaviours of the systems. Instances that are sampled include simplifications with sentence splitting, simplifications that significantly modify the source sentence, output sentences with a high compression rate, those that display lexical simplifications, among others. Each of these aspects is illustrated with 10 instances. An example of the report can

be viewed at `https://github.com/feralvam/easse/blob/master/demo/report.gif`.

6.2 Experiments

We collected publicly available outputs of several Sentence Simplification systems (Sec. 6.2.1) to evaluate their performance using the functionalities available in EASSE. In particular, we compare them using automatic metrics, and provide some insights on the reasoning behind their results (Sec. 6.2.2).

6.2.1 Sentence Simplification Systems

EASSE provides access to various Sentence Simplification systems' outputs that follow different approaches for the task. For instance, we include those that rely on phrase-based statistical MT, either by itself (e.g. PBSMT-R [Wubben et al., 2012]), or coupled with semantic analysis, (e.g. Hybrid [Narayan and Gardent, 2014]). We also include SBSMT-SARI [Xu et al., 2016], which relies on syntax-based statistical MT; DRESS-LS [Zhang and Lapata, 2017], a neural model using the standard encoder-decoder architecture with attention combined with reinforcement learning; and DMASS-DCSS [Zhao et al., 2018], the current state-of-the-art in the TurkCorpus, which is based on the Transformer architecture [Vaswani et al., 2017].

6.2.2 Comparison and Analysis of Scores

Automatic Metrics For illustration purposes, we compare systems' outputs using BLEU and SARI in TurkCorpus (with 8 manual simplification references), and SAMSA in HSplit. For calculating Reference values in Table 6.2, we sample one of the 8 human references for each instance as others have done [Zhang and Lapata, 2017].

When reporting SAMSA scores, we only use the first 70 sentences of TurkCorpus that also appear in HSplit.[8] This allows us to compute Reference scores for instances that contain

[8] At the time of this submission only a subset of 70 sentences had been released from HSplit. However, the full corpus will soon be available in EASSE.

structural simplifications (i.e. sentence splits). We calculate SAMSA scores for each of the four manual simplifications in HSplit, and choose the highest as an upper-bound Reference value. The results for all three metrics are shown in Table 6.2.

	TurkCorpus		HSplit
System	SARI	BLEU	SAMSA
Reference	49.88	97.41	54.00
PBSMT-R	38.56	**81.11**	**47.59**
Hybrid	31.40	48.97	46.68
SBSMT-SARI	39.96	73.08	41.41
DRESS-LS	37.27	80.12	45.94
DMASS-DCSS	**40.42**	73.29	35.45

Table 6.2: Comparison of systems' performance based on automatic metrics.

DMASS-DCSS is the state-of-the-art in TurkCorpus according to SARI. However, it gets the lowest SAMSA score, and the third to last BLEU score. PBSMT-R is the best in terms of these two metrics. Finally, across all metrics, the Reference stills gets the highest values, with significant differences from the top performing systems.

Word-level Transformations In order to better understand the previous results, we use the word-level annotations of text transformations (Table 6.3). Since SARI was design to evaluate mainly paraphrasing transformations, the fact that SBSMT-SARI is the best at performing replacements and second place in copying explains its high SARI score. DMASS-DCSS is second best in replacements, while PBSMT-R (which achieved the highest BLEU score) is the best at copying. Hybrid is the best at performing deletions, but is the worst at replacements, which SARI mainly measures. The origin of the TurkCorpus set itself could explain some of these observations. According to Xu et al. [2016], the annotators in TurkCorpus were instructed to mainly produce paraphrases, i.e. mostly replacements with virtually no deletions. As such, copying words is also a significant transformation, so systems that are good at performing it better mimic the characteristics of the human simplifications in this dataset.

System	Delete	Move	Replace	Copy
PBSMT-R	34.18	2.64	23.65	**93.50**
Hybrid	**49.46**	**7.37**	1.03	70.73
SBSMT-SARI	28.42	1.26	**37.21**	92.89
DRESS-LS	40.31	1.43	12.62	86.76
DMASS-DCSS	38.03	5.10	34.79	86.70

Table 6.3: Transformation-based performance of the sentence simplification systems in the TurkCorpus test set.

Quality Estimation Features Table 6.4 displays a subset of QE features that reveal other aspects of the simplification systems. For instance, the scores make it clear that Hybrid compresses the input way more than other systems (compression ratio of 0.57 vs. $\geq$0.78 for the other systems) but almost never adds new words (addition proportion of 0.01). This additional information explains the high Delete and low Replace performance of this system in Table 6.3. DRESS-LS keeps the source sentence unmodified 26% of the time, which does not show in the word-level analysis. This confirms that QE features are complementary to automatic metrics and word-level analysis.

System	Compression ratio	Exact matches	Additions proportion	Deletion proportion
PBSMT-R	0.95	0.1	0.1	0.11
Hybrid	**0.57**	0.03	0.01	**0.41**
SBSMT-SARI	0.94	0.11	**0.16**	0.13
DRESS-LS	0.78	**0.26**	0.04	0.26
DMASS-DCSS	0.89	0.05	0.15	0.21

Table 6.4: Quality estimation features, which give additional information on the output of different systems.

Report Figure 6.2 displays the quantitative part of the HTML report generated for the DMASS-DCSS system. The report compares the system to a reference human simplification. The "System vs. Reference" table and the two plots indicate that DMASS-DCSS closely matches different aspects of human simplifications, according to QE features. This contributes to explaining the high SARI score of the this system in Table 6.2.

6.3 Summary and Final Remarks

EASSE provides easy access to commonly used automatic metrics as well as to more detailed word-level transformation analysis and QE features which allows us to compare the quality of the generated outputs of different Sentence Simplification systems on public test datasets. We reported some experiments on the use of automatic metrics to obtain overall performance scores, followed by measurements of how effective the Sentence Simplification systems are at executing specific simplification transformations using word-level analysis and QE features. The former analysis provided insights about the simplification capabilities of each system, which help better explain the initial automatic scores.

In the next Chapters, we use EASSE as our de facto tool for Sentence Simplification evaluation.

EASSE report

Scores

System vs. Reference

	BLEU	SARI	FKGL	Compression ratio	Sentence splits	Levenshtein similarity	Exact matches	Additions proportion	Deletions proportion	Lexical complexity score
System output	73.29	40.42	7.66	0.89	0.98	0.82	0.05	0.15	0.21	8.44
Reference	68.02	41.31	8.22	0.85	1.01	0.8	0.08	0.13	0.26	8.59

By sentence length (characters)

	BLEU	SARI	FKGL	Compression ratio	Sentence splits	Levenshtein similarity	Exact matches	Additions proportion	Deletions proportion	Lexical complexity score
length=[33;77]	66.10	41.76	3.74	0.87	1.00	0.80	0.08	0.19	0.25	8.28
length=[77;102]	70.04	40.45	5.58	0.89	1.02	0.82	0.07	0.15	0.21	8.44
length=[102;131]	74.61	39.36	7.55	0.92	0.95	0.85	0.06	0.15	0.18	8.28
length=[131;167]	70.07	40.85	8.89	0.88	0.96	0.81	0.04	0.16	0.24	8.49
length=[167;354]	77.48	40.01	11.06	0.89	0.95	0.85	0.01	0.10	0.18	8.71

Plots

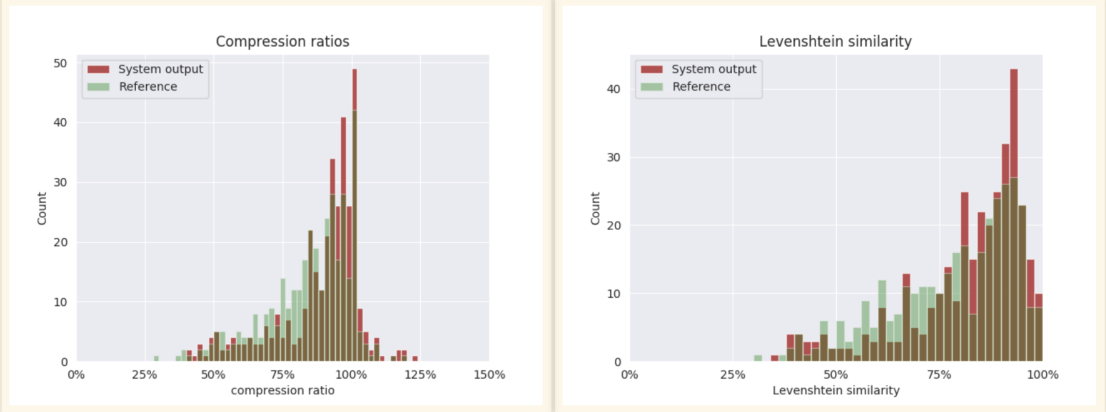


Figure 6.2: Overview of the HTML report for the DMASS-DCSS system (zoom in for more details).

Chapter 7

ASSET: A New Evaluation Dataset

In Chapter 5 we discussed how to estimate the quality of Sentence Simplification systems outputs without using any reference simplification. When reference simplifications are available, metrics such as SARI or BLEU are often used (Chapter 6). In this chapter we try to improve a complementary aspect of automatic evaluation: the gold reference simplifications.

In order to simplify a sentence, several rewriting transformations can be performed: replacing complex words/phrases with simpler synonyms (i.e. lexical paraphrasing), changing the syntactic structure of the sentence (e.g. splitting), or removing superfluous information that make the sentence more complicated [Petersen, 2007, Aluísio et al., 2008, Bott and Saggion, 2011b]. However, models for automatic Sentence Simplification are evaluated on datasets whose simplifications are not representative of this variety of transformations. For instance, TURKCORPUS [Xu et al., 2016], a standard dataset for assessment in Sentence Simplification, contains simplifications produced mostly by lexical paraphrasing, while reference simplifications in HSplit [Sulem et al., 2018b] focus on splitting sentences. The Newsela corpus [Xu et al., 2015] contains simplifications produced by professionals applying multiple rewriting transformations, but sentence alignments are automatically computed and thus imperfect, and its data can only be accessed after signing a restrictive public-sharing licence and cannot be redistributed, hampering reproducibility.

These limitations in evaluation data prevent studying models' capabilities to perform a broad range of simplification transformations. Even though most Sentence Simplification

models are trained on simplification instances displaying several text transformations (e.g. WikiLarge [Zhang and Lapata, 2017]), we currently do not measure their performance in more *abstractive* scenarios, i.e. cases with substantial modifications to the original sentences.

In this chapter we introduce **ASSET** (**A**bstractive **S**entence **S**implification **E**valuation and **T**uning), a new dataset for tuning and evaluation of automatic Sentence Simplification models. ASSET consists of 23,590 human simplifications associated with the 2,359 original sentences from TURKCORPUS (10 simplifications per original sentence). Simplifications in ASSET were collected via crowdsourcing (§ 7.2), and encompass a variety of rewriting transformations (§ 7.3), which make them simpler than those in TURKCORPUS and HSPLIT (§ 7.4), thus providing an additional suitable benchmark for comparing and evaluating automatic Sentence Simplification models. In addition, we study the applicability of standard metrics for evaluating Sentence Simplification using simplifications in ASSET as references (§ 7.5). We analyse whether BLEU [Papineni et al., 2002] or SARI [Xu et al., 2016] scores correlate with human judgements of fluency, adequacy and simplicity, and find that neither of the metrics shows a strong correlation with simplicity ratings. This motivates the need for developing better metrics for assessing Sentence Simplification when multiple rewriting transformations are performed.

We make the following contributions:

- A high quality large dataset for tuning and evaluation of Sentence Simplification models containing simplifications produced by applying multiple rewriting transformations.[1]

- An analysis of the characteristics of the dataset that turn it into a new suitable benchmark for evaluation.

- A study questioning the suitability of popular metrics for evaluating automatic simplifications in a multiple-transformation scenario.

[1] ASSET is released with a CC-BY-NC license at `https://github.com/facebookresearch/asset`.

7.1 Related Work

7.1.1 Studies on Human Simplification

A few corpus studies have been carried out to analyse how humans simplify sentences, and to attempt to determine the rewriting transformations that are performed.

[Petersen and Ostendorf, 2007] analyzed a corpus of 104 original and professionally simplified news articles in English. Sentences were manually aligned and each simplification instance was categorized as dropped (1-to-0 alignment), split (1-to-N), total (1-to-1) or merged (2-to-1). Some splits were further sub-categorized as edited (i.e. the sentence was split and some part was dropped) or different (i.e. same information but very different wording). This provides evidence that sentence splitting and deletion of information can be performed simultaneously.

[Aluísio et al., 2008] studied six corpora of simple texts (different genres) and a corpus of complex news texts in Brazilian Portuguese, to produce a manual for Portuguese text simplification [Specia et al., 2008]. It contains several rules to perform the task focused on syntactic alterations: to split adverbial/coordinated/subordinated sentences, to reorder clauses to a subject-verb-object structure, to transform passive to active voice, among others.

[Bott and Saggion, 2011b] worked with a dataset of 200 news articles in Spanish with their corresponding manual simplifications. After automatically aligning the sentences, the authors determined the simplification transformations performed: change (e.g. difficult words, pronouns, voice of verb), delete (words, phrases or clauses), insert (word or phrases), split (relative clauses, coordination, etc.), proximisation (add locative phrases, change from third to second person), reorder, select, and join (sentences).

From all these studies, it can be argued that the scope of rewriting transformations involved in the simplification process goes beyond only replacing words with simpler synonyms. In fact, human perception of complexity is most affected by syntactic features related to sentence structure [Brunato et al., 2018]. Therefore, since human editors make several changes to both the lexical content and syntactic structure of sentences when simplifying them, we should expect that models for automatic sentence simplification can also make such changes.

7.1.2 Evaluation Data for Sentence Simplification

Most datasets for Sentence Simplification [Zhu et al., 2010, Coster and Kauchak, 2011a, Hwang et al., 2015] consist of automatic sentence alignments between related articles in English Wikipedia (EW) and Simple English Wikipedia (SEW). In SEW, contributors are asked to write texts using simpler language, such as by shortening sentences or by using words from Basic English [Ogden, 1930]. However, [Yasseri et al., 2012] found that the syntactic complexity of sentences in SEW is almost the same as in EW. In addition, [Xu et al., 2015] determined that automatically-aligned simple sentences are sometimes just as complex as their original counterparts, with only a few words replaced or dropped and the rest of the sentences left unchanged.

More diverse simplifications are available in the Newsela corpus [Xu et al., 2015], a dataset of 1,130 news articles that were each manually simplified to up to 5 levels of simplicity. The parallel articles can be automatically aligned at the sentence level to train and test simplification models [Alva-Manchego et al., 2017, Štajner et al., 2018]. However, the Newsela corpus can only be accessed after signing a restrictive license that prevents publicly sharing train/test splits of the dataset, which impedes reproducibility.

Evaluating models on automatically-aligned sentences is problematic. Even more so if only one (potentially noisy) reference simplification for each original sentence is available. With this concern in mind, [Xu et al., 2016] collected the TURKCORPUS, a dataset with 2,359 original sentences from EW, each with 8 manual reference simplifications. The dataset is divided into two subsets: 2,000 sentences for validation and 359 for testing of sentence simplification models. TURKCORPUS is suitable for automatic evaluation that involves metrics requiring multiple references, such as BLEU [Papineni et al., 2002] and SARI [Xu et al., 2016]. However, [Xu et al., 2016] focused on simplifications through lexical paraphrasing, instructing annotators to rewrite sentences by reducing the number of difficult words or idioms, but without deleting content or splitting the sentences. This prevents evaluating a model's ability to perform a more diverse set of rewriting transformations when simplifying sentences. HSplit [Sulem et al., 2018b], on the other hand, provides simplifications involving only splitting for sentences in the test set of TURKCORPUS. We

build on TURKCORPUS and HSplit by collecting a dataset that provides several manually-produced simplifications involving multiple types of rewriting transformations.

7.1.3 Crowdsourcing Manual Simplifications

A few projects have been carried out to collect manual simplifications through crowdsourcing. [Pellow and Eskenazi, 2014a] built a corpus of everyday documents (e.g. driving test preparation materials), and analyzed the feasibly of crowdsourcing their sentence-level simplifications. Of all the quality control measures taken, the most successful was providing a training session to workers, since it allowed to block spammers and those without the skills to perform the task. Additionally, they proposed to use workers' self-reported confidence scores to flag submissions that could be discarded or reviewed. Later on, [Pellow and Eskenazi, 2014b] presented a preliminary study on producing simplifications through a collaborative process. Groups of four workers were assigned one sentence to simplify, and they had to discuss and agree on the process to perform it. Unfortunately, the data collected in these studies is no longer publicly available.

Simplifications in TURKCORPUS were also collected through crowdsourcing. Regarding the methodology followed, [Xu et al., 2016] only report removing bad workers after manual check of their first several submissions. More recently, [Scarton et al., 2018] used volunteers to collect simplifications for SimPA, a dataset with sentences from the Public Administration domain. One particular characteristic of the methodology followed is that lexical and syntactic simplifications were performed independently.

7.2 Creating ASSET

We extended TURKCORPUS [Xu et al., 2016] by using the same original sentences, but crowdsourced manual simplifications that encompass a richer set of rewriting transformations. Since TURKCORPUS was adopted as the standard dataset for evaluating Sentence Simplification models, several system outputs on this data are already publicly available [Zhang and Lapata, 2017, Zhao et al., 2018]. Therefore, we can now assess the capabilities of these and other systems in scenarios with varying simplification expectations: lexical para-

Original	Their eyes are quite small, and their visual acuity is poor.
TURKCORPUS	Their eyes are very little, and their sight is inferior.
HSplit	Their eyes are quite small. Their visual acuity is poor as well.
ASSET	They have small eyes and poor eyesight.
Original	His next work, Saturday, follows an especially eventful day in the life of a successful neurosurgeon.
TURKCORPUS	His next work at Saturday will be a successful Neurosurgeon.
HSplit	His next work was Saturday. It follows an especially eventful day in the life of a successful Neurosurgeon.
ASSET	"Saturday" records a very eventful day in the life of a successful neurosurgeon.
Original	He settled in London, devoting himself chiefly to practical teaching.
TURKCORPUS	He rooted in London, devoting himself mainly to practical teaching.
HSplit	He settled in London. He devoted himself chiefly to practical teaching.
ASSET	He lived in London. He was a teacher.

Table 7.1: Examples of simplifications collected for ASSET together with their corresponding version from TURKCORPUS and HSPLIT for the same original sentences.

phrasing with TURKCORPUS, sentence splitting with HSplit, and multiple transformations with ASSET.

7.2.1 Data Collection Protocol

Manual simplifications were collected using Amazon Mechanical Turk (AMT). AMT allows us to publish HITs (Human Intelligence Tasks), which workers can choose to work on, submit an answer, and collect a reward if the work is approved. This was also the platform used for TURKCORPUS.

Worker Requirements. Participants were workers who: (1) have a HIT approval rate $>= 95\%$; (2) have a number of HITs approved > 1000; (3) are residents of the United States of America, the United Kingdom or Canada; and (4) passed the corresponding Qualification Test designed for our task (more details below). The first two requirements are measured by the AMT platform and ensure that the workers have experience on different tasks and have had most of their work approved by previous requesters. The last two requirements are intended to ensure that the workers have a proficient level of English, and are capable of performing the simplification task.

Qualification Test. We provided a training session to workers in the form of a Qualification Test (QT). Following [Pellow and Eskenazi, 2014a], we showed them explanations and examples of multiple simplification transformations (see details below). Each HIT consisted of three sentences to simplify, and all submissions were manually checked to filter out spammers and workers who could not perform the task correctly. The sentences used in this stage were extracted from the QATS dataset [Štajner et al., 2016b]. We had 100 workers take the QT, out of which 42 passed the test (42%) and worked on the task.

Annotation Round. Workers who passed the QT had access to this round. Similar to [Pellow and Eskenazi, 2014a], each HIT now consisted of four original sentences that needed to be simplified. In addition to the simplification of each sentence, workers were asked to submit confidence scores on their simplifications using a 5-point likert scale (1:Very Low, 5:Very High). We collected 10 simplifications (similar to [Pellow and Eskenazi, 2014a]) for each of the 2,359 original sentences in TURKCORPUS.

Simplification Instructions. For both the QT and the Annotation Round, workers received the same set of instructions about how to simplify a sentence. We provided examples of lexical paraphrasing (lexical simplification and reordering), sentence splitting, and compression (deleting unimportant information). We also included an example where all transformations were performed. However, we clarified that it was at their discretion to decide which types of rewriting to execute in any given original sentence.[2]

Table 7.1 presents a few examples of simplifications in ASSET, together with references from TURKCORPUS and HSPLIT, randomly sampled for the same original sentences. It can be noticed that annotators in ASSET had more freedom to change the structure of the original sentences.

7.2.2 Dataset Statistics

ASSET contains 23,590 human simplifications associated with the 2,359 original sentences from TURKCORPUS (2,000 from the validation set and 359 from the test set). Table 7.2

[2]Full instructions are available in the dataset's repository.

presents some general statistics from simplifications in ASSET. We show the same statistics for TURKCORPUS and HSPLIT for comparison.[3]

In addition to having more references per original sentence, ASSET's simplifications offer more variability, for example containing many more instances of natural sentence splitting than TURKCORPUS. In addition, reference simplifications are shorter on average in ASSET, given that we allowed annotators to delete information that they considered unnecessary. In the next section, we further compare these datasets with more detailed text features.

	ASSET	TURKCORPUS	HSPLIT
Original Sentences	2,359	2,359	359
Num. of References	10	8	4
Type of Simp. Instances			
1-to-1	17,245	18,499	408
1-to-N	6,345	373	1,028
Tokens per Reference	19.04	21.29	25.49

Table 7.2: General surface statistics for ASSET compared with TURKCORPUS and HSPLIT. A simplification instance is an original-simplified sentence pair.

7.3 Rewriting Transformations in ASSET

We study the simplifications collected for ASSET through a series of text features to measure the *abstractiveness* of the rewriting transformations performed by the annotators. From here on, the analysis and statistics reported refer to the test set only (i.e. 359 original sentences), so that we can fairly compare ASSET, TURKCORPUS and HSPLIT.

7.3.1 Text Features

In order to quantify the rewriting transformations, we computed several low-level features for all simplification instances using the features from Chapter 5:

[3]HSPLIT is composed of two sets of simplifications: one where annotators were asked to split sentences as much as they could, and one where they were asked to split the original sentence only if it made the simplification easier to read and understand. However, we consider HSPLIT as a whole because differences between datasets far outweigh differences between these two sets.

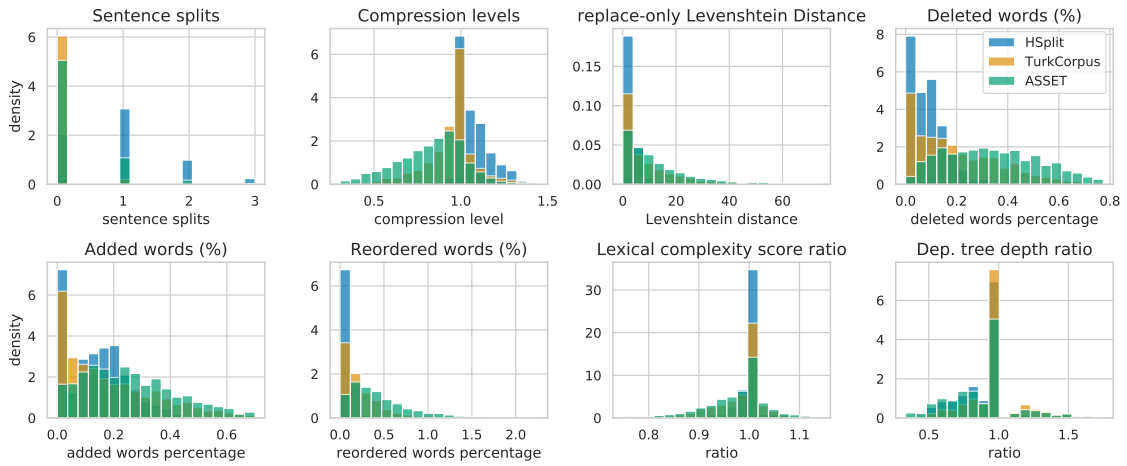


Figure 7.1: Density of text features in simplifications from HSPLIT, TURKCORPUS, and ASSET.

- **Number of sentence splits:** Corresponds to the difference between the number of sentences in the simplification and the number of sentences in the original sentence. In `tseval`, the number of sentences is calculated using NLTK [Bird and Loper, 2004].

- **Compression level:** Number of characters in the simplification divided by the number of characters in the original sentence.

- **Replace-only Levenshtein distance:** Computed as the normalized character-level Levenshtein distance [Levenshtein, 1966] for replace operations only, between the original sentence and the simplification. Replace-only Levenshtein distance is computed as follows (with o the original sentence and s the simplification):

$$\frac{replace\_ops(o,s)}{min(len(o),len(s))}$$

  We do not consider insertions and deletions in the Levenshtein distance computation so that this feature is independent from the compression level. It therefore serves as a proxy for measuring the lexical paraphrases of the simplification.

- **Proportion of words deleted, added and reordered:** Number of words deleted/reordered from the original sentence divided by the number of words in the original sentence; and the number of words that were added to the original sentence divided by the

number of words in the simplification.

- **Exact match:** Boolean feature that equals to true when the original sentence and the simplification are exactly the same, to account for unchanged sentences.

- **Word deletion only:** Boolean feature that equals to true when the simplification is obtained only by deleting words from the original sentence. This feature captures extractive compression.

- **Lexical complexity score ratio:** We compute the score as the mean squared log-ranks of content words in a sentence (i.e. without stopwords). We use the 50k most frequent words of the FastText word embeddings vocabulary [Bojanowski et al., 2017]. This vocabulary was originally sorted with frequencies of words in the Common Crawl. This score is a proxy to the lexical complexity of the sentence given that word ranks (in a frequency table) have been shown to be best indicators of word complexity [Paetzold and Specia, 2016a]. The ratio is then the value of this score on the simplification divided by that of the original sentence.

- **Dependency tree depth ratio:** We compute the ratio of the depth of the dependency parse tree of the simplification relative to that of the original sentence. When a simplification is composed by more than one sentence, we choose the maximum depth of all dependency trees. Parsing is performed using spaCy.[4] This feature serves as a proxy to measure improvements in structural simplicity.

Each feature was computed for all simplification instances in the dataset and then aggregated as a histogram (Figure 7.1) and as a percentage (Table 7.3).

7.3.2 Results and Analysis

Figure 7.1 shows the density of all features in ASSET, and compares them with those in TURKCORPUS and HSPLIT. Table 7.3 highlights some of these statistics. In particular, we report the percentage of sentences that: have at least one sentence split, have a compression

[4] `github.com/explosion/spaCy`

	ASSET	TURKCORPUS	HSPLIT
Sentence Splitting	20.2%	4.6%	68.2%
Compression (<75%)	31.2%	9.9%	0.1%
Word Reordering	28.3%	19.4%	10.1%
Exact Match	0.4%	16.3%	26.5%
Word Deletion Only	4.5%	3.9%	0.0%

Table 7.3: Percentage of simplifications featuring one of different rewriting transformations operated in ASSET, TURKCORPUS and HSPLIT. A simplification is considered as compressed when its character length is less than 75% of that of the original sentence.

level of 75% or lower, have at least one reordered word, are exact copies of the original sentences, and operated word deletion only (e.g. by removing only an adverb).

Sentence splits are practically non-existent in TURKCORPUS (only 4.6% have one split or more), and are more present and distributed in HSPLIT. In ASSET, annotators tended to not split sentences, and those who did mostly divided the original sentence into just two sentences (1 split).

Compression is a differentiating feature of ASSET. Both TURKCORPUS and HSPLIT have high density of a compression ratio of 1.0, which means that no compression was performed. In fact, HSPLIT has several instances with compression levels greater than 1.0, which could be explained by splitting requiring adding words to preserve fluency. In contrast, ASSET offers more variability, perhaps signaling that annotators consider deleting information as an important simplification operation.

By analyzing replace-only Levenshtein distance, we can see that simplifications in ASSET paraphrase the input more. For TURKCORPUS and HSPLIT, most simplifications are similar to their original counterparts (higher densities closer to 0). On the other hand, ASSET's simplifications are distributed in all levels, indicating more diversity in the rewordings performed. This observation is complemented by the distributions of deleted, added and reordered words. Both TURKCORPUS and HSPLIT have high densities of ratios close to 0.0 in all these features, while ASSET's are more distributed. Moreover, these ratios are rarely equal to 0 (low density), meaning that for most simplifications, at least some effort was put into rewriting the original sentence. This is confirmed by the low percentage of exact matches in ASSET (0.4%) with respect to TURKCORPUS (16.3%) and HSPLIT

(26.5%). Once again, it suggests that more rewriting transformations are being performed in ASSET.

In terms of lexical complexity, HSPLIT has a high density of ratios close to 1.0 due to its simplifications being structural and not lexical. TURKCORPUS offers more variability, as expected, but still their simplifications contain a high number of words that are equally complex, perhaps due to most simplifications just changing a few words. On the other hand, ASSET's simplifications are more distributed across different levels of reductions in lexical complexity.

Finally, all datasets show high densities of a 1.0 ratio in dependency tree depth. This could mean that significant structural changes were not made, which is indicated by most instances corresponding to operations other than splitting. However, ASSET still contains more simplifications that reduce syntactic complexity than TURKCORPUS and HSPLIT.

7.4 Rating Simplifications in ASSET

Here we measure the quality of the collected simplifications using human judges. In particular, we study if the *abstractive* simplifications in ASSET (test set) are preferred over lexical-paraphrase-only or splitting-only simplifications in TURKCORPUS (test set) and HSPLIT, respectively.

7.4.1 Collecting Human Preferences

Preference judgments were crowdsourced with a protocol similar to that of the simplifications (§ 7.2.1).

Selecting Human Judges. Workers needed to comply with the same basic requirements as described in § 7.2.1. For this task, the Qualification Test (QT) consisted in rating the quality of simplifications based on three criteria: fluency (or grammaticality), adequacy (or meaning preservation), and simplicity. Each HIT consisted of six original-simplified sentence pairs, and workers were asked to use a continuous scale (0-100) to submit their level of agreement (0: Strongly disagree, 100: Strongly agree) with the following statements:

1. The Simplified sentence adequately expresses the meaning of the Original, perhaps omitting the least important information.

2. The Simplified sentence is fluent, there are no grammatical errors.

3. The Simplified sentence is easier to understand than the Original sentence.

Using continuous scales when crowdsourcing human evaluations is common practice in Machine Translation [Bojar et al., 2018, Barrault et al., 2019], since it results in higher levels of inter-annotator consistency [Graham et al., 2013]. The six sentence pairs for the Rating QT consisted of:

- Three submissions to the Annotation QT, manually selected so that one contains splitting, one has a medium level of compression, and one contains grammatical and spelling mistakes. These allowed to check that the particular characteristics of each sentence pair affect the corresponding evaluation criteria.

- One sentence pair extracted from WikiLarge [Zhang and Lapata, 2017] that contains several sentence splits. This instance appeared twice in the HIT and allowed checking for intra-annotator consistency.

- One sentence pair from WikiLarge where the Original and the Simplification had no relation to each other. This served to check the attention level of the worker.

All submitted ratings were manually reviewed to validate the quality control established and to select the qualified workers for the task.

Preference Task. For each of the 359 original sentences in the test set, we randomly sampled one reference simplification from ASSET and one from TURKCORPUS, and then asked qualified workers to choose which simplification answers best each of the following questions:

- **Fluency**: Which sentence is more fluent?

- **Meaning**: Which sentence expresses the original meaning the best?

	Fluency	Meaning	Simplicity
ASSET	**38.4%***	23.7%	**41.2%***
TURKCORPUS	22.8%	**37.9%***	20.1%
Similar	38.7%	38.4%	38.7%
ASSET	**53.5%***	17.0%	**59.0%***
HSPLIT	19.5%	**51.5%***	14.8%
Similar	27.0%	31.5%	26.2%

Table 7.4: Percentages of human judges who preferred simplifications in ASSET or TURKCORPUS, and ASSET or HSPLIT, out of 359 comparisons. * indicates a statistically significant difference between the two datasets (binomial test with p-value < 0.001).

- **Simplicity**: Which sentence is easier to read and understand?

Workers were also allowed to judge simplifications as "similar" when they could not determine which one was better. The same process was followed to compare simplifications in ASSET against those in HSPLIT. Each HIT consisted of 10 sentence pairs.

7.4.2 Results and Analysis

Table 7.4 (top section) presents, for each evaluation dimension, the percentage of times a simplification from ASSET or TURKCORPUS was preferred over the other, and the percentage of times they were judged as "similar". In general, judges preferred ASSET's simplifications in terms of fluency and simplicity. However, they found TURKCORPUS' simplifications more meaning preserving. This is expected since they were produced mainly by replacing words/phrases with virtually no deletion of content.

A similar behaviour was observed when comparing ASSET to HSPLIT (bottom section of Table 7.4). In this case, however, the differences in preferences are greater than with TURKCORPUS. This could indicate that changes in syntactic structure are not enough for a sentence to be consider simpler.

7.5 Evaluating Evaluation Metrics

In this Section we study the behaviour of evaluation metrics for Sentence Simplification when using ASSET's simplifications (test set) as references. In particular, we measure the

correlation of standard metrics with human judgements of fluency, adequacy and simplicity, on simplifications produced by automatic systems. In our experiments, we used the implementations of these metrics available in the EASSE package for automatic sentence simplification evaluation that we introduced in Chapter 6.[5]

7.5.1 Experimental Setup

Evaluation Metrics. We analyzed the behaviour of two standard metrics in automatic evaluation of Sentence Simplification outputs: BLEU [Papineni et al., 2002] and SARI [Xu et al., 2016]. BLEU is a precision-oriented metric that relies on the number of n-grams in the output that match n-grams in the references, independently of position. SARI measures improvement in the simplicity of a sentence based on the n-grams added, deleted and kept by the simplification system. It does so by comparing the output of the simplification model to multiple references and the original sentence, using both precision and recall. BLEU has shown positive correlation with human judgements of grammaticality and meaning preservation [Štajner et al., 2014, Wubben et al., 2012, Xu et al., 2016], while SARI has high correlation with judgements of simplicity gain [Xu et al., 2016]. We computed all the scores at sentence-level as in the experiment by [Xu et al., 2016], where they compared sentence-level correlations of FKGL, BLEU and SARI with human ratings. We used a smoothed sentence-level version of BLEU so that comparison is possible, even though BLEU was designed as a corpus-level metric.

System Outputs. We used publicly-available simplifications produced by automatic Sentence Simplification systems: PBSMT-R [Wubben et al., 2012], which is a phrase-based MT model; Hybrid [Narayan and Gardent, 2014], which uses phrase-based MT coupled with semantic analysis; SBSMT-SARI [Xu et al., 2016], which relies on syntax-based MT; NTS-SARI [Nisioi et al., 2017], a neural sequence-to-sequence model with a standard encoder-decoder architecture; and ACCESS, a system that we introduce and detail in Chapter 9 based on an encoder-decoder architecture conditioned on explicit attributes of sentence simplification.

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

Collection of Human Ratings. We randomly chose 100 original sentences from ASSET and, for each of them, we sampled one system simplification. The automatic simplifications were selected so that the distribution of simplification transformations (e.g. sentence splitting, compression, paraphrases) would match that from human simplifications in ASSET. That was done so that we could obtain a sample that has variability in the types of rewritings performed. For each sentence pair (original and automatic simplification), we crowdsourced 15 human ratings on fluency (i.e. grammaticality), adequacy (i.e. meaning preservation) and simplicity, using the same worker selection criteria and HIT design of the Qualification Test as in § 7.4.1.

7.5.2 Inter-Annotator Agreement

We followed the process suggested in [Graham et al., 2013]. First, we normalized the scores of each rater by their individual mean and standard deviation, which helps eliminate individual judge preferences. Then, the normalized continuous scores were converted to five interval categories using equally spaced bins. After that, we followed [Pavlick and Tetreault, 2016] and computed quadratic weighted Cohen's κ [Cohen, 1968] simulating two raters: for each sentence, we chose one worker's rating as the category for annotator A, and selected the rounded average scores for the remaining workers as the category for annotator B. We then computed κ for this pair over the whole dataset. We repeated the process 1,000 times to compute the mean and variance of κ. The resulting values are: 0.687 ± 0.028 for Fluency, 0.686 ± 0.030 for Meaning and 0.628 ± 0.032 for Simplicity. All values point to a moderate level of agreement, which is in line with the subjective nature of the simplification task.

7.5.3 Correlation with Evaluation Metrics

We computed the Pearson correlation between the normalized ratings and the evaluation metrics of our interest (BLEU and SARI) using ASSET or TURKCORPUS as the set of references. We refrained from experimenting with HSPLIT since neither BLEU nor SARI correlate with human judgements when calculated using that dataset as references [Sulem et al., 2018b]. Results are reported in Table 7.5.

Metric	References	Fluency	Meaning	Simplicity
BLEU	ASSET	0.42*	0.61*	0.31*
	TURKCORPUS	0.35*	0.59*	0.18
SARI	ASSET	0.16	0.13	0.28*
	TURKCORPUS	0.14	0.10	0.17

Table 7.5: Pearson correlation of human ratings with **automatic metrics** on system simplifications. * indicates a significance level of p-value < 0.05.

BLEU shows a strong positive correlation with Meaning Preservation using either simplifications from ASSET or TURKCORPUS as references. There is also some positive correlation with Fluency judgements, but that is not always the case for Simplicity: no correlation when using TURKCORPUS and moderate when using ASSET. This is in line with previous studies that have shown that BLEU is not a good estimate for simplicity [Wubben et al., 2012, Xu et al., 2016, Sulem et al., 2018a].

In the case of SARI, correlations are positive but low with all criteria and significant only for simplicity with ASSET's references. [Xu et al., 2016] showed that SARI correlated with human judgements of simplicity gain, when instructing judges to *"grade the quality of the variations by identifying the words/phrases that are altered, and counting how many of them are good simplifications"*.[6] The judgements they requested differ from the ones we collected, since theirs were tailored to rate simplifications produced by lexical paraphrasing only. These results show that SARI might not be suitable for the evaluation of automatic simplifications with multiple rewrite operations.

In Table 7.6, we further analyse the human ratings collected, and compute their correlations with similar text features as in § 7.3. The results shown reinforce our previous observations that judgements on Meaning correlate with making few changes to the sentence: strong negative correlation with Levenshtein distance, and strong negative correlation with proportion of words added, deleted, and reordered. No conclusions could be drawn with respect to Simplicity.

[6] https://github.com/cocoxu/simplification/tree/master/HIT_MTurk_crowdsourcing

Feature	Fluency	Meaning	Simplicity
Length	0.12	0.31*	0.03
Sentence Splits	-0.13	-0.06	-0.08
Compression Level	0.26*	0.46*	0.04
Levenshtein Distance	-0.40*	-0.67*	-0.18
Replace-only Lev. Dist.	-0.04	-0.17	-0.06
Prop. Deleted Words	-0.43*	-0.67*	-0.19
Prop. Added Words	-0.19	-0.38*	-0.12
Prop. Reordered Words	-0.37*	-0.57*	-0.18
Dep. Tree Depth Ratio	0.20	0.24	0.06
Word Rank Ratio	0.04	0.08	-0.05

Table 7.6: Pearson correlation of human ratings with **text features** on system simplifications. * indicates a significance level of p-value < 0.01.

7.6 Summary and Final Remarks

We have introduced ASSET, a new dataset for tuning and evaluation of Sentence Simplification models. Simplifications in ASSET were crowdsourced, and annotators were instructed to apply multiple rewriting transformations. This improves current publicly-available evaluation datasets, which are focused on only one type of transformation. Through several experiments, we have shown that ASSET contains simplifications that are more *abstractive*, and that are consider simpler than those in other evaluation corpora. Furthermore, we have motivated the need to develop new metrics for automatic evaluation of Sentence Simplification models, especially when evaluating simplifications with multiple rewriting operations. In Chapter 8 we show that traditional metrics perform even more poorly on human generated simplification and explore new neural-based evaluation metrics for Sentence Simplification. Finally, we hope that ASSET's multi-transformation features will motivate the development of Sentence Simplification models that benefit a variety of target audiences according to their specific needs such as people with low literacy or cognitive disabilities. In Chapter 9 we propose a controllable Sentence Simplification system in the hope that it can be better adapted to any type of target audience.

Chapter 8

A New Approach to Automatic Evaluation of Sentence Simplification

In previous Chapters, we explored multiple aspects of Sentence Simplification evaluation: quality estimation, traditional metrics, and reference simplifications with various type of rewriting operations. However, automatic evaluation for NLG is known to be an open research question [Peyrard, 2019, Scialom et al., 2020b], and Sentence Simplification is no exception [Xu et al., 2016, Sulem et al., 2018b]. The standard for evaluation in Sentence Simplification, BLEU [Papineni et al., 2002] and SARI [Xu et al., 2016], have subsequently been shown to have low correlation with human judgments for various settings of Sentence Simplification as highlighted in [Sulem et al., 2018b] and Chapter 7.

The recent BERTScore [Zhang et al., 2020] metric has been shown to compare favorably compared to BLEU in Machine Translation. However, it still computes a pairwise similarity at the token-level which has important theoretical limitations [Novikova et al., 2017]. In particular, not enough human references are available to cover all the possible ways to write the same idea.

Beyond token-level metrics, QUESTEVAL has recently obtained promising results in measuring Meaning Preservation in Summarization [Scialom et al., 2021]. However, it uses exact matches between tokens to compute an F1 score, penalizing the use of synonyms and reformulations, hence preventing a direct application for Sentence Simplification. In this chapter, we propose a simple modification of QUESTEVAL to adapt it to Sentence

Simplification.

Further, to the best of our knowledge, no work has yet studied the correlations for any of these two recent metrics for Sentence Simplification. We show that both BERTScore and QUESTEVAL improve over BLEU and SARI, achieving new state-of-the-art correlations on all measured dimensions: Fluency, Meaning Preservation and Simplicity. While this result is not surprising for Meaning Preservation, it is rather unexpected for Simplicity: neither BERTScore or QUESTEVAL should have the ability to measure the simplicity of a text. Indeed, QUESTEVAL only compares the factual content of two texts, irrespective of their complexity. BERTScore for its part is robust to synonyms and sentence structure: while this behavior is desirable in Machine Translation, this is not the case in Sentence Simplification where a simpler word or sentence structure should be scored higher.

We hypothesize that the inter-correlations between the evaluated dimensions could be responsible for spurious correlations: a system that generates simplifications that are not fluent tends to perform poorly also on Meaning Preservation and Simplicity.

Moreover, we are coming to a point where the outputs from neural systems are close to a human-level for their Fluency [Zellers et al., 2020, Scialom et al., 2020a]. We hypothesize that under this state of Fluency, the correlations of automatic metrics might vanish w.r.t. human judgment.

To investigate such phenomenon, we propose to analyse the correlations on human-written simplifications: such texts should be less prompt to spurious correlations given that most of them should be perfectly fluent. To this purpose, we release a new human evaluation of *human-written* simplifications.[1] This corpus allows us to conduct extensive experiments and better analyse the metrics' correlations w.r.t. human judgment. In particular, our findings show very different conclusions than the evaluation of system-generated simplification. For instance, neither BLEU or SARI significantly correlate with any dimensions. The only metrics with significant correlation are QUESTEVAL for Meaning Preservation and FKGL for Simplicity.

In summary, our contributions are:

[1] `http://dl.fbaipublicfiles.com/questeval/simplification_human_evaluations.tar.gz`

1. We propose an adaptation of QUESTEVAL for Sentence Simplification and show that it compares favorably on Meaning Preservation.

2. We release a new corpus of 9000 human evaluation of human-written simplifications.

3. We conduct an extensive analysis of several metrics, including for the first time the recent BERTScore and our adaptation of QUESTEVAL. We draw very different conclusions compared to previous works.

8.1 Related Work

Automatic metrics serves as a proxy for human judgments, their correlations with human ratings is therefore important to compare systems. [Xu et al., 2016] found significant correlations for SARI with Fluency, Meaning, and Simplicity, and for BLEU with Fluency and Meaning but not with Simplicity.

However in Chapter 7, we observed lower correlations of automatic metrics with human judgments than previously reported and released a set of human ratings of systems simplifications along with a corpus of human-written simplifications, ASSET. To the best of our knowledge, this is so far the largest published human rating dataset for Sentence Simplification, with a total of 9,000 ratings of system-generated simplifications.

Source Text: In the Soviet years, the Bolsheviks demolished two of Rostov's principal landmarks- St Alexander Nevsky cathedral (1908) and St George cathedral in Nakhichevan (1783-1807).				
Simplification: The Bolsheviks destroyed St. Alexander Nevsky cathedral and St. George cathedral in Nakhichevan during the Soviet years.				
Generated Question	**Answers**		**Score**	
	On Source	On Simplif.	F1	BERTScore
When did the Bolsheviks demolish St George cathedral?	the Soviet years	Soviet years	0.8	0.89
Who demolished St Alexander Nevsky cathedral?	demolished	destroyed	0.0	0.82
How many of Rostov's main landmarks were demolished?	two	Unanswerable	0.0	0.0
What cathedral was demolished in 1908?	Rostov	Unanswerable	0.0	0.0
[...]	[...]	[...]	[...]	[...]

Table 8.1: Example of questions automatically generated and answered by QUESTEVAL given a source text and its simplification.

8.2 Human Evaluation Corpora

In this section we describe the two human ratings corpora we used to compute the metric correlations: they provide assessments over simplifications originating from automatic systems or humans.

System-Likert We reuse the existing human evaluation corpus described from ASSET (Chapter 7). It is composed of ratings on systems-generated simplifications on a Likert Scale. Each simplification has been evaluated over three dimensions:

1. **Fluency**: how fluent is the evaluated text?
2. **Meaning Preservation**: how well the evaluated text expresses the original meaning?
3. **Simplicity**: to what extent is the evaluated text easier to read and understand?

In total, 100 unique simplifications were evaluated with, for each of them, 30 ratings per dimension.

Human-Likert We collect this second corpus following the exact same methodology used for *System-Likert*, obtaining 9000 ratings of *human-written* simplifications sampled from the references available in the test sets of ASSET and TURKCORPUS, and scored by human annotators given a 5-point Likert scale (1: Very Low, 5: Very High).

We follow the methodology of Chapter 7 and reuse the same interface. We collect annotations using Amazon Mechanical (AMT). The requirements for annotators are exactly the same as ASSET (Chapter 7), namely: (1) have a HIT approval rate $>= 95\%$; (2) have a number of HITs approved > 1000; (3) are residents of the United States of America, the United Kingdom or Canada; and (4) passed the corresponding Qualification Test designed for by the authors and provided on their repository.

The qualification test consists in a training session explaining what is Sentence Simplification and a rating session where the annotators had to rate 6 pairs of source-simplification pairs. Annotators were asked to use a (0: Strongly disagree - 100: Strongly agree) continuous scale to rate sentences on three aspects represented by the following statements:

1. The Simplified sentence adequately expresses the meaning of the Original, perhaps omitting the least important information.

2. The Simplified sentence is fluent, there are no grammatical errors.

3. The Simplified sentence is easier to understand than the Original sentence.

The 6 sentence pairs evaluated are the same as in Chapter 7, and were chosen to represent various simplification operations and typical errors in meaning preservation or fluency. We then manually evaluated qualification tests to filter out spammers or workers that didn't perform the task correctly.

We used the same 100 source sentences as the System-Likert corpus and sampled one simplification each from either ASSET, TURKCORPUS, or HSPLIT, resulting in 100 unique source-simplification pairs. We finally collected 30 ratings per pair and per dimension (fluency, meaning, simplicity) resulting in 9000 total ratings.

8.3 Metrics considered

8.3.1 Token-Level Metrics

This section serves as a brief reminder to the more detailed metrics' descriptions in Chapter 2.

FKGL [Kincaid et al., 1975] is a reference-less metric that measures readability using only sentence lengths and word lengths.

SARI [Xu et al., 2016] was designed for Sentence Simplification by measuring the accuracy and recall of words that are added, deleted and kept.

BLEU [Papineni et al., 2002] measures the overlap of n-grams between a reference text and the evaluated one.

BERTScore [Zhang et al., 2020] leverages on the contextualised representation of BERT to compute the similarity between the tokens.

These token-level metrics share the same limitation: they depend on the number of available references; given less references, their correlations naturally decrease.

8.3.2 QUESTEVAL for Sentence Simplification

Beyond token-level metrics, a trend of using Question Generation and Question Answering for Automatic Summarization evaluation has recently emerged [Chen et al., 2018, Scialom et al., 2019, 2021]. We consider the more recent QUESTEVAL [Scialom et al., 2021].

QUESTEVAL evaluates if a summary is factually consistent w.r.t. its source document. To do so, it (i) generates a list of questions on the evaluated summary, and (ii) retrieves the corresponding answers from the source document: if the answers are similar, the summary is deemed satisfactory.[2]

Adapting QUESTEVAL to Sentence Simplification To measure the similarity between two answers, the most popular approach in Question Answering is to compute the F1 score [Rajpurkar et al., 2016].

This is effective in the context of extractive Question Answering, since the answer belongs by definition to the input paragraph. In QUESTEVAL, the authors chose to compute the similarity via this F1. We argue that in Sentence Simplification, using synonyms and reformulations is inherent to the task. To alleviate this limitation, we propose to replace the F1 score with a more suitable metric: BERTScore. By leveraging its dense representations, a smoother function than the F1 can be computed, allowing for reformulations.

In Table 8.1 we show an example of a source text, its simplification and some of the generated questions by QUESTEVAL. The simplification used a synonym, replacing *demolished* with *destroyed*. While both *demolished* with *destroyed* share the same meaning, the F1 Score incorrectly scored 0, as opposed to BERTScore.[3]

[2]The QUESTEVAL metric is depicted in more detail in Figure 1 of the original paper [Scialom et al., 2021].

[3]It is also interesting that the third question (*How many of Rostov's main landmarks were demolished?*) was predicted to be unanswered. While the answer, i.e. *two*, could be deduced from the text, it could not

	Ref-less	System-generated simplifications			Human-written simplifications		
		Fluency	Simplicity	Meaning	Fluency	Simplicity	Meaning
Fluency		—	86.2**	79.5**	—	73.6**	52.7**
Simplicity		86.2**	—	67.2**	73.6**	—	37.0**
Meaning		79.5**	67.2**	—	52.7**	37.0**	—
-FKGL	✓	16.8	8.9	28.9*	**19.0**	**34.7***	2.9
SARI	✗	18.3	25.2	16.0	0.9	9.7	5.8
BLEU	✗	37.9**	30.2*	41.1**	15.2	12.1	9.8
BERTScore	✗	**53.6****	**41.5****	63.3**	13.8	8.7	19.4
QUESTEVAL	✓	45.8**	37.3**	**66.5****	-7.5	-7.4	**21.7***

Table 8.2: Pearson Correlation Coefficient between human judgment and automatic metrics for system-generated simplification (left-hand), and for human-generated simplifications (right-hand). We report -FKGL so higher is better for all the metrics (a lower FKGL is supposed to indicate a simpler text). * indicates p-value < 0.01 and ** < 0.001.

To BERTScore or not to BERTScore? Like BLEU, BERTScore is a *token-level metric*, and therefore suffers from token misalignment: two texts can share the same meaning but be written in very different ways. The longer the texts, the more likely their tokens will not be aligned. Further, BERTScore assigns high similarity to tokens with the same meaning, thus being robust to synonymy but oblivious to their complexity. It is also insensitive to simplified sentence structures (e.g. word reordering, sentence splitting). For these reasons, BERTScore is not suited for measuring simplicity, even when several references are available.

Nonetheless, for the same exact reasons, BERTScore can effectively be used as a similarity metric for the short answers generated in QUESTEVAL, see Table 8.1.

8.4 Results and Discussion

Metric Correlations on Systems Simplifications In the left half of Table 8.2, we report the Pearson correlations for 5 evaluations metrics.

Both SARI and FKGL do not perform well, with low correlations (<30) on all dimensions. Conversely, BERTScore and QUESTEVAL obtain the highest correlations, with an edge for BERTScore on Fluency and Simplicity and QUESTEVAL leading in Meaning. More surprisingly, both QUESTEVAL and BERTScore correlate on Simplicity ($\sim$40), de-

extracted. This emphasizes a current limitation for QUESTEVAL, which could largely benefit from better and more abstractive QA models in the future.

spite BERTScore being robust to synonyms, and QUESTEVAL only evaluating content preservation regardless to the complexity. Therefore, neither should be equipped to measure Simplicity.

Also visible in Table 8.2 are the strong inter-correlations between the three evaluated dimensions: e.g. the Fluency correlates with the Meaning better than any metric (79.1 Pearson coefficient). These inter-correlations could create undesired spurious correlations for the metrics. This would explain BERTScore and QUESTEVAL strong correlations on Simplicity.

Right for the wrong reasons? In order to get a deeper understanding , one needs to limit the inter-correlations between the different dimensions. With this purpose in mind, we compute the correlations, this time on *Human-Likert*, our corpus on human-written simplifications instead of system generated ones. We report the results in the right half of table 8.2.

All inter-correlations are lower than for system-generated simplifications although still high. In particular, the Meaning is less impacted by the Fluency and the Simplicity. This allows a clearer analysis of the intrinsic metric correlations, leading to very different conclusions.

With respect to Simplicity, neither BERTScore or QUESTEVAL correlate anymore. FKGL obtains the only significant result on this dimension (34.7).

For Meaning Preservation, QUESTEVAL achieves the highest and only significant correlation. This result is emphasized by - this time - the slight anti-correlation on Simplicity and Fluency. We also observe that BERTScore correlates slightly on all the dimensions but with no statistical significance. These results confirm that inter-correlations between dimensions cause spurious correlations among automatic metrics, when evaluated on system generated simplifications. In other words, a system might score higher with QUESTEVAL or BERTScore. *This does not necessarily mean that the system produces simpler sentences!*

8.5 Conclusion

In this chapter we adapted a Question-based metric to Sentence Simplification. By using BERTScore for the similarity function, we provide a smoother way to compare two answers than in the original metric, allowing to take into account synonyms.

Further, we conducted an extensive analysis of the metrics for Sentence Simplification on both system-generated and human-written examples. On system-generated simplifications, we show that both BERTScore and QUESTEVAL improve over BLEU and SARI, but likely due to spurious correlations. However, on the human-written simplifications, we raise concerns about very low correlations for most of traditional metrics: actually only FKGL and QUESTEVAL are able to significantly measure Simplicity and Meaning Preservation. This chapter thus calls for more frequent re-evaluation of the metrics, along with systems advances.

In future work, we plan on studying the interactions between the different dimensions more in depth, with the objective to propose an evaluation protocol that will allow to limit inter-correlations in human evaluation.

Part III

Towards more Adaptable Simplification Systems

Chapter 9

ACCESS: Controllable Sentence Simplification in English

In previous chapters, we saw that evaluation of Sentence Simplification is difficult, and in particular due to the fact that sentences can be simplified in many different ways. In this chapter, we propose to address this challenge from a modeling perspective by proposing a model that can be controlled based on the user preferences and evaluate it on English data. We use the English language because of the wider availability of training and evaluation data used for our supervised approach. In Part IV we will show how this method can be used in other languages where no labelled data is available.

Indeed, many audiences can benefit from Sentence Simplification, for instance people with cognitive disabilities such as aphasia [Carroll et al., 1998], dyslexia [Rello et al., 2013] and autism [Evans et al., 2014] but also for second language learners [Xia et al., 2016] and people with low literacy [Watanabe et al., 2009]. The type of simplification needed for each of these audiences is different. Some aphasic patients struggle to read sentences with a high cognitive load such as long sentences with intricate syntactic structures, whereas second language learners might not understand texts with rare or specific vocabulary. Yet, research in Sentence Simplification has been mostly focused on developing models that generate a single generic simplification for a given source text with no possibility to adapt outputs for the needs of various target populations.

In this chapter we propose a controllable simplification model that provides explicit

ways for users to manipulate and update simplified outputs as they see fit. This work only considers the task of *Sentence Simplification* (SS) where the input of the model is a single source sentence and the output can be composed of one sentence or split into multiple. Our work builds upon previous work on controllable text generation [Kikuchi et al., 2016, Fan et al., 2018, Scarton and Specia, 2018, Nishihara et al., 2019] where a Sequence-to-Sequence (Seq2Seq) model is modified to control attributes of the output text. We tailor this mechanism to the task of Sentence Simplification by considering relevant attributes of the output sentence such as the output length, the amount of paraphrasing, lexical complexity, and syntactic complexity. To this end, we condition the model at train time, by feeding control tokens representing these attributes along with the source sentence as additional inputs.

Our contributions are the following: (1) We adapt a parametrization mechanism to the specific task of Sentence Simplification by conditioning on relevant attributes; (2) We show through a detailed analysis that our model can indeed control the considered attributes, making the simplifications potentially able to fit the needs of various end audiences; (3) With careful calibration, our controllable parametrization improves the performance of out-of-the-box Seq2Seq models leading to a new state-of-the-art score of 41.87 SARI [Xu et al., 2016] on the WikiLarge benchmark [Zhang and Lapata, 2017], a +1.42 gain over previous scores, without requiring any external resource or modified training objective.

9.1 Related Work

9.1.1 Controllable Text Generation

Conditional training with Seq2Seq models was applied to multiple natural language processing tasks such as summarization [Kikuchi et al., 2016, Fan et al., 2018], dialog [See et al., 2019], sentence compression [Fevry and Phang, 2018, Mallinson et al., 2018] or poetry generation [Ghazvininejad et al., 2017].

Most approaches for controllable text generation are either decoding-based or learning-based.

Decoding-based methods Decoding- based methods use a standard Seq2Seq training setup but modify the system during decoding to control a given attribute. For instance, the length of summaries was controlled by preventing the decoder from generating the End-Of-Sentence token before reaching the desired length or by only selecting hypotheses of a given length during the beam search [Kikuchi et al., 2016]. Weighted decoding (i.e. assigning weights to specific words during decoding) was also used with dialog models [See et al., 2019] or poetry generation models [Ghazvininejad et al., 2017] to control the number of repetitions, alliterations, sentiment or style.

Learning-based methods On the other hand, learning-based methods condition Seq2Seq on the considered attribute at train time, and can then be used to control the output at inference time. [Kikuchi et al., 2016] explored learning-based methods to control the length of summaries, e.g. by feeding a target length vector to the neural network. They concluded that learning-based methods worked better than decoding-based methods and allowed finer control on the length without degrading performances. Length control was likewise used in sentence compression by feeding the network a length countdown scalar [Fevry and Phang, 2018] or a length vector [Mallinson et al., 2018]. [Ficler and Goldberg, 2017] concatenate a context vector to the hidden state of each time step of their recurrent neural network decoder. This context vector represents the controlled stylistic attributes of the text, where an embedding is learnt for each attribute value. [Hu et al., 2017] achieved controlled text generation by disentangling the latent space representations of a variational auto-encoder between the text representation and its controlled attributes such as sentiment and tense. They impose the latent space structure during training by using additional discriminators.

Concurrently to the publication of this work [Martin et al., 2020a], [Mallinson and Lapata, 2019] have proposed a controllable approach using lexical and syntactic constraints. Lexical constraints operate at the token-level: the model is trained at replacing or keeping specific tokens. At test time, the user can then manually select which tokens to keep or discard during the simplification process. Additional syntactic information is added in the form of linearized parse trees to the source and target sentences. This allows using syntactic simplification rules at test time.

In this work we condition the generation process by concatenating plain text control tokens to the source text. This method only modifies the source data and not the training procedure. Such mechanism was used to control politeness in MT [Sennrich et al., 2016a], to control summaries in terms of length, of news source style, or to make the summary more focused on a given named entity [Fan et al., 2018]. It was applied to Sentence Simplification in [Scarton and Specia, 2018] and [Nishihara et al., 2019] to control grade-level readability or coarse-grained simplification operations. Our work goes further by using a more diverse set of control tokens that represent specific grammatical attributes of the Sentence Simplification process. Moreover, we investigate the influence of those control tokens on the generated simplification in a detailed analysis.

9.2 Adding Control Tokens to Seq2Seq

We present ACCESS, our approach for **A**udien**C**e-**CE**ntric **S**entence **S**implification. We want to control the process of Sentence Simplification using explicit control tokens. We first identify attributes that cover important aspects of the simplification process and then find explicit control tokens to represent each of those attributes. Parametrization is then achieved by conditioning a Seq2Seq model on those control tokens.

9.2.1 Controlled Attributes

Based on previous findings, we identify four attributes related to the process of Sentence Simplification: amount of compression, amount of paraphrasing, lexical complexity and syntactic complexity,.

- **Amount of compression:** The amount of compression is directly dependent on the length of sentences which is itself very correlated to simplicity (Chapter 5), and is one of the two variables used in FKGL [Kincaid et al., 1975]. It also accounts for the amount of content that is preserved between the source and target text, and can therefore control the simplicity-adequacy trade-off that is witnessed in Sentence Simplification [Schwarzer and Kauchak, 2018].

- **Paraphrasing:** Paraphrasing is an important aspect for good Sentence Simplification systems [Wubben et al., 2012], especially because it allows the user from choosing if he prefers very safe simplifications (i.e. close to the source) or to try and simplify the input more at the cost of more mistakes when using imperfect systems. The amount of paraphrasing was also shown to correlate with human judgment of meaning preservation and simplicity sometimes even more than traditional metrics such as BLEU [Papineni et al., 2002] and SARI [Xu et al., 2016].

- **Lexical and Syntactic complexity:** [Shardlow, 2014] identified lexical simplification and syntactic simplification as core components of Sentence Simplification systems, which often decomposes there approach into these two sub-components. Audiences also have different simplification needs along these two attributes. In order to understand a text correctly, second language learner will require a text with less complicated words. On the other hand, some specific types of aphasia will make people struggle more with complex syntactic structures, intricate clauses, and long sentence, thus requiring syntactic simplification.

Other more specific attributes could be considered such as the tense or the use passive-active voice. We only consider the previous attributes for simplicity and leave the rest for future work. We do not consider "readability" measured with FKGL because it is just a linear combination of other attributes, namely sentence length and word complexity.

9.2.2 Explicit Control Tokens

For each of the four aforementioned attributes, we choose an explicit "proxy" control token that can be computed using the source and simplified sentence and used as a plain text token. We describe these for explicit control tokens in this subsection.

- **NbChars:** character length ratio between source sentence and target sentence (compression level). This control token accounts for sentence compression, and content deletion. We showed in Chapter 5 that simplicity is best correlated with length-based metrics, and especially in terms of number of characters. The number of charac-

ters indeed accounts for the lengths of words which is itself correlated to lexical complexity.

- **LevSim:** normalized character-level Levenshtein similarity [Levenshtein, 1966] between source and target. LevSim quantifies the amount of modification operated on the source sentence (through paraphrasing, adding and deleting content).

- **WordRank:** as a proxy to lexical complexity, we compute a sentence-level measure, that we call *WordRank*, by taking the third-quartile of log-ranks (inverse frequency order) of all words in a sentence. We subsequently divide the *WordRank* of the target by that of the source to get a ratio. Word frequencies have shown to be the best indicators of word complexity in the Semeval 2016 task 11 [Paetzold and Specia, 2016a].

- **DepTreeDepth:** maximum depth of the dependency tree of the source divided by that of the target (we do not feed any syntactic information other than this ratio to the model). This control token is designed to approximate syntactic complexity. Deeper dependency trees indicate dependencies that span longer and possibly more intricate sentences. DepTreeDepth proved better in early experiments over other candidates for measuring syntactic complexity such as the maximum length of a dependency relation, or the maximum inter-word dependency flux.

We parametrize a Seq2Seq model on a given attribute of the target simplification, e.g. its length, by prepending a control token at the beginning of the source sentence. The control token value is the ratio[1] of this control token calculated on the target sentence with respect to its value on the source sentence. For example when trying to control the number of characters of a generated simplification, we compute the compression ratio between the number of characters in the source and the number of characters in the target sentence (see Table 9.1 for an illustration). Ratios are discretized into bins of fixed width of 0.05 in our experiments and capped to a maximum ratio of 2. Control tokens are then included in the vocabulary (40 unique values per control token).

[1]Early experiments showed that using a ratio instead of an absolute value allowed finer control on the respective attributes.

Source	*<NbChars_0.3> <LevSim_0.4> He settled in London , devoting himself chiefly to practical teaching .*
Target	*He teaches in London .*

Table 9.1: Example of parametrization on the number of characters. Here the source and target simplifications respectively contain 71 and 22 characters which gives a compression ratio of 0.3. We prepend the <NbChars_0.3> token to the source sentence. Similarly, the Levenshtein similarity between the source and the sentence is 0.37 which gives the <LevSim_0.4> control token after bucketing.

At inference time, we just set the ratio to a fixed value for all samples[2]. For instance, to get simplifications that are 80% of the source length, we prepend the token <NbChars_0.8> to each source sentence. This fixed ratio can be user-defined or automatically set. In our setting, we choose fixed ratios that maximize the SARI on the validation set.

9.3 Experiments

9.3.1 Experimental Setting

Architecture details We train a Transformer model [Vaswani et al., 2017] using the FairSeq toolkit [Ott et al., 2019]. Our architecture is the base architecture from [Vaswani et al., 2017]. We used an embedding dimension of 512, fully connected layers of dimension 2048, 8 attention heads, 6 layers in the encoder and 6 layers in the decoder. Dropout is set to 0.2. We use the Adam optimizer [Kingma and Ba, 2014] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$ and a learning rate of $lr = 0.00011$. We add label smoothing with a uniform prior distribution of $\epsilon = 0.54$. We use early stopping when SARI does not increase for more than 5 epochs. We tokenize sentences using the NLTK NIST tokenizer and preprocess using SentencePiece [Kudo and Richardson, 2018] with 10k vocabulary size to handle rare and unknown words. For generation we use beam search with a beam size of 8. [3]

[2]We did not investigate predicting ratios on a per sentence basis as done by [Scarton and Specia, 2018], and leave this for future work. End-users can nonetheless choose the target ratios as they see fit, for each source sentence.

[3]Code and pretrained models are released with an open-source license at `https://github.com/facebookresearch/access`.

Training and evaluation datasets Our models are trained and evaluated on the **WikiLarge dataset** [Zhang and Lapata, 2017] which contains 296,402/2,000/359 samples (train/validation/test). WikiLarge is a set of automatically aligned complex-simple sentence pairs from English Wikipedia (EW) and Simple English Wikipedia (SEW). It is compiled from previous extractions of EW-SEW [Zhu et al., 2010, Woodsend and Lapata, 2011, Kauchak, 2013]. Its validation and test sets are taken from Turkcorpus [Xu et al., 2016], where each complex sentence has 8 human simplifications created by Amazon Mechanical Turk workers.[4] Human annotators were instructed to only paraphrase the source sentences while keeping as much meaning as possible. Hence, no sentence splitting, minimal structural simplification and little content reduction occurs in this test set [Xu et al., 2016]. We are not able to use the Newsela dataset [Xu et al., 2015] because of legal constraints related to its limited public availability. The Newsela dataset can only be accessed by signing a one year Data Sharing Agreement and comes with a restrictive non-commercial license. Additionally, all publications using the dataset need to be sent in advance to Newsela for approval. This limited public availability also prevents the research community from agreeing on a public train/validation/test split which hampers reproducibility of results.[5]

Evaluation metrics We evaluate our methods with **FKGL** (Flesch-Kincaid Grade Level) [Kincaid et al., 1975] to account for simplicity and **SARI** [Xu et al., 2016] as an overall score. FKGL is a commonly used metric for measuring readability however it should not be used alone for evaluating systems because it does not account for grammaticality and meaning preservation [Wubben et al., 2012]. Please refer to Chapter 2 for more details on those metrics.

We compute FKGL and SARI using the EASSE python package for Sentence Simplification introduced in Chapter 6. We do not use BLEU because it is not suitable for evaluating Sentence Simplification systems [Sulem et al., 2018b]. BLEU is also misleading because it favors models that do not modify the source sentence [Xu et al., 2016] on TurkCorpus. For instance copying the source sentence in place of simplification gives a BLEU of 99.37 on WikiLarge.

[4]We do not use ASSET in this chapter, because this work was conducted prior to the creation of ASSET.
[5]We were able to use Newsela in Chapter 10 though.

9.3.2 Overall Performance

Table 9.2 compares our best model to state-of-the-art methods:

PBMT-R [Wubben et al., 2012]

Phrase-Based MT system with candidate reranking. Dissimilar candidates are favored based on their Levenshtein distance to the source.

Hybrid [Narayan and Gardent, 2014]

Deep semantics sentence representation fed to a monolingual MT system.

SBMT+PPDB+SARI [Xu et al., 2016]

Syntax-based MT model augmented using the PPDB paraphrase database [Pavlick et al., 2015] and fine-tuned towards SARI.

DRESS-LS [Zhang and Lapata, 2017]

Seq2Seq trained with reinforcement learning, combined with a lexical simplification model.

Pointer+Ent+Par [Guo et al., 2018]

Seq2Seq model based on the pointer-copy mechanism and trained via multi-task learning on the Entailment and Paraphrase Generation tasks.

NTS+SARI [Nisioi et al., 2017]

Standard Seq2Seq model. The second beam search hypothesis is selected during decoding; the hypothesis number is an hyper-parameter fine-tuned with SARI.

NSELSTM-S [Vu et al., 2018]

Seq2Seq with a memory-augmented Neural Semantic Encoder, tuned with SARI.

DMASS+DCSS [Zhao et al., 2018]

Seq2Seq integrating the simple PPDB simplification database [Pavlick and Callison-Burch, 2016] as a dynamic memory. The database is also used to modify the loss and re-weight word probabilities to favor simpler words.

WikiLarge **(test)**	SARI ↑	FKGL ↓
PBMT-R	38.56	8.33
Hybrid	31.40	**4.56**
SBMT+PPDB+SARI	39.96	7.29
DRESS-LS	37.27	6.62
Pointer+Ent+Par	37.45	—
NTS+SARI	37.25	—
NSELSTM-S	36.88	—
DMASS+DCSS	40.45	8.04
ACCESS: $\text{NbChars}_{0.95}$ + $\text{LevSim}_{0.75}$ + $\text{WordRank}_{0.75}$	**41.87**	7.22

Table 9.2: Comparison to the literature. We report the results of the model that performed the best on the validation set among all runs and parametrizations. The ratios used for parametrizations are written as subscripts.

We select the model with the best SARI on the validation set and report its score on the test set. This model uses three control tokens out of four: $\text{NbChars}_{0.95}$, $\text{LevSim}_{0.75}$ and $\text{WordRank}_{0.75}$ (optimal target ratios in subscript).

ACCESS scores best on SARI (41.87), a significant improvement over previous state of the art (40.45), and third to best FKGL (7.22). The second and third models in terms of SARI, DMASS+DCSS (40.45) and SBMT+PPDB+SARI (39.96), both use the external resource Simple PPDB [Pavlick and Callison-Burch, 2016] that was extracted from 1000 times more data than what we used for training. Our FKGL is also better (lower) than these methods. The Hybrid model scores best on FKGL (4.56) i.e. they generated the simplest (and shortest) sentences, but it was done at the expense of SARI (31.40).

Parametrization encourages the model to rely on explicit aspects of the simplification process, and to associate them with the control tokens. The model can then be adapted more precisely to the type of simplification needed. In WikiLarge, for instance, the compression ratio distribution is different than that of human simplifications (see Figure 9.1). The NbChars control token helps the model decorrelate the compression aspect from other attributes of the simplification process. This control token is then adapted to the amount of compression required in a given evaluation dataset, such as a true, human simplified Sentence Simplification dataset. Our best model indeed worked best with a NbChars target ratio set to 0.95 which is the closest bucketed value to the compression ratio of human

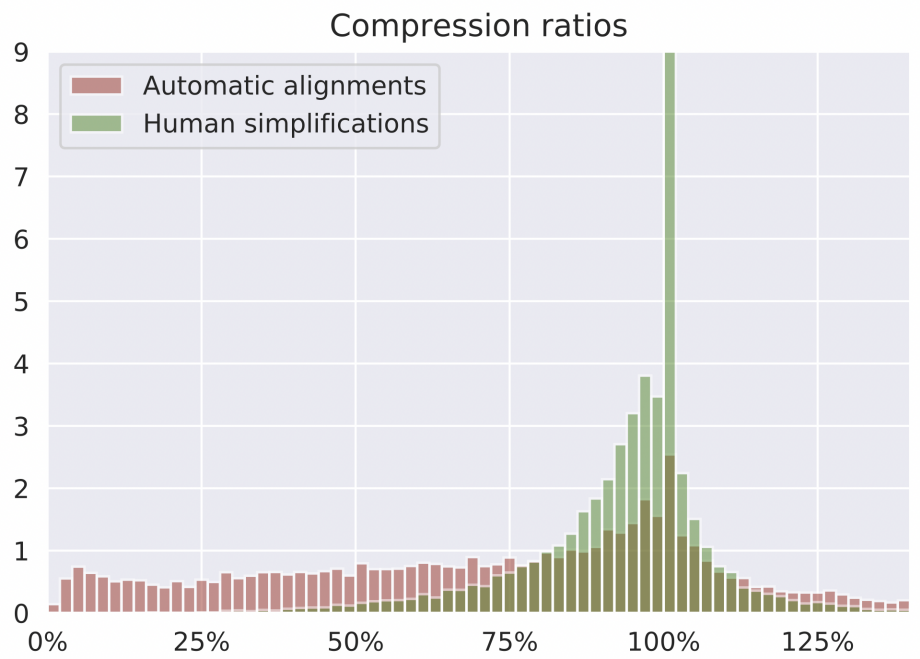


Figure 9.1: Density distribution of the **compression ratios** between the source sentence and the target sentence. The automatically aligned pairs from WikiLarge train set are spread (red) while human simplifications from the validation and test set (green) are gathered together with a mean ratio of 0.93 (i.e. nearly no compression).

annotators on the WikiLarge validation set (0.93).

9.4 Ablation Studies

In this section we investigate the contribution of each control token to the final SARI score of ACCESS. Table 9.3 reports scores of models trained with different combinations of control tokens on the WikiLarge validation set (2000 source sentences, with 8 human simplifications each). We combined control tokens using greedy forward selection; at each step, we add the control token leading to the best performance when combined with previously added control tokens.

With only one control token, WordRank proves to be best (+2.28 SARI over models without parametrization). As the WikiLarge validation set mostly contains small paraphrases, it seems natural that the control token linked to lexical simplification increases the performance the most.

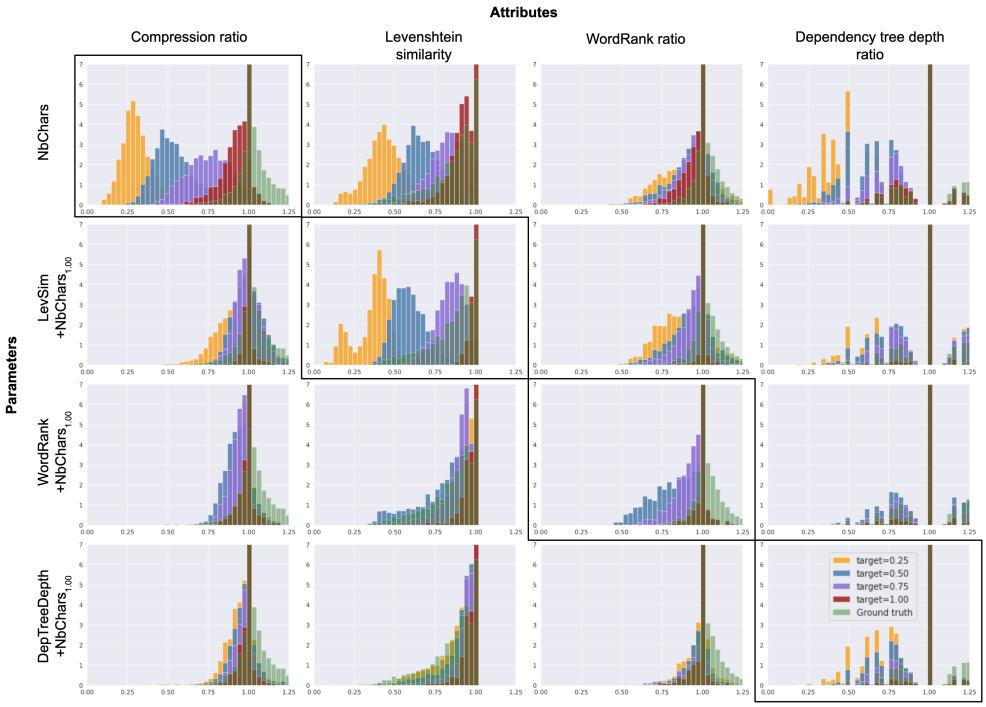
WikiLarge **(validation)**	SARI ↑	FKGL ↓
Transformer	37.06 ± 0.25	7.66 ± 0.42
+DepTreeDepth	$37.72^{*} \pm 0.18$	7.64 ± 0.22
+NbChars	$37.94^{*} \pm 0.09$	7.87 ± 0.15
+LevSim	$38.29^{*} \pm 0.66$	7.53 ± 0.21
+WordRank	$39.35^{*} \pm 0.25$	7.61 ± 0.19
+WordRank+LevSim	$41.1^{*} \pm 0.14$	$6.86^{*} \pm 0.17$
+WordRank+LevSim +NbChars	$\mathbf{41.29}^{*} \pm 0.27$	$7.25^{*} \pm 0.26$
all	$41.03^{*} \pm 0.39$	$\mathbf{6.72}^{*} \pm 0.39$

Table 9.3: Ablation study on the control tokens using greedy forward selection. We report SARI and FKGL on WikiLarge **validation** set. Each score is a mean over 10 runs with a 95% confidence interval. Scores with * are statistically significantly better than the Transformer baseline (p-value < 0.01 for a Student's T-test).

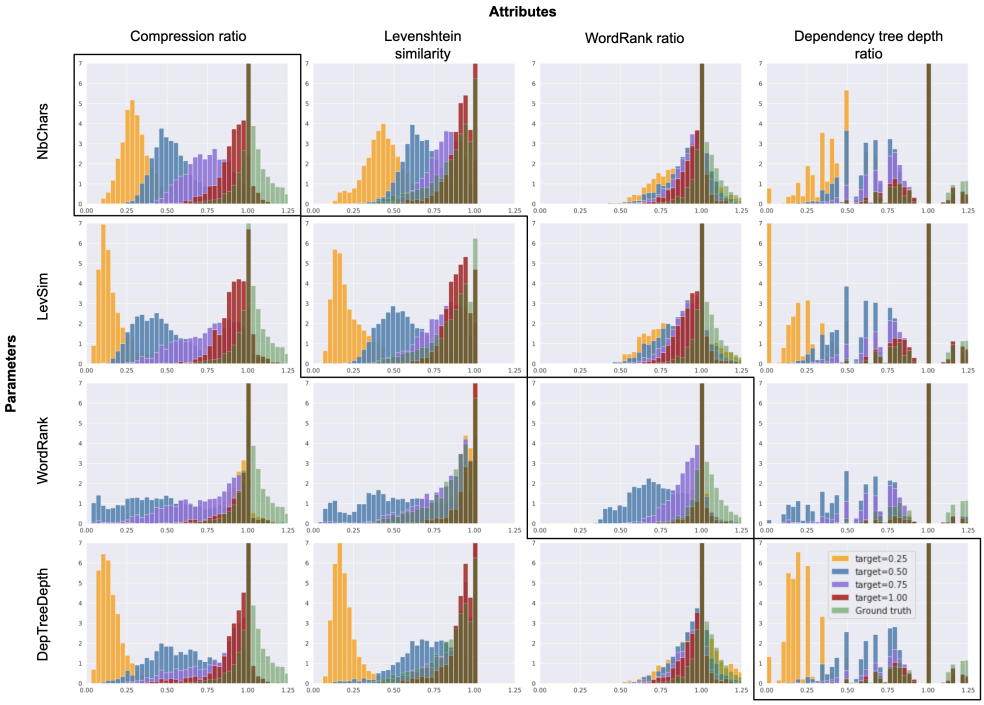
LevSim (+1.23) is the second best control token. This confirms the intuition that hypotheses that are more dissimilar to the source are better simplifications, as claimed in [Wubben et al., 2012, Nisioi et al., 2017].

There is little content reduction in the WikiLarge validation set (see Figure 9.1), thus control tokens that are closely related to sentence length will be less effective. This is the case for the NbChars and DepTreeDepth control tokens (shorter sentences, will have lower tree depths): they bring more modest improvements, +0.88 and +0.66.

The performance boost is nearly additive at first when adding more control tokens (WordRank+LevSim: +4.04) but saturates quickly with 3+ control tokens. In fact, no combination of 3 or more control tokens gets a statistically significant improvement over the WordRank+LevSim setup (p-value < 0.01 for a Student's T-test). This indicates that control tokens are not all useful to improve the scores on this benchmark, and that they might be not independent from one another. The addition of the DepTreeDepth as a final control token even decreases the SARI score slightly, most probably because the considered validation set does not include sentence splitting and structural modifications.



(a) With the $\text{NbChars}_{1.00}$ constraint.



(b) Without the $\text{NbChars}_{1.00}$ constraint.

Figure 9.2: Influence of each control token on the corresponding attributes of the output simplifications. **Rows represent control tokens** (each model is trained either only with one control token or with one control token and the $\text{NbChars}_{1.00}$ constraint), **columns represent output attributes** of the predictions and **colors represent the fixed target ratio** of the control token (yellow=0.25, blue=0.50, violet=0.75, red=1.00, green=Ground truth). We plot the results on the 2000 validation sentences. Figure 9.2a uses the $\text{NbChars}_{1.00}$ constraint, whereas Figure 9.2b does not.

Target control tokens	Sentence
Source	Some trails are designated as nature trails , and are used by people learning about the natural world .
NbChars$_{1.00}$	Some trails are **called** nature trails , and are used by people about the natural world .
NbChars$_{0.75}$	Some trails are **called** nature trails , and are used by people about the natural world .
NbChars$_{0.50}$	Some trails are used by people about the natural world .
NbChars$_{0.25}$	Some trails are used by people .
LevSim$_{1.00}$+NbChars$_{1.00}$	Some trails are designated as nature trails , and are used by people learning about the natural world .
LevSim$_{0.75}$+NbChars$_{1.00}$	Some trails are **made for** nature trails **. They** are used by people **who learn** about the natural world .
LevSim$_{0.50}$+NbChars$_{1.00}$	**The** trails **that** are used by people learning about the natural world **, because the trails are good trails** .
LevSim$_{0.25}$+NbChars$_{1.00}$	**Mechanical** trails **(also known as " trail trail " or " trails ")** are trails **that** are used **for trails** .
WordRank$_{1.00}$+NbChars$_{1.00}$	Some trails are designated as nature trails , and are used by people learning about the natural world .
WordRank$_{0.75}$+NbChars$_{1.00}$	Some trails are **called** nature trails , and are used by people learning about the natural world .
WordRank$_{0.50}$+NbChars$_{1.00}$	Some trails are **known** as nature trails , and are used by people **as well as by people who are in** the world .
WordRank$_{0.25}$+NbChars$_{1.00}$	Some trails are **also called** nature trails , and are used by people learning about the natural world .
DepTreeDepth$_{1.00}$+NbChars$_{1.00}$	Some trails are designated as nature trails , and are used by people learning about the natural world .
DepTreeDepth$_{0.75}$+NbChars$_{1.00}$	Some trails are designated as nature trails **. They** are used by people learning about the natural world .
DepTreeDepth$_{0.50}$+NbChars$_{1.00}$	Some trails are designated as nature trails **. They** are used by people learning about the natural world .
DepTreeDepth$_{0.25}$+NbChars$_{1.00}$	Some trails are designated as nature trails **. They** are used by people **to learn** about the natural world .

Source	Iron Maiden , released on April 14 , 1980 , is the debut album by heavy metal band Iron Maiden .
NbChars$_{1.00}$	Iron Maiden , released on April 14 , 1980 , is the debut album by heavy metal band Iron Maiden .
NbChars$_{0.75}$	Iron Maiden is the debut album by heavy metal band Iron Maiden .
NbChars$_{0.50}$	Iron Maiden , released on April 14 , 1980 .
NbChars$_{0.25}$	Iron Maiden **was** released on April 14 , 1980 .
LevSim$_{1.00}$+NbChars$_{1.00}$	Iron Maiden , released on April 14 , 1980 , is the debut album by heavy metal band Iron Maiden .
LevSim$_{0.75}$+NbChars$_{1.00}$	Iron Maiden is the debut album by heavy metal band Iron Maiden . **It was released on April 14 , 1980 .**
LevSim$_{0.50}$+NbChars$_{1.00}$	Iron Maiden is the debut album by heavy metal band Iron Maiden . **It was released on April 14 , 1980 .**
LevSim$_{0.25}$+NbChars$_{1.00}$	Iron Maiden is **an** album by **American** heavy metal band Iron Maiden . **It was released on April 14 , 1980 .**
WordRank$_{1.00}$+NbChars$_{1.00}$	Iron Maiden is the **first** album **released** by heavy metal band Iron Maiden . **It was released in 1980 .**
WordRank$_{0.75}$+NbChars$_{1.00}$	Iron Maiden **is a first album by the band Iron Maiden . It was** released on April 14 , 1980 .
WordRank$_{0.50}$+NbChars$_{1.00}$	Iron Maiden is **a city of** the **state of Arkansas in the United States of America** .
WordRank$_{0.25}$+NbChars$_{1.00}$	Iron Maiden **is a first album** released **by the band Iron Maiden . It was released** on April 14 , 1980 .
DepTreeDepth$_{1.00}$+NbChars$_{1.00}$	Iron Maiden , released on April 14 , 1980 , is the **first** album by heavy metal band Iron Maiden .
DepTreeDepth$_{0.75}$+NbChars$_{1.00}$	Iron Maiden is **a first** album by **British** heavy metal band Iron Maiden . **It was released on April 14 , 1980 .**
DepTreeDepth$_{0.50}$+NbChars$_{1.00}$	Iron Maiden is **an** album by **British** heavy metal band Iron Maiden . **It was released on April 14 , 1980 .**
DepTreeDepth$_{0.25}$+NbChars$_{1.00}$	Iron Maiden **was** released on April 14 , 1980 **. It was released in** Iron Maiden **on April 14 , 1980** .

Source	Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy .
NbChars$_{1.00}$	Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy .
NbChars$_{0.75}$	Nocturnes is an orchestral composition in three movements by the French composer Claude **Debus** .
NbChars$_{0.50}$	Nocturnes is an orchestral composition in three movements .
NbChars$_{0.25}$	Nocturnes is an orchestral composition .
LevSim$_{1.00}$+NbChars$_{1.00}$	Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy .
LevSim$_{0.75}$+NbChars$_{1.00}$	Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy .
LevSim$_{0.50}$+NbChars$_{1.00}$	Nocturnes is **a piece of music for orchestra that was composed** by **a** French composer **called** Claude Debussy .
LevSim$_{0.25}$+NbChars$_{1.00}$	**Claude Debussy was a** French composer **who wrote music for the orchestra when he was 17 years old** .
WordRank$_{1.00}$+NbChars$_{1.00}$	Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy .
WordRank$_{0.75}$+NbChars$_{1.00}$	Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy .
WordRank$_{0.50}$+NbChars$_{1.00}$	Nocturnes is **a piece of music** by the French composer Claude Debussy .
WordRank$_{0.25}$+NbChars$_{1.00}$	Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy .
DepTreeDepth$_{1.00}$+NbChars$_{1.00}$	Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy .
DepTreeDepth$_{0.75}$+NbChars$_{1.00}$	Nocturnes is an orchestral composition in three movements by the French composer Claude Debussy .
DepTreeDepth$_{0.50}$+NbChars$_{1.00}$	Nocturnes is an orchestral composition by the French composer Claude Debussy **in three movements** .
DepTreeDepth$_{0.25}$+NbChars$_{1.00}$	Nocturnes is **a French orchestra . It was started by Claude Debussy** in three movements .

Source	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
NbChars$_{0.75}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) .
NbChars$_{0.50}$	**This** means that it is very dark in colouring (darker than soot) .
NbChars$_{0.25}$	It is an F-type asteroid .
LevSim$_{1.00}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
LevSim$_{0.75}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) **made up of carbonate metal** .
LevSim$_{0.50}$+NbChars$_{1.00}$	F-type **asteroids can be made up of darker than soot (darker than soot)** , **or darker** (darker than soot) **, or dark (darker)** .
LevSim$_{0.25}$+NbChars$_{1.00}$	**IAUC 2003 September 6 (naming the moon) was discovered by Eros in 2005 by E. H. E. E. J. E. J. J. J. J. J. J. J. R. J.** [...]
WordRank$_{1.00}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
WordRank$_{0.75}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a **made of carbonate** .
WordRank$_{0.50}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a **very dark made up of** .
WordRank$_{0.25}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
DepTreeDepth$_{1.00}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
DepTreeDepth$_{0.75}$+NbChars$_{1.00}$	It is an F-type asteroid , which means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
DepTreeDepth$_{0.50}$+NbChars$_{1.00}$	It is an F-type asteroid **. It** means that it is very dark in colouring (darker than soot) with a carbonaceous composition .
DepTreeDepth$_{0.25}$+NbChars$_{1.00}$	It is an F-type asteroid **. It** means that it is very dark in colouring (darker than soot) with a carbonaceous composition .

Table 9.4: Influence of control tokens on example sentences. Each source sentence is simplified with models trained with each of the four control tokens with varying target ratios; modified words are in bold. The NbChars$_{1.00}$ constraint is added for LevSim, WordRank and DepTreeDepth.

9.5 Analysis of the Influence of Control Tokens

Our goal is to give the user control over how the model will simplify sentences on four important attributes of Sentence Simplification: length, paraphrasing, lexical complexity and syntactic complexity. To this end, we introduced four control tokens: NbChars, LevSim, WordRank and DepTreeDepth. Even though the control tokens improve the performance in terms of SARI, it is not sure whether they have the desired effect on their associated attribute. In this section we investigate to what extent each control token controls the generated simplification. We first used separate models, each trained with a single control token to isolate their respective influence on the output simplifications. However, we witnessed that with only one control token, the effect of LevSim, WordRank and DepTreeDepth was mainly to reduce the length of the sentence (Figure 9.2b). Indeed, shortening the sentence will decrease the Levenshtein similarity, decrease the WordRank (when complex words are deleted) and decrease the dependency tree depth (shorter sentences have shallower dependency trees). Therefore, to clearly study the influence of those control tokens, we also add the NbChars control token during training, and set its ratio to 1.00 at inference time, as a constraint toward not modifying the length.

Figure 9.2a highlights the cross influence of each of the four control tokens on their four associated attributes. Control tokens are successively set to ratios of 0.25 (yellow), 0.50 (blue), 0.75 (violet) and 1.00 (red); the ground truth is displayed in green. Plots located on the diagonal show that control tokens control their respective attributes (e.g. NbChars affects the compression ratio), although not with the same effectiveness.

The histogram located at (row 1, col 1) shows the effect of the NbChars control token on the compression ratio of the predicted simplifications. The resulting distributions are centered on the 0.25, 0.5, 0.75 and 1 target ratios as expected, and with little overlap. This indicates that the lengths of predictions closely follow what is asked of the model. Table 9.4 illustrates this with an example. The NbChars control token affects Levenshtein similarity: reducing the length decreases the Levenshtein similarity. Finally, NbChars has a marginal impact on the WordRank ratio distribution, but clearly influences the dependency tree depth. This is natural considered that the depth of a dependency tree is very correlated with the

length of the sentence.

The LevSim control token also has a clear cut impact on the Levenshtein similarity (row 2, col 2). The first example in Table 9.4 highlights that LevSim increases the amount of paraphrasing in the simplifications. With an extreme target ratio of 0.25, the model outputs ungrammatical and meaningless predictions, thus indicating that the choice of a target ratio is important for generating proper simplifications.

WordRank and DepTreeDepth do not seem to control their respective attribute as well as NbChars and LevSim according to Figure 9.2a. However we witness more lexical simplifications when using the WordRank ratio than with other control tokens. In Table 9.4's first example, "designated as" is simplified by "called" or "known as" with the WordRank control token. Equivalently, DepTreeDepth splits the source sentence in multiple shorter sentences in Table 9.4's first example. WordRank and DepTreeDepth control tokens therefore have the desired effect.

9.6 Summary and Final Remarks

This chapter showed that explicitly conditioning Seq2Seq models on control tokens such as length, paraphrasing, lexical complexity or syntactic complexity increases their performance significantly for sentence simplification. We confirmed through an analysis that each control token has the desired effect on the generated simplifications. In addition to being easy to extend to other attributes of Sentence Simplification, our method paves the way toward adapting the simplification to audiences with different needs.

Part IV

Extending Sentence Simplification to Other Languages

In this chapter and the next, we extend our methods to languages other than English. First we propose a method to mine training data for simplification models in any languag using semantic sentence embeddings. This paraphrase data is then used to train controllable sentence simplification models with strong performance. In Chapter 11, we study the pretraining of language models in French and use our findings to obtain even stronger simplification models in French.

Chapter 10

MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases

Research has mostly focused on English simplification, where source texts and associated simplified texts exist and can be automatically aligned, such as English Wikipedia and Simple English Wikipedia [Zhang and Lapata, 2017]. This is indeed the case of our proposed supervised method ACCESS in Chapter 9. However, such data is limited in terms of size and domain, and difficult to find in other languages. Additionally, simplifying a sentence can be achieved in multiple ways, and depend on the target audience. Simplification guidelines are not uniquely defined, outlined by the stark differences in English simplification benchmarks (Chapter 7). This highlights the need for more general models that can adjust to different simplification contexts and scenarios.

In this chapter[1], we propose to train controllable models using sentence-level paraphrase data only, i.e. parallel sentences that have the same meaning but phrased differently. In order to generate simplifications and not paraphrases at test time, we use ACCESS (Chapter 9) to control attributes such as length, lexical and syntactic complexity. Paraphrase data is more readily available, and opens the door to training flexible models that can adjust to more varied simplification scenarios. Our original goal was to mine simplifications from the

[1]This chapter is an adapted version of [Martin et al., 2020b].

web, but we surprisingly discovered that mining paraphrases leads to controllable models with better simplification performance while being more straightforward and requiring less prior assumptions (cf. Section 10.4.5). We propose to gather such paraphrase data in any language by mining sentences from Common Crawl using semantic sentence embeddings. Simplification models trained on mined paraphrase data actually proves to work as well as models trained on large existing English paraphrase corpora (cf. Section 10.4.5).

Our resulting Multilingual Unsupervised Sentence Simplification method, MUSS, is *unsupervised* because it can be trained without relying on *labeled* simplification data,[2] even though we mine using supervised sentence embeddings.[3] We apply MUSS on English, French, and Spanish to closely match or outperform the supervised state of the art in all languages. MUSS further improves the state of the art on all English datasets by incorporating additional labeled simplification data. We make the following contributions:

- We introduce a novel approach to training simplification models with paraphrase data only and propose a mining procedure to create large paraphrase corpora for any language.
- Our approach obtains strong performance. Without any labeled simplification data, we match or outperform the supervised state of the art in English, French and Spanish. We further improve the English state of the art by incorporating labeled simplification data.
- We release pretrained models, paraphrase data, and code for mining and training.[4]

10.1 Related work

Data-driven methods have been predominant in **English sentence simplification** in recent years [Alva-Manchego et al., 2020], requiring large supervised training corpora of

[2]We use the term *labeled simplifications* to refer to parallel datasets where texts were manually simplified by humans.

[3]Previous works have also used the term *unsupervised simplification* to describe works that do not use any labeled parallel simplification data while leveraging supervised components such as constituency parsers and knowledge bases [Kumar et al., 2020], external synonymy lexicons [Surya et al., 2019], and databases of simplified synonyms [Zhao et al., 2020]. We shall come back to these works in Section 10.1.

[4]`https://github.com/facebookresearch/muss`

complex-simple aligned sentences [Wubben et al., 2012, Xu et al., 2016, Zhang and Lapata, 2017, Zhao et al., 2018]. Methods have automatically aligned English and Simple English Wikipedia articles [Zhu et al., 2010, Coster and Kauchak, 2011b, Woodsend and Lapata, 2011, Kauchak, 2013, Zhang and Lapata, 2017]. Professional quality datasets such as NEWSELA [Xu et al., 2015] exist, but they are rare and come with restrictive licenses that hinder reproducibility and widespread usage.

Simplification in other languages has been explored in Brazilian Portuguese [Aluísio et al., 2008], Spanish [Saggion et al., 2015, Štajner et al., 2015b], Italian [Brunato et al., 2015, Tonelli et al., 2016], Japanese [Goto et al., 2015, Kajiwara and Komachi, 2018, Katsuta and Yamamoto, 2019], and French [Gala et al., 2020]. The lack of large parallel corpora has slowed research down. In this work, we show that a method trained on mined data can reach state-of-the-art results in each language.

Previous work on **parallel dataset mining** have been used mostly in machine translation using document retrieval [Munteanu and Marcu, 2005], language models [Koehn et al., 2018, 2019], and embedding space alignment [Artetxe and Schwenk, 2019b] to create large corpora [Tiedemann, 2012b, Schwenk et al., 2019]. We focus on paraphrasing for sentence simplifications, which presents new challenges. Unlike machine translation, where the same sentence should be identified in two languages, we develop a method to identify varied paraphrases of sentences, that have a wider array of surface forms, including different lengths, multiple sentences, different vocabulary usage, and removal of content from more complex sentences.

Previous **unsupervised paraphrasing** research has aligned sentences from various parallel corpora [Barzilay and Lee, 2003] with multiple objective functions [Liu et al., 2020a]. Bilingual pivoting relied on MT datasets to create large databases of word-level paraphrases [Pavlick et al., 2015], lexical simplifications [Pavlick and Callison-Burch, 2016, Kriz et al., 2018], or sentence-level paraphrase corpora [Wieting and Gimpel, 2018]. This has not been applied to multiple languages or to sentence-level simplification. Additionally, we use raw monolingual data to create our paraphrase corpora instead of relying on parallel MT datasets.

	Type	# Sequence Pairs	# Avg. Tokens per Sequence
WIKILARGE (English)	Labeled Parallel Simplifications	296,402	original: 21.7 simple: 16.0
NEWSELA (English)	Labeled Parallel Simplifications	94,206	original: 23.4 simple: 14.2
English	Mined	1,194,945	22.3
French	Mined	1,360,422	18.7
Spanish	Mined	996,609	22.8

Table 10.1: Statistics on our mined paraphrase training corpora compared to standard simplification datasets (see section 10.3.3 for more details).

10.2 Method

We now describe MUSS, our approach to training controllable simplification models on mined data.

10.2.1 Mining Paraphrases in Many Languages

Extracting Sequences Simplification consists of multiple rewriting operations, some of which span multiple sentences (e.g. sentence splitting or fusion). To allow such operations to be represented in our mined data, we extract chunks of text composed of multiple sentences that we call *sequences*.

We extract such sequences by first tokenizing a document into individual sentences $\{s_1, s_2, \ldots, s_n\}$ using NLTK [Bird and Loper, 2004]. We then extract sequences of adjacent sentences with maximum length of 300 characters: $\{[s_1], [s_1, s_2], [s_1, \ldots, s_k], [s_2], [s_2, s_3], ...\}$. Noisy sequences are filtered out when they have more than 10% punctuation characters and when they have low language model probability according to a 3-gram language model trained with `kenlm` [Heafield, 2011] on Wikipedia.

Source texts are taken from CCNET [Wenzek et al., 2020], an extraction of Common Crawl (snapshot of the web). We only consider documents from the HEAD split in CCNET—this represents the third of the data with the best perplexity using a language model. For English and French, we extract 1 billion sequences. For Spanish we extract 650 millions sequences, the maximum for this language in CCNET after filtering out noisy text.

Creating a Sequence Index Using Embeddings To automatically mine our paraphrase corpora, we first compute n-dimensional embeddings for each extracted sequence using LASER [Artetxe and Schwenk, 2019b]. LASER provides joint multilingual sentence embeddings in 93 languages that have been successfully applied to the task of bilingual bitext mining [Schwenk et al., 2019]. In this work, we show that LASER can also be used to mine monolingual paraphrase datasets but also highlights its limits (cf. Section 10.4.4). For each language we then index embeddings for each sequence using `faiss` for fast nearest neighbor search. We compute LASER embeddings of dimension 1024 and reduce dimensionality with a 512 PCA followed by random rotation. We further compress them using 8 bit scalar quantization. The compressed embeddings are then stored in a `faiss` inverted file index with 32,768 cells (nprobe=16). These embeddings are used to mine pairs of paraphrases.

Mining Paraphrases We use each sequence as a query q_i against the billion-scale `faiss` index to retrieve the top-8 nearest neighbor in the LASER embedding space (L2 distance). We then use an upper bound on L2 distance (0.05) and a margin criterion following [Artetxe and Schwenk, 2019a] to filter out nearest neighbors with low similarity (relative distance compared to other top-8 nearest neighbors lower than 0.6). The remaining nearest neighbors constitute a set of candidate aligned paraphrases to the query sequence: $\{(q_i, c_{i,1}), \ldots, (q_i, c_{i,k})\}$. We finally filter out poor alignments and remove almost identical paraphrases by enforcing a case-insensitive character-level Levenshtein distance [Levenshtein, 1966] greater or equal to 20%. We remove paraphrases that come from the same document to avoid aligning sequences that overlapped each other in the text. We also remove paraphrases where one of the sequence is contained in the other. We further filter out any sequence that is present in our evaluation datasets.

We report statistics of the mined corpora in English, French and Spanish in Table 10.1, examples of mined paraphrases in Table 10.2, and limits of this mining method in Section 10.4.4. Models trained on these mined paraphrases obtain similar performance than models trained on existing paraphrase datasets (cf. Section 10.4.5).

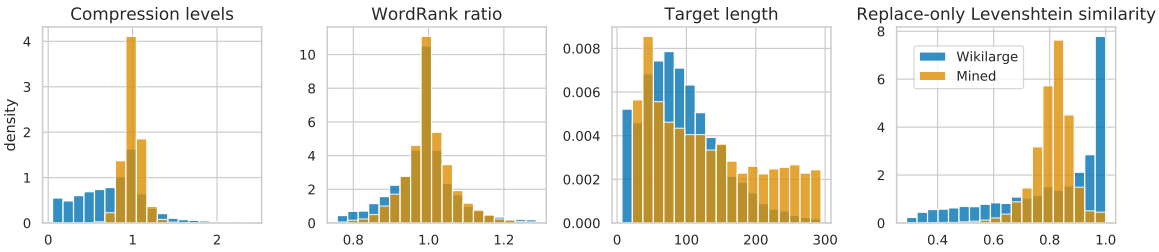


Figure 10.1: Density of several text features in WIKILARGE and our mined data. The WordRank ratio is a measure of lexical complexity reduction (Chapter 9). Replace-only Levenshtein similarity only considers replace operations in the traditional Levenshtein similarity and assigns 0 weights to insertions and deletions.

Characteristics of the mined data We show in Figure 10.1 the distribution of different surface features of our mined data versus those of WIKILARGE. Some examples of mined paraphrases are shown in Table 10.2.

10.2.2 Simplifying with ACCESS

In this section we describe how we adapt ACCESS (Chapter 9) to train controllable sequence-to-sequence models on mined paraphrases, instead of labeled parallel simplifications. ACCESS is a method to make any sequence-to-sequence model controllable by conditioning on simplification-specific control tokens.

Training with Control Tokens At training time, the model is provided with control tokens that give oracle information on the target sequence, such as the amount of compression between the target and the source (length control). For example, when the target sequence is 80% of the length of the source sequence, the control token <NumChars_80%> is provided. At inference time generation can be controlled by selecting a given target control value. We adapt the original Levenshtein similarity control to only consider replace operations but otherwise use the same controls as in Chapter 9. The controls used are: character length ratio, *replace-only*[5] Levenshtein similarity, aggregated word frequency ratio, and dependency tree depth ratio. We thus prepend to every source in the training set the following 4 control tokens

[5]We modify the Levenshtein similarity parameter to only consider replace operations, by assigning a 0 weight to insertions and deletions. This change helps decorrelate the Levenshtein similarity control token from the length control token and produced better results in preliminary experiments.

Query	**For insulation, it uses foam-injected** polyurethane which helps **ensure** the quality of the ice **produced by the machine. It comes with** an easy to clean air filter.
Mined	**It has** polyurethane **for insulation** which **is foam-injected. This** helps **to maintain** the quality of the ice **it produces. The unit has** an easy to clean air filter.
Query	Here are some **useful** tips and **tricks** to **identify and** manage your **stress.**
Mined	Here are some tips and **remedies you can follow** to manage **and control** your **anxiety.**
Query	**As cancer** cells **break apart,** their contents are **released** into the **blood.**
Mined	**When brain** cells **die,** their contents are **partially spilled back** into the **blood in the form of debris.**
Query	**The trail** is ideal for **taking** a short **hike with small children** or a **longer, more rugged overnight trip.**
Mined	**It** is **the** ideal **location** for a short **stroll, a nature walk** or a **longer walk.**
Query	Thank you for **joining us, and please** check **out** the **site.**
Mined	Thank you for **calling us. Please** check the **website.**

Table 10.2: **Examples of Mined Paraphrases**. Paraphrases, although sometimes not preserving the entire meaning, display various rewriting operations, such as lexical substitution, compression or sentence splitting.

with sample-specific values: <NumChars_XX%> <LevSim_YY%> <WordFreq_ZZ%> <DepTreeDepth_TT%>.

Selecting Control Values at Inference After training with oracle controls, we can adjust the controls at inference to obtain the desired type of simplifications. Sentence simplification indeed depends on the context and target audience: shorter sentences are more adapted to people with cognitive disabilities, while using more frequent words is useful to second language learners. It is important that supervised and unsupervised simplification systems can be adapted to different conditions: [Kumar et al., 2020] choose operation-specific weights of their unsupervised simplification model for each evaluation set and [Surya et al., 2019] select different models using SARI on each validation set. Similarly, we set the 4 control hyper-parameters of ACCESS using SARI on each validation set and keep them fixed for all samples in the test set. As mentioned in Section "Simplifying with ACCESS", we select the 4 ACCESS hyper-parameters using SARI on the validation set. We use zero-order optimization with the NEVERGRAD library [Rapin and Teytaud, 2018]. We use the OnePlusOne optimizer with a budget of 64 evaluations (approximately 1 hour of

optimization on a single GPU). The hyper-parameters are contained in the $[0.2, 1.5]$ interval. The 4 hyper-parameter values are then kept fixed for all sentences in the associated test set.

These 4 control hyper-parameters are intuitive and easy to interpret: when no validation set is available, they can also be set using prior knowledge on the task and still lead to solid performance (cf. Section 10.4.5).

10.2.3 Leveraging Unsupervised Pretraining

We combine our controllable models with unsupervised pretraining. For English, we finetune the pretrained generative model BART [Lewis et al., 2020] with ACCESS control tokens on our newly created training corpora. BART is a pretrained sequence-to-sequence model that generalizes other recent pretrained methods such as BERT [Devlin et al., 2019] for encoder-decoder models. For non-English, we use its multilingual version MBART [Liu et al., 2020b], pretrained on 25 languages.

10.3 Experimental Setting

We assess the performance of our approach on three languages: English, French, and Spanish. We implement our models with `fairseq` [Ott et al., 2019]. All our models are Transformers [Vaswani et al., 2017] based on the $\text{BART}_{\text{Large}}$ architecture (388M parameters), keeping the optimization procedure and hyper-parameters fixed to those used in the original implementation [Lewis et al., 2020][6]. We either randomly initialize weights for the standard sequence-to-sequence experiments or initialize with pretrained BART for the BART experiments. When initializing the weights randomly, we use a learning rate of 3.10^{-4} versus the original 3.10^{-5} when finetuning BART. For a given seed, the model is trained on 8 Nvidia V100 GPUs during approximately 10 hours.

In all our experiments, we report scores on the test sets averaged over 5 random seeds with 95% confidence intervals.

[6]All hyper-parameters and training commands for `fairseq` can be found here: `https://github.com/pytorch/fairseq/blob/master/examples/bart/README.summarization.md`

	ASSET (en)		TURKCORPUS (en)		NEWSELA (en)	
	SARI ↑	FKGL ↓	SARI ↑	FKGL ↓	SARI ↑	FKGL ↓
Baselines and Gold Reference						
Gold Reference	44.87±0.36	6.49±0.15	40.04±0.30	8.77±0.08	—	—
Unsupervised Systems						
BTRLTS [Zhao et al., 2020]	33.95	7.59	33.09	8.39	37.22	3.80
UNTS [Surya et al., 2019]	35.19	7.60	36.29	7.60	—	—
RM+EX+LS+RO [Kumar et al., 2020]	36.67	7.33	37.27	7.33	**38.33**	2.98
MUSS (mined data only)	**42.65**±0.23	8.23±0.62	**40.85**±0.15	8.79±0.30	**38.09**±0.59	5.12±0.47
Supervised Systems						
EditNTS [Dong et al., 2019]	34.95	8.38	37.66	8.38	39.30	3.90
DMASS-DCSS [Zhao et al., 2018]	38.67	7.73	39.92	7.73	—	—
ACCESS(Chapter 10.2.2)	40.13	7.29	41.38	7.29	—	—
MUSS (labeled data only)	**43.63**±0.71	6.25±0.42	**42.62**±0.27	6.98±0.95	**42.59**±1.00	2.74±0.98
MUSS (labeled + mined data)	**44.15**±0.56	6.05±0.51	**42.53**±0.36	7.60±1.06	**41.17**±0.95	2.70±1.00

Table 10.3: **Unsupervised and Supervised Sentence Simplification for English.** We display SARI and FKGL on ASSET, TURKCORPUS and NEWSELA test sets for English. Supervised models are trained on WIKILARGE for the first two test sets, and NEWSELA for the last. Best SARI scores within confidence intervals are in bold.

10.3.1 Baselines

In addition to comparisons with previous works, we implement multiple baselines to assess the performance of our models, especially for French and Spanish where no previous simplification systems have open-source implementations.

Identity The entire original sequence is kept unchanged and used as the simplification.

Truncation The original sequence is truncated to the first 80% words. It is a strong baseline according to standard simplification metrics.

Pivot We use machine translation to use English models in other languages. The source non-English sentence is translated to English, simplified with our best supervised English simplification system, and then translated back into the source language. For French and Spanish translation, we use CCMATRIX [Schwenk et al., 2019] to train Transformer models with 240 million parameters with LayerDrop [Fan et al., 2019]. We train for 36 hours on 8

Baselines	**ALECTOR (fr)** SARI ↑	**NEWSELA (es)** SARI ↑
Identity	26.16	16.99
Truncate	33.44	27.34
Pivot	33.48±0.37	**36.19**±0.34
MUSS†	**41.73**±0.67	**35.67**±0.46

Table 10.4: **Unsupervised Sentence Simplification in French and Spanish.** We display SARI scores in French (ALECTOR) and Spanish (NEWSELA). Best SARI scores within confidence intervals are in bold. †MBART+ACCESS model.

GPUs following the suggested parameters in Ott et al. [2019]. We use MUSS trained on mined data + WIKILARGE as the English simplification model.

Gold Reference We report gold reference scores for ASSET and TURKCORPUS as multiple references are available. We evaluate each reference against all others in a leave-one-out scenario, and then average the scores.[7]

10.3.2 Evaluation Metrics

We evaluate with the standard metrics SARI[8] and FKGL. FKGL was designed to be used on English texts only, we do not report it on French and Spanish. We do not report BLEU [Papineni et al., 2002] due its dubious suitability for sentence simplification [Sulem et al., 2018a].

10.3.3 Training Data

For all languages we use the mined data described in Table 10.1 as training data. In English we show that training with additional labeled simplification data leads to better performance. We use two labeled datasets: **WIKILARGE** [Zhang and Lapata, 2017] and **NEWSELA** [Xu et al., 2015]. As a reminder to Chapter 2, WIKILARGE is composed of

[7]To avoid creating a discrepancy in terms of number of references between the gold reference scores, where we leave one reference out, and when we evaluate the models with all references, we compensate by duplicating one of the other references at random so that the total number of references is unchanged.

[8]We use the latest version of SARI in `EASSE` which fixes bugs and inconsistencies from the traditional implementation. We recompute scores using previous work's system predictions available in `EASSE`.

296k simplifications automatically aligned from English Wikipedia and Simple English Wikipedia. NEWSELA is a collection of news articles with professional simplifications, aligned into 94k simplifications by Zhang and Lapata [2017].[9]

10.3.4 Evaluation Data

English We evaluate our English models on **ASSET** (Chapter 7), **TURKCORPUS** [Xu et al., 2016] and **NEWSELA** [Xu et al., 2015]. As a reminder to Chapter 7, TURKCORPUS and ASSET were created using the same 2000 valid and 359 test source sentences and they respectively contain 8 and 10 reference simplifications per source sentence. ASSET is a features more varied set of rewriting operations than TURKCORPUS, and is considered simpler by human judges. For NEWSELA, we evaluate on the split from [Zhang and Lapata, 2017], which includes 1129 validation and 1077 test sentence pairs.

French We use the French **ALECTOR** dataset [Gala et al., 2020]. ALECTOR is a collection of literary (tales, stories) and scientific (documentary) texts along with their manual document-level simplified versions. These documents were extracted from material available to French primary school pupils. The ALECTOR corpus comes as source documents and their manual simplifications but not sentence-level alignment is provided. Luckily, most of these documents were simplified line by line, each line consisting of a few sentences. For each source document, we therefore align each line, provided it is not too long (less than 6 sentences), with the most appropriate line in the simplified document, using the LASER embedding space. The resulting alignments are split into validation and test by randomly sampling the documents for the validation (450 sentence pairs) and rest for test (416 sentence pairs).

Spanish We use the **Spanish part of NEWSELA** [Xu et al., 2015]. We use the alignments from [Aprosio et al., 2019], composed of 2794 validation and 2795 test sentence pairs. Even though sentences were aligned using the CATS simplification alignment tool [Štajner et al.,

[9]We experimented with other alignments (wiki-auto and newsela-auto [Jiang et al., 2020]) but with lower performance.

2018], some alignment errors remain and automatic scores should be taken with a pinch of salt.

10.4 Results

10.4.1 English Simplification

We report MUSS automatic scores in Table 10.3. We also compare to other state-of-the-art supervised models: DMASS-DCSS [Zhao et al., 2018], EditNTS [Dong et al., 2019], ACCESS (Chapter 9); and unsupervised models: UNTS [Surya et al., 2019], BTRLTS [Zhao et al., 2020], and RM+EX+LS+RO [Kumar et al., 2020].

MUSS Unsupervised Results On the ASSET benchmark, with no labeled simplification data, MUSS obtains a +5.98 SARI improvement with respect to previous unsupervised methods, and a +2.52 SARI improvement over the state-of-the-art supervised methods. For the TURKCORPUS and NEWSELA datasets, the unsupervised MUSS approach achieves strong results, either outperforming or closely matching unsupervised and supervised previous works.

When incorporating labeled data from WIKILARGE and NEWSELA, MUSS obtains state-of-the-art results on all datasets. Using labeled data along with mined data does not always help compared to training only with labeled data, especially with the NEWSELA training set. NEWSELA is a high quality dataset focused on the specific domain of news articles. It might not benefit from additional lesser quality mined data.

Examples of Simplifications Various examples from our unsupervised system are shown in Table 10.5. Examining the simplifications, we see reduced sentence length, sentence splitting, and simpler vocabulary usage. For example, the words *in the town's western outskirts* is changed into *near the town* and *aerial nests* is simplified into *nests in the air*. We also witnessed errors related factual consistency and especially with respect with named entity hallucination or disappearance which would be an interesting area of improvement for future work.

Original	**History** Landsberg prison, which is **in** the **town's western outskirts,** was **completed** in 1910.
Simplified	**The** Landsberg prison, which is **near** the **town,** was **built** in 1910.
Original	The name "hornet" is used for this and related species **primarily** because **of their habit of making aerial** nests **(similar to** the true hornets) rather than **subterranean nests.**
Simplified	The name "hornet" is used for this and related species because **they make** nests **in** the **air (like the** true hornets) rather than **in the ground.**
Original	Nocturnes is **an orchestral composition in three movements** by the French composer Claude Debussy.
Simplified	Nocturnes is **a piece of music for orchestra** by the French composer Claude Debussy.

Table 10.5: **Examples of Generated Simplifications.** We show simplifications generated by our best unsupervised model: MUSS trained on mined data only. Bold highlights differences between original and simplified.

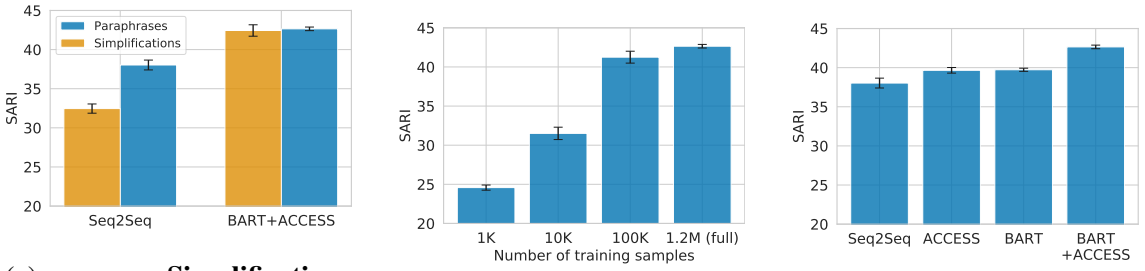
10.4.2 French and Spanish Simplification

Our unsupervised approach to simplification can be applied to any language. Similar to English, we first create a corpus of paraphrases composed of 1.4 million sequence pairs in French and 1.0 million sequence pairs in Spanish (cf. Table 10.1). To incorporate multilingual pretraining, we replace the monolingual BART with MBART, which was trained on 25 languages.

We report the performance of models trained on the mined corpus in Table 10.4. Unlike English, where labeled parallel training data has been created using Simple English Wikipedia, no such datasets exist for French or Spanish. Similarly, no other simplification systems are available in these languages. We thus compare to several baselines, namely the identity, truncation and the strong pivot baseline.

Results MUSS outperforms our strongest baseline by +8.25 SARI for French, while matching the pivot baseline performance for Spanish.

Besides using state-of-the-art machine translation models, the pivot baseline relies on a strong backbone simplification model that has two advantages compared to the French and Spanish simplification model. First the simplification model of the pivot baseline was trained on labeled simplification data from WIKILARGE, which obtains +1.5 SARI in English compared to training only on mined data. Second it uses the stronger monolingual



(a) **Simplifications vs. Paraphrases** (b) **Large-Scale Mining** (c) **BART and ACCESS**

Figure 10.2: **Ablations** We display averaged SARI scores on the English ASSET test set with 95% confidence intervals (5 runs). (a) Models trained on mined simplifications or mined paraphrases, (b) MUSS trained on varying amounts of mined data, (c) Models trained with or without BART and/or ACCESS.

	English			French			Spanish		
	Adequacy	Fluency	Simplicity	Adequacy	Fluency	Simplicity	Adequacy	Fluency	Simplicity
ACCESS (Chapter 9)	3.10±0.32	3.46±0.28	1.40±0.29	—	—	—	—	—	—
Pivot baseline	—	—	—	1.78±0.40	2.10±0.47	1.16±0.31	2.02±0.28	3.48±0.22	2.20±0.29
Gold Reference	**3.71**±0.18	3.78±0.18	1.78±0.30	**3.56**±0.21	**3.92**±0.10	**1.71**±0.32	**3.12**±0.29	**3.52**±0.25	1.70±0.46
MUSS (mined data)	3.20±0.28	3.84±0.14	1.88±0.33	2.88±0.34	3.50±0.32	1.22±0.25	2.26±0.29	3.48±0.25	**2.56**±0.29
MUSS (mined + labeled data)	3.12±0.34	**3.90**±0.14	**2.22**±0.36	—	—	—	—	—	—

Table 10.6: **Human Evaluation** Human ratings of adequacy, fluency and simplicity for ACCESS (Chapter 9), pivot baseline, reference human simplifications, and MUSS. Scores are averaged over 50 ratings per system with 95% confidence intervals.

BART model instead of MBART. In experiments, we noticed that MBART has a small loss in performance of 1.54 SARI compared to its monolingual counterpart BART, due to the fact that it handles 25 languages instead of one. Further improvements could be achieved by using monolingual BART models trained for French or Spanish, possibly outperforming the pivot baseline.

10.4.3 Human Evaluation

To further validate the quality of our models, we conduct a human evaluation in all languages according to adequacy, fluency, and simplicity and report the results in Table 10.6.

Human Ratings Collection For human evaluation, we recruit volunteer native speakers for each language (5 in English, 2 in French, and 2 in Spanish). We evaluate three linguistic aspects on a 5 point Likert scale (0-4): adequacy (*is the meaning preserved?*), fluency (*is*

the simplification fluent?) and simplicity (*is the simplification actually simpler?*). For each system and each language, 50 simplifications are annotated and each simplification is rated once only by a single annotator. The simplifications are taken from ASSET (English), ALECTOR (French), and NEWSELA (Spanish).

Discussion Table 10.6 displays the average ratings along with 95% confidence intervals. Human judgments confirm that our unsupervised and supervised MUSS models are more fluent and produce simpler outputs than previous state-of-the-art ACCESS. They are deemed as fluent and simpler than the human simplifications from ASSET test set, which indicates our model is able to reach a high level of simplicity thanks to the control mechanism. In French and Spanish, our unsupervised model performs better or similar in all aspects than the supervised pivot baseline which has been trained on labeled English simplifications.

10.4.4 Fine-grained Analysis of MUSS Outputs

	Operation-specific SARI (F1 scores) ↑			**Quality Estimation** (%)		
	Additions	Deletions	Keeps	Exact Copies	Compression	Sent. Splits
BTRLTS [Zhao et al., 2020]	1.99	42.09	57.77	19.22	91.72	16.43
UNTS [Surya et al., 2019]	0.83	45.98	58.75	21.45	85.34	1.39
RM+EX+LS+RO [Kumar et al., 2020]	1.29	51.33	57.40	12.81	84.73	2.51
MUSS (mined data only)	8.09±0.74	60.87±0.61	59.00±0.48	0.11±0.19	88.61±7.16	3.45±2.31
EditNTS [Dong et al., 2019]	2.41	42.69	59.73	11.70	83.74	0.00
DMASS-DCSS [Zhao et al., 2018]	4.36	51.37	60.29	5.29	88.96	6.13
ACCESS(Chapter 9)	6.54	50.85	62.99	4.18	94.08	20.89
MUSS (mined + labeled data)	11.14±0.34	60.40±1.64	60.90±1.30	0.11±0.19	88.92±3.34	34.26±12.97

Table 10.7: **Fine-grained Analysis of MUSS** We compare MUSS predictions with other systems on ASSET using the three operation-specific SARI components, % of simplifications which are exact copies of the source, average compression ratios, and % of simplifications with sentence splits.

In table 10.7, we analyse the types of simplifications that MUSS performs using quality estimation features computed with the EASSE library. We decompose the SARI score into its three building blocks: F1 scores accounting for n-gram additions, deletions and keeps.

Copying the Source Over Simplification systems have suffered from not modifying the source sentence enough and often fall back to keeping it entirely unchanged [Wubben et al.,

2012]. MUSS on the other hand almost never resorts to exactly copying the source sentence which leads to higher addition and deletion F1.

Mined Data limits Sentence Splitting MUSS rarely perform sentence splitting when trained on mined data only (3.45% of the time) while it becomes way better at this operation when incorporating labelled data from WIKILARGE (34.26%). Investigating the mined data reveals that our mining approach was not able to mined sentence splitting examples. Our intuition is that this is due to the fact that LASER embeddings do not work well across multiple sentences, thus preventing a single sentences to be matched with multiple corresponding sentences. We identify mining sentence splitting examples as a promising direction of future work.

10.4.5 Ablations

Mining Simplifications vs. Paraphrases In this work, we mined paraphrases to train simplification models. This has the advantage of making fewer assumptions earlier on, by keeping the mining and models as general as possible, so that they are able to adapt to more simplification scenarios.

We also compared to directly mining simplifications using simplification heuristics to make sure that the target side is simpler than the source, following previous work [Kajiwara and Komachi, 2016, Surya et al., 2019]. To mine a simplification dataset, we followed the same paraphrase mining procedure of querying 1 billion sequences on an index of 1 billion sequences. Out of the resulting paraphrases, we kept only pairs that either contained sentence splits, reduced sequence length, or simpler vocabulary (similar to how previous work enforce an FKGL difference). We removed the paraphrase constraint that enforced sentences to be different enough. We tuned these heuristics to optimize SARI on the validation set. The resulting dataset has 2.7 million simplification pairs. In Figure 10.2a, we show that seq2seq models trained on mined paraphrases achieve better performance. A similar trend exists with BART and ACCESS, thus confirming that mining paraphrases can obtain better performance than mining simplifications.

How Much Mined Data Do You Need? We investigate the importance of a scalable mining approach that can create million-sized training corpora for sentence simplification. In Figure 10.2b, we analyze the performance of training our best model on English on different amounts of mined data. By increasing the number of mined pairs, SARI drastically improves, indicating that efficient mining at scale is critical to performance. Unlike human-created training sets, unsupervised mining allows for large datasets in multiple languages.

Improvements from Pretraining and Control We compare the respective influence of pretraining BART and controllable generation ACCESS in Figure 10.2c. While both BART and ACCESS bring improvement over standard sequence-to-sequence, they work best in combination. Unlike previous approaches to text simplification, we use pretraining to train our simplification systems. We find that the main qualitative improvement from pretraining is increased fluency and meaning preservation. For example, in Table 10.10, the model trained only with ACCESS substituted *culturally akin* with *culturally much like*, but when using BART, it is simplified to the more fluent *closely related*. While models trained on mined data see several million sentences, pretraining methods are typically trained on billions. Combining pretraining with controllable simplification enhances simplification performance by flexibly adjusting the type of simplification.

Method	**ASSET** SARI ↑	**TURKCORPUS** SARI ↑	**NEWSELA** SARI ↑
SARI on valid	42.65±0.23	40.85±0.15	38.09±0.59
Approx. value	42.49±0.34	39.57±0.40	36.16±0.35

Table 10.8: **Set ACCESS Controls Wo. Parallel Data**
Setting ACCESS parameters of MUSS +MINED model either using SARI on the validation set or using only 50 *unaligned* sentence pairs from the validation set. All ACCESS parameters are set to the same approximated value: ASSET $= 0.8$, TURKCORPUS $= 0.95$, and NEWSELA $= 0.4$).

Set ACCESS Control Parameters Without Parallel Data In our experiments we adjusted our model to the different dataset conditions by selecting our ACCESS control tokens with SARI on each validation set. When no such parallel validation set exists, we show that

strong performance can still be obtained by using prior knowledge for the given downstream application. This can be done by setting all 4 ACCESS control hyper-parameters to an intuitive guess of the desired compression ratio.

To illustrate this for the considered evaluation datasets, we first independently sample 50 source sentences and 50 random *unaligned* simple sentences from each validation set. These two groups of non-parallel sentences are used to approximate the character-level compression ratio between complex and simplified sentences. We do so by dividing the average length of the simplified sentences by the average length of the 50 source sentences. We finally use this approximated compression ratio as the value of all 4 ACCESS hyper-parameters. In practice, we obtain the following approximations: ASSET $= 0.8$, TURKCORPUS $= 0.95$, and NEWSELA $= 0.4$ (rounded to 0.05). Results in Table 10.8 show that the resulting model performs very close to when we adjust the ACCESS hyper-parameters using SARI on the complete validation set.

Comparing to Existing Paraphrase Datasets We compare using our mined paraphrase data with existing large-scale paraphrase datasets in Table 10.9. We use PARANMT [Wieting and Gimpel, 2018], a large paraphrase dataset created using back-translation on an existing labeled parallel machine translation dataset. We use the same 5 million top-scoring sentences that the authors used to train their sentence embeddings. Training MUSS on the mined data or on PARANMT obtains similar results for text simplification, confirming that mining paraphrase data is a viable alternative to using existing paraphrase datasets relying on labeled parallel machine translation corpora.

Data	**ASSET** SARI ↑	**TURKCORPUS** SARI ↑	**NEWSELA** SARI ↑
MINED	$42.65_{\pm 0.23}$	$40.85_{\pm 0.15}$	$38.09_{\pm 0.59}$
PARANMT	$42.50_{\pm 0.33}$	$40.50_{\pm 0.16}$	$39.11_{\pm 0.88}$

Table 10.9: **Mined Data vs. ParaNMT**
We compare SARI scores of MUSS trained either on our mined data or on PARANMT [Wieting and Gimpel, 2018] on the test sets of ASSET, TURKCORPUS and NEWSELA.

Original	They are culturally akin to the coastal peoples of Papua New Guinea.
ACCESS	**They're** culturally **much like** the Papua New **Guinea coastal peoples.**
BART+ACCESS	They are **closely related** to coastal **people** of Papua New Guinea
Original	Orton and his wife welcomed Alanna Marie Orton on July 12, 2008.
ACCESS	Orton and his wife **had been called** Alanna Marie Orton on July **12**.
BART+ACCESS	Orton and his wife **gave birth to** Alanna Marie Orton on July 12, 2008.
Original	He settled in London, devoting himself chiefly to practical teaching.
ACCESS	He **set up** in **London and made** himself **mainly for** teaching.
BART+ACCESS	He settled in **London and devoted** himself to teaching.

Table 10.10: **Influence of BART on Simplifications.** We display some examples of generations that illustrate how BART improves the fluency and meaning preservation of generated simplifications.

Influence of BART on Fluency In Table 10.10, we present some selected samples that highlight the improved fluency of simplifications when using BART.

Seq2Seq Models on Mined Data When training a Transformer sequence-to-sequence model (Seq2Seq) on WIKILARGE compared to the mined corpus, models trained on the mined data perform better. It is surprising that a model trained solely on paraphrases achieves such good results on simplification benchmarks. Previous works have shown that simplification models suffer from not making enough modifications to the source sentence and found that forcing models to rewrite the input was beneficial [Wubben et al., 2012]. This is confirmed when investigating the F1 deletion component of SARI which is 20 points higher for the model trained on paraphrases.

10.5 Summary and Final Remarks

We propose a sentence simplification approach that does not rely on labeled parallel simplification data thanks to controllable generation, pretraining and large-scale mining of paraphrases from the web. This approach is language-agnostic and matches or outperforms previous state-of-the-art results, even from supervised systems that use labeled simplification data, on three languages: English, French, and Spanish. In future work, we plan to investigate how to scale this approach to more languages and types of simplification, and to apply this method to paraphrase generation. Another interesting direction for future work would to examine and improve factual consistency, especially related to named entity hallucination or disappearance.

Chapter 11

CamemBERT: Using Pretrained Monolingual Models for French Simplification

In the previous Chapter, we explored how to build Sentence Simplification models for languages other than English without using labelled simplification data. One key component was the use of the multilingual pretrained model MBART. In this thesis, however we have a specific focus on the french language. As a result, in this section, we explore in more detail how to pretrain models specifically for French and show that monolingual models can outperform multilingual models in some tasks, including Sentence Simplification.[1]

Pretrained word representations have a long history in Natural Language Processing (NLP), from non-contextual [Brown et al., 1992, Ando and Zhang, 2005, Mikolov et al., 2013, Pennington et al., 2014] to contextual word embeddings [Peters et al., 2018, Akbik et al., 2018]. Word representations are usually obtained by training language model architectures on large amounts of textual data and then fed as an input to more complex task-specific architectures. More recently, these specialized architectures have been replaced altogether by large-scale pretrained language models which are *fine-tuned* for each application considered. This shift has resulted in large improvements in performance over a wide range of tasks [Devlin et al., 2019, Radford et al., 2019, Liu et al., 2019, Raffel et al., 2020].

[1] This chapter is an extended version of [Martin et al., 2020c].

These transfer learning methods exhibit clear advantages over more traditional task-specific approaches. In particular, they can be trained in an *unsupervised* manner, thereby taking advantage of the information contained in large amounts of raw text. Yet they come with implementation challenges, namely the amount of data and computational resources needed for pretraining, which can reach hundreds of gigabytes of text and require hundreds of GPUs [Yang et al., 2019, Liu et al., 2019]. This has limited the availability of these state-of-the-art models to the English language, at least in the monolingual setting. This is particularly inconvenient as it hinders their practical use in NLP systems. It also prevents us from investigating their language modeling capacity, for instance in the case of morphologically rich languages.

Although multilingual models give remarkable results, they are often larger, and their results, as we will observe for French, can lag behind their monolingual counterparts for high-resource languages.

In order to reproduce and validate results that have so far only been obtained for English, we take advantage of the then newly available multilingual corpora OSCAR [Ortiz Suárez et al., 2019] to train a monolingual language model for French, dubbed CamemBERT. We also train alternative versions of CamemBERT on different smaller corpora with different levels of homogeneity in genre and style in order to assess the impact of these parameters on downstream task performance. CamemBERT uses the RoBERTa architecture [Liu et al., 2019], an improved variant of the high-performing and widely used BERT architecture [Devlin et al., 2019].

We evaluate our model on four different downstream tasks for French: part-of-speech (POS) tagging, dependency parsing, named entity recognition (NER) and natural language inference (NLI). CamemBERT improves on the state of the art in all four tasks compared to previous monolingual and multilingual approaches including mBERT, XLM and XLM-R, which confirms the effectiveness of large pretrained language models for French.

Finally we adapt CamemBERT to the task of Sentence Simplification using methods from the Chapter 10 and show that it can obtain new state-of-the-art performance for Sentence Simplification in French.

We make the following contributions:

- First release of a monolingual RoBERTa model for the French language using recently introduced large-scale open source corpora from the Oscar collection and first outside the original BERT authors to release such a large model for an other language than English.[2]

- We achieve state-of-the-art results on four downstream tasks: POS tagging, dependency parsing, NER and NLI, confirming the effectiveness of BERT-based language models for French.

- We demonstrate that small and diverse training sets can achieve similar performance to large-scale corpora, by analyzing the importance of the pretraining corpus in terms of size and domain.

- We finally adapt CamemBERT to the task of Sentence Simplification and obtain state-of-the-art performance, even outperforming the French MUSS model from Chapter 10.

11.1 Previous work

11.1.1 Contextual Language Models

From non-contextual to contextual word embeddings The first neural word vector representations were non-contextualized word embeddings, most notably word2vec [Mikolov et al., 2013], GloVe [Pennington et al., 2014] and fastText [Mikolov et al., 2018], which were designed to be used as input to task-specific neural architectures. Contextualized word representations such as ELMo [Peters et al., 2018] and flair [Akbik et al., 2018], improved the representational power of word embeddings by taking context into account. Among other reasons, they improved the performance of models on many tasks by handling words polysemy. This paved the way for larger contextualized models that replaced downstream architectures altogether in most tasks. Trained with language modeling objectives, these

[2]Released at: `https://camembert-model.fr` under the MIT open-source license.

approaches range from LSTM-based architectures such as [Dai and Le, 2015], to the successful transformer-based architectures such as GPT2 [Radford et al., 2019], BERT [Devlin et al., 2019], RoBERTa [Liu et al., 2019] and more recently ALBERT [Lan et al., 2019] and T5 [Raffel et al., 2020].

Non-English contextualized models Following the success of large pretrained language models, they were extended to the multilingual setting with multilingual BERT (hereafter mBERT) [Devlin et al., 2018], a single multilingual model for 104 different languages trained on Wikipedia data, and later XLM [Lample and Conneau, 2019], which significantly improved unsupervised machine translation. More recently XLM-R [Conneau et al., 2020], extended XLM by training on 2.5TB of data and outperformed previous scores on multilingual benchmarks. They show that multilingual models can obtain results competitive with monolingual models by leveraging higher quality data from other languages on specific downstream tasks.

A few non-English monolingual models have been released: ELMo models for Japanese, Portuguese, German and Basque[3] and BERT for Simplified and Traditional Chinese [Devlin et al., 2018] and German [Chan et al., 2019].

However, to the best of our knowledge, no particular effort has been made toward training models for languages other than English at a scale similar to the latest English models (e.g. RoBERTa trained on more than 100GB of data).

BERT and RoBERTa Our approach is based on RoBERTa [Liu et al., 2019] which itself is based on BERT [Devlin et al., 2019]. BERT is a multi-layer bidirectional Transformer encoder trained with a masked language modeling (MLM) objective, inspired by the Cloze task [Taylor, 1953]. It comes in two sizes: the $\text{BERT}_{\text{BASE}}$ architecture and the $\text{BERT}_{\text{LARGE}}$ architecture. The $\text{BERT}_{\text{BASE}}$ architecture is 3 times smaller and therefore faster and easier to use while $\text{BERT}_{\text{LARGE}}$ achieves increased performance on downstream tasks. RoBERTa improves the original implementation of BERT by identifying key design choices for better performance, using dynamic masking, removing the next sentence prediction task, training

[3] https://allennlp.org/elmo

with larger batches, on more data, and for longer.

11.2 Downstream evaluation tasks

In this section, we present the four downstream tasks that we use to evaluate CamemBERT, namely: Part-Of-Speech (POS) tagging, dependency parsing, Named Entity Recognition (NER) and Natural Language Inference (NLI). We also present the baselines that we will use for comparison.

Tasks POS tagging is a low-level syntactic task, which consists in assigning to each word its corresponding grammatical category. Dependency parsing consists in predicting the labeled syntactic tree in order to capture the syntactic relations between words.

For both of these tasks we run our experiments using the Universal Dependencies (UD)[4] framework and its corresponding UD POS tag set [Petrov et al., 2012] and UD treebank collection which was used for the CoNLL 2018 shared task [Seker et al., 2018]. We perform our evaluations on the four freely available French UD treebanks in UD v2.2: GSD [McDonald et al., 2013], Sequoia[5] [Candito and Seddah, 2012, Candito et al., 2014], Spoken [Lacheret et al., 2014, Bawden et al., 2014][6], and ParTUT [Sanguinetti and Bosco, 2015].

Treebank	#Tokens	#Sentences	Genres
GSD	389,363	16,342	Blogs, News Reviews, Wiki
Sequoia	68,615	3,099	Medical, News Non-fiction, Wiki
Spoken	34,972	2,786	Spoken
ParTUT	27,658	1,020	Legal, News, Wikis
FTB	350,930	27,658	News

Table 11.1: Statistics on the treebanks used in POS tagging, dependency parsing, and NER (FTB).

We also evaluate our model in NER, which is a sequence labeling task predicting which

[4] https://universaldependencies.org
[5] https://deep-sequoia.inria.fr
[6] Speech transcript uncased that includes annotated disfluencies without punctuation.

words refer to real-world objects, such as people, locations, artifacts and organisations. We use the French Treebank[7] (FTB) [Abeillé et al., 2003] in its 2008 version introduced by Candito and Crabbé [2009] and with NER annotations by Sagot et al. [2012]. The FTB contains more than 11 thousand entity mentions distributed among 7 different entity types. A brief overview of the FTB can also be found in Table 11.1.

Finally, we evaluate our model on NLI, using the French part of the XNLI dataset [Conneau et al., 2018]. NLI consists in predicting whether a hypothesis sentence is entailed, neutral or contradicts a premise sentence. The XNLI dataset is the extension of the Multi-Genre NLI (MultiNLI) corpus [Williams et al., 2018] to 15 languages by translating the validation and test sets manually into each of those languages. The English training set is machine translated for all languages other than English. The dataset is composed of 122k train, 2490 development and 5010 test examples for each language. As usual, NLI performance is evaluated using accuracy.

Baselines In dependency parsing and POS-tagging we compare our model with:

- *mBERT*: The multilingual cased version of BERT (see Section 11.1.1). We fine-tune mBERT on each of the treebanks with an additional layer for POS-tagging and dependency parsing, in the same conditions as our CamemBERT model.

- $XLM_{MLM\text{-}TLM}$: A multilingual pretrained language model from Lample and Conneau [2019], which showed better performance than mBERT on NLI. We use the version available in the Hugging's Face transformer library [Wolf et al., 2019]; like mBERT, we fine-tune it in the same conditions as our model.

- *UDify* [Kondratyuk, 2019]: A multitask and multilingual model based on mBERT, UDify is trained simultaneously on 124 different UD treebanks, creating a single POS tagging and dependency parsing model that works across 75 different languages. We report the scores from Kondratyuk [2019] paper.

- *UDPipe Future* [Straka, 2018]: An LSTM-based model ranked 3rd in dependency

[7]This dataset has only been stored and used on Inria's servers after signing the research-only agreement.

parsing and 6th in POS tagging at the CoNLL 2018 shared task [Seker et al., 2018]. We report the scores from Kondratyuk [2019] paper.

- *UDPipe Future + mBERT + Flair* [Straka et al., 2019]: The original UDPipe Future implementation using mBERT and Flair as feature-based contextualized word embeddings. We report the scores from Straka et al. [2019] paper.

In French, no extensive work has been done on NER due to the limited availability of annotated corpora. Thus we compare our model with the only recent available baselines set by Dupont [2017], who trained both CRF [Lafferty et al., 2001] and BiLSTM-CRF [Lample et al., 2016] architectures on the FTB and enhanced them using heuristics and pretrained word embeddings. Additionally, as for POS and dependency parsing, we compare our model to a fine-tuned version of mBERT for the NER task.

For XNLI, we provide the scores of mBERT which has been reported for French by Wu and Dredze [2019]. We report scores from $\text{XLM}_{\text{MLM-TLM}}$ (described above), the best model from Lample and Conneau [2019]. We also report the results of XLM-R [Conneau et al., 2020].

11.3 CamemBERT: a French Language Model

In this section, we describe the pretraining data, architecture, training objective and optimisation setup we use for CamemBERT.

11.3.1 Training data

Pretrained language models benefits from being trained on large datasets [Devlin et al., 2018, Liu et al., 2019, Raffel et al., 2020]. We therefore use the French part of the OSCAR corpus [Ortiz Suárez et al., 2019], a pre-filtered and pre-classified version of Common Crawl.[8]

OSCAR is a set of monolingual corpora extracted from Common Crawl snapshots. It follows the same approach as [Grave et al., 2018] by using a language classification model

[8] `https://commoncrawl.org/about/`

based on the fastText linear classifier [Joulin et al., 2017, 2016] pretrained on Wikipedia, Tatoeba and SETimes, which supports 176 languages. No other filtering is done. We use a non-shuffled version of the French data, which amounts to 138GB of raw text and 32.7B tokens after subword tokenization.

11.3.2 Pre-processing

We segment the input text data into subword units using SentencePiece [Kudo and Richardson, 2018]. SentencePiece is an extension of Byte-Pair encoding (BPE) [Sennrich et al., 2016b] and WordPiece [Kudo, 2018] that does not require pre-tokenization (at the word or token level), thus removing the need for language-specific tokenizers. We use a vocabulary size of 32k subword tokens. These subwords are learned on 10^7 sentences sampled randomly from the pretraining dataset. We do not use subword regularization (i.e. sampling from multiple possible segmentations) for the sake of simplicity.

11.3.3 Language Modeling

Transformer Similar to RoBERTa and BERT, CamemBERT is a multi-layer bidirectional Transformer [Vaswani et al., 2017]. Given the widespread usage of Transformers, we do not describe them here and refer the reader to [Vaswani et al., 2017]. CamemBERT uses the original architectures of $\text{BERT}_{\text{BASE}}$ (12 layers, 768 hidden dimensions, 12 attention heads, 110M parameters) and $\text{BERT}_{\text{LARGE}}$ (24 layers, 1024 hidden dimensions, 16 attention heads, 335M parameters). CamemBERT is very similar to RoBERTa, the main difference being the use of whole-word masking and the usage of SentencePiece tokenization [Kudo and Richardson, 2018] instead of WordPiece [Schuster and Nakajima, 2012].

Pretraining Objective We train our model on the Masked Language Modeling (MLM) task. Given an input text sequence composed of N tokens $x_1, ..., x_N$, we select 15% of tokens for possible replacement. Among those selected tokens, 80% are replaced with the special `<MASK>` token, 10% are left unchanged and 10% are replaced by a random token. The model is then trained to predict the initial masked tokens using cross-entropy loss.

Following the RoBERTa approach, we dynamically mask tokens instead of fixing them statically for the whole dataset during preprocessing. This improves variability and makes the model more robust when training for multiple epochs.

Since we use SentencePiece to tokenize our corpus, the input tokens to the model are a mix of whole words and subwords. An upgraded version of BERT[9] and Joshi et al. [2020] have shown that masking whole words instead of individual subwords leads to improved performance. Whole-word Masking (WWM) makes the training task more difficult because the model has to predict a whole word rather than predicting only part of the word given the rest. We train our models using WWM by using whitespaces in the initial untokenized text as word delimiters.

WWM is implemented by first randomly sampling 15% of the words in the sequence and then considering all subword tokens in each of this 15% for candidate replacement. This amounts to a proportion of selected tokens that is close to the original 15%. These tokens are then either replaced by <MASK> tokens (80%), left unchanged (10%) or replaced by a random token.

Subsequent work has shown that the next sentence prediction (NSP) task originally used in BERT does not improve downstream task performance [Lample and Conneau, 2019, Liu et al., 2019], thus we also remove it.

Optimisation Following [Liu et al., 2019], we optimize the model using Adam [Kingma and Ba, 2014] ($\beta_1 = 0.9$, $\beta_2 = 0.98$) for 100k steps with large batch sizes of 8192 sequences, each sequence containing at most 512 tokens. We enforce each sequence to only contain complete paragraphs (which correspond to lines in the our pretraining dataset).

Pretraining We use the RoBERTa implementation in the fairseq library [Ott et al., 2019]. Our learning rate is warmed up for 10k steps up to a peak value of 0.0007 instead of the original 0.0001 given our large batch size, and then fades to zero with polynomial decay. Unless otherwise specified, our models use the BASE architecture, and are pretrained for 100k backpropagation steps on 256 Nvidia V100 GPUs (32GB each) for a day. We do not

[9] https://github.com/google-research/bert/blob/master/README.md

train our models for longer due to practical considerations, even though the performance still seemed to be increasing.

11.3.4 Using CamemBERT for downstream tasks

We use the pretrained CamemBERT in two ways. In the first one, which we refer to as *fine-tuning*, we fine-tune the model on a specific task in an end-to-end manner. In the second one, referred to as *feature-based embeddings* or simply *embeddings*, we extract frozen contextual embedding vectors from CamemBERT. These two complementary approaches shed light on the quality of the pretrained hidden representations captured by CamemBERT.

Fine-tuning For each task, we append the relevant predictive layer on top of CamemBERT's architecture. Following the work done on BERT [Devlin et al., 2019], for sequence tagging and sequence labeling we append a linear layer that respectively takes as input the last hidden representation of the `<s>` special token and the last hidden representation of the first subword token of each word. For dependency parsing, we plug a bi-affine graph predictor head as inspired by Dozat and Manning [2017]. We refer the reader to this article for more details on this module. We fine-tune on XNLI by adding a classification head composed of one hidden layer with a non-linearity and one linear projection layer, with input dropout for both.

We fine-tune CamemBERT independently for each task and each dataset. We optimize the model using the Adam optimiser [Kingma and Ba, 2014] with a fixed learning rate. We run a grid search on a combination of learning rates and batch sizes. We select the best model on the validation set out of the 30 first epochs. For NLI we use the default hyper-parameters provided by the authors of RoBERTa on the MNLI task.[10] Although this might have pushed the performances even further, we do not apply any regularization techniques such as weight decay, learning rate warm-up or discriminative fine-tuning, except for NLI. We show that fine-tuning CamemBERT in a straightforward manner leads to state-of-the-art results on all tasks and outperforms the existing BERT-based models in all cases. The POS tagging,

[10]More details at `https://github.com/pytorch/fairseq/blob/master/examples/roberta/README.glue.md`.

dependency parsing, and NER experiments are run using Hugging Face's Transformer library extended to support CamemBERT and dependency parsing [Wolf et al., 2019]. The NLI experiments use the fairseq library following the RoBERTa implementation.

Embeddings Following Straková et al. [2019] and Straka et al. [2019] for mBERT and the English BERT, we make use of CamemBERT in a feature-based embeddings setting. In order to obtain a representation for a given token, we first compute the average of each sub-word's representations in the last four layers of the Transformer, and then average the resulting sub-word vectors.

We evaluate CamemBERT in the embeddings setting for POS tagging, dependency parsing and NER; using the open-source implementations of [Straka et al., 2019] and [Straková et al., 2019].[11]

11.4 Evaluation of CamemBERT

In this section, we measure the performance of our models by evaluating them on the four aforementioned tasks: POS tagging, dependency parsing, NER and NLI.

Model	GSD		Sequoia		Spoken		ParTUT	
	UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS
mBERT (fine-tuned)	97.48	89.73	98.41	91.24	96.02	78.63	97.35	91.37
$XLM_{MLM-TLM}$ (fine-tuned)	98.13	90.03	98.51	91.62	96.18	80.89	97.39	89.43
UDify [Kondratyuk, 2019]	97.83	91.45	97.89	90.05	96.23	80.01	96.12	88.06
UDPipe Future [Straka, 2018]	97.63	88.06	98.79	90.73	95.91	77.53	96.93	89.63
+ mBERT + Flair (emb.) [Straka et al., 2019]	97.98	90.31	**99.32**	93.81	**97.23**	81.40	97.64	92.47
CamemBERT (fine-tuned)	**98.18**	**92.57**	99.29	**94.20**	96.99	81.37	**97.65**	**93.43**
UDPipe Future + CamemBERT (embeddings)	97.96	90.57	99.25	93.89	97.09	**81.81**	97.50	92.32

Table 11.2: **POS** and **dependency parsing** scores on 4 French treebanks, reported on test sets assuming gold tokenization and segmentation (best model selected on validation out of 4). Best scores in bold, second best underlined.

POS tagging and dependency parsing For POS tagging and dependency parsing, we compare CamemBERT with other models in the two settings: *fine-tuning* and as *feature-*

[11] UDPipe Future is available at `https://github.com/CoNLL-UD-2018/UDPipe-Future`, and the code for nested NER is available at `https://github.com/ufal/acl2019_nested_ner`.

Model	F1
SEM (CRF) [Dupont, 2017]	85.02
LSTM-CRF [Dupont, 2017]	85.57
mBERT (fine-tuned)	87.35
CamemBERT (fine-tuned)	<u>89.08</u>
LSTM+CRF+CamemBERT (embeddings)	**89.55**

Table 11.3: **NER** scores on the FTB (best model selected on validation out of 4). Best scores in bold, second best underlined.

Model	Acc.	#Params
mBERT [Devlin et al., 2019]	76.9	175M
$\text{XLM}_{\text{MLM-TLM}}$ [Lample and Conneau, 2019]	<u>80.2</u>	250M
$\text{XLM-R}_{\text{BASE}}$ [Conneau et al., 2020]	80.1	270M
CamemBERT (fine-tuned)	**82.5**	110M
Supplement: LARGE models		
$\text{XLM-R}_{\text{LARGE}}$ [Conneau et al., 2020]	<u>85.2</u>	550M
$\text{CamemBERT}_{\text{LARGE}}$ (fine-tuned)	**85.7**	335M

Table 11.4: **NLI** accuracy on the French XNLI test set (best model selected on validation out of 10). Best scores in bold, second best underlined.

based embeddings. We report the results in Table 11.2.

CamemBERT reaches state-of-the-art scores on all treebanks and metrics in both scenarios. The two approaches achieve similar scores, with a slight advantage for the fine-tuned version of CamemBERT, thus questioning the need for complex task-specific architectures such as UDPipe Future.

Despite a much simpler optimisation process and no task specific architecture, fine-tuning CamemBERT outperforms UDify on all treebanks and sometimes by a large margin (e.g. +4.15% LAS on Sequoia and +5.37 LAS on ParTUT). CamemBERT also reaches better performance than other multilingual pretrained models such as mBERT and $\text{XLM}_{\text{MLM-TLM}}$ on all treebanks.

CamemBERT achieves overall slightly better results than the previous state-of-the-art and task-specific architecture UDPipe Future+mBERT +Flair, except for POS tagging on Sequoia and POS tagging on Spoken, where CamemBERT lags by 0.03% and 0.14% UPOS respectively. UDPipe Future+mBERT +Flair uses the contextualized string embeddings Flair [Akbik et al., 2018], which are in fact pretrained contextualized character-level word

embeddings specifically designed to handle misspelled words as well as subword structures such as prefixes and suffixes. This design choice might explain the difference in score for POS tagging with CamemBERT, especially for the Spoken treebank where words are not capitalized, a factor that might pose a problem for CamemBERT which was trained on capitalized data, but that might be properly handle by Flair on the UDPipe Future+mBERT +Flair model.

Named-Entity Recognition For NER, we similarly evaluate CamemBERT in the fine-tuning setting and as input embeddings to the task specific architecture LSTM+CRF. We report these scores in Table 11.3.

In both scenarios, CamemBERT achieves higher F1 scores than the traditional CRF-based architectures, both non-neural and neural, and than fine-tuned multilingual BERT models.[12]

Using CamemBERT as embeddings to the traditional LSTM+CRF architecture gives slightly higher scores than by fine-tuning the model (89.08 vs. 89.55). This demonstrates that although CamemBERT can be used successfully without any task-specific architecture, it can still produce high quality contextualized embeddings that might be useful in scenarios where powerful downstream architectures exist.

Natural Language Inference On the XNLI benchmark, we compare CamemBERT to previous state-of-the-art multilingual models in the fine-tuning setting. In addition to the standard CamemBERT model with a BASE architecture, we train another model with the LARGE architecture, referred to as $\text{CamemBERT}_{\text{LARGE}}$, for a fair comparison with $\text{XLM-R}_{\text{LARGE}}$. This model is trained with the CCNET corpus, described in Sec. 11.5, for 100k steps.[13] We expect that training the model for longer would yield even better performance.

CamemBERT reaches higher accuracy than its BASE counterparts reaching +5.6% over mBERT, +2.3 over $\text{XLM}_{\text{MLM-TLM}}$, and +2.4 over $\text{XLM-R}_{\text{BASE}}$. CamemBERT also uses as

[12] $\text{XLM}_{\text{MLM-TLM}}$ is a lower-case model. Case is crucial for NER, therefore we do not report its low performance (84.37%)

[13] We train our LARGE model with the CCNET corpus for practical reasons. Given that BASE models reach similar performance when using OSCAR or CCNET as pretraining corpus (Table 11.7), we expect an OSCAR LARGE model to reach comparable scores.

few as half as many parameters (110M vs. 270M for XLM-R_{BASE}).

CamemBERT$_{LARGE}$ achieves a state-of-the-art accuracy of 85.7% on the XNLI benchmark, as opposed to 85.2, for the recent XLM-R_{LARGE}.

CamemBERT uses fewer parameters than multilingual models, mostly because of its smaller vocabulary size (e.g. 32k vs. 250k for XLM-R). Two elements might explain the better performance of CamemBERT over XLM-R. Even though XLM-R was trained on an impressive amount of data (2.5TB), only 57GB of this data is in French, whereas we used 138GB of French data. Additionally XLM-R also handles 100 languages, and the authors show that when reducing the number of languages to 7, they can reach 82.5% accuracy for French XNLI with their BASE architecture.

Summary of CamemBERT's results CamemBERT improves the state of the art for the 4 downstream tasks considered, thereby confirming on French the usefulness of Transformer-based models. We obtain these results when using CamemBERT as a fine-tuned model or when used as contextual embeddings with task-specific architectures. This questions the need for more complex downstream architectures, similar to what was shown for English [Devlin et al., 2019]. Additionally, this suggests that CamemBERT is also able to produce high-quality representations out-of-the-box without further tuning.

Dataset	Size	GSD		Sequoia		Spoken		ParTUT		**Average**		NER	NLI
		UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	UPOS	LAS	F1	Acc.
Fine-tuning													
Wiki	4GB	98.28	93.04	98.74	92.71	96.61	79.61	96.20	89.67	97.45	88.75	89.86	78.32
CCNet	4GB	98.34	93.43	98.95	93.67	96.92	**82.09**	96.50	**90.98**	97.67	**90.04**	90.46	**82.06**
OSCAR	4GB	98.35	93.55	98.97	93.70	96.94	81.97	96.58	90.28	97.71	89.87	90.65	81.88
OSCAR	138GB	**98.39**	**93.80**	**98.99**	**94.00**	**97.17**	81.18	**96.63**	90.56	**97.79**	89.88	**91.55**	81.55
Embeddings (with UDPipe Future (tagging, parsing) or LSTM+CRF (NER))													
Wiki	4GB	98.09	92.31	98.74	93.55	96.24	78.91	95.78	89.79	97.21	88.64	91.23	-
CCNet	4GB	**98.22**	**92.93**	99.12	94.65	97.17	**82.61**	**96.74**	89.95	97.81	90.04	**92.30**	-
OSCAR	4GB	98.21	92.77	99.12	**94.92**	97.20	82.47	**96.74**	**90.05**	**97.82**	**90.05**	91.90	-
OSCAR	138GB	98.18	92.77	**99.14**	94.24	**97.26**	82.44	96.52	89.89	97.77	89.84	91.83	-

Table 11.5: Results on the four tasks using language models pre-trained on data sets of varying homogeneity and size, reported on validation sets (average of 4 runs for POS tagging, parsing and NER, average of 10 runs for NLI).

11.5 Impact of corpus origin and size

In this section we investigate the influence of the homogeneity and size of the pretraining corpus on downstream task performance. With this aim, we train alternative version of CamemBERT by varying the pretraining datasets. For this experiment, we fix the number of pretraining steps to 100k, and allow the number of epochs to vary accordingly (more epochs for smaller dataset sizes). All models use the BASE architecture.

In order to investigate the need for homogeneous clean data versus more diverse and possibly noisier data, we use alternative sources of pretraining data in addition to OSCAR:

- **Wikipedia**, which is homogeneous in terms of genre and style. We use the official 2019 French Wikipedia dumps[14]. We remove HTML tags and tables using Giuseppe Attardi's *WikiExtractor*.[15]

- **CCNET** [Wenzek et al., 2020], a dataset extracted from Common Crawl with a different filtering process than for OSCAR. It was built using a language model trained on Wikipedia, in order to filter out bad quality texts such as code or tables.[16] As this filtering step biases the noisy data from Common Crawl to more Wikipedia-like text, we expect CCNET to act as a middle ground between the unfiltered "noisy" OSCAR dataset, and the "clean" Wikipedia dataset. As a result of the different filtering processes, CCNET contains longer documents on average compared to OSCAR with smaller—and often noisier—documents weeded out.

Table 11.6 summarizes statistics of these different corpora.

In order to make the comparison between these three sources of pretraining data, we randomly sample 4GB of text (at the document level) from OSCAR and CCNET, thereby creating samples of both Common-Crawl-based corpora of the same size as the French Wikipedia. These smaller 4GB samples also provides us a way to investigate the impact of pretraining data size. Downstream task performance for our alternative versions of CamemBERT are provided in Table 11.5. The upper section reports scores in the fine-tuning

[14] `https://dumps.wikimedia.org/backup-index.html`.
[15] `https://github.com/attardi/wikiextractor`.
[16] We use the HEAD split, which corresponds to the top 33% of documents in terms of filtering perplexity.

Corpus	Size	#tokens	#docs	Tokens/doc Percentiles:		
				5%	50%	95%
Wikipedia	4GB	990M	1.4M	102	363	2530
CCNet	135GB	31.9B	33.1M	128	414	2869
OSCAR	138GB	32.7B	59.4M	28	201	1946

Table 11.6: Statistics on the pretraining datasets used.

setting while the lower section reports scores for the embeddings.

11.5.1 Common Crawl vs. Wikipedia?

Table 11.5 clearly shows that models trained on the 4GB versions of OSCAR and CCNET (Common Crawl) perform consistently better than the the one trained on the French Wikipedia. This is true both in the fine-tuning and embeddings setting. Unsurprisingly, the gap is larger on tasks involving texts whose genre and style are more divergent from those of Wikipedia, such as tagging and parsing on the Spoken treebank. The performance gap is also very large on the XNLI task, probably as a consequence of the larger diversity of Common-Crawl-based corpora in terms of genres and topics. XNLI is indeed based on multiNLI which covers a range of genres of spoken and written text.

The downstream task performances of the models trained on the 4GB version of CCNET and OSCAR are much more similar.[17]

11.5.2 How much data do you need?

An unexpected outcome of our experiments is that the model trained "only" on the 4GB sample of OSCAR performs similarly to the standard CamemBERT trained on the whole 138GB OSCAR. The only task with a large performance gap is NER, where "138GB" models are better by 0.9 F1 points. This could be due to the higher number of named entities present in the larger corpora, which is beneficial for this task. On the contrary, other tasks do not seem to gain from the additional data.

[17]We provide the results of a model trained on the whole CCNET corpus in the Table 11.7. The conclusions are similar when comparing models trained on the full corpora: downstream results are similar when using OSCAR or CCNET.

In other words, when trained on corpora such as OSCAR and CCNET, which are heterogeneous in terms of genre and style, 4GB of uncompressed text is large enough as pretraining corpus to reach state-of-the-art results with the BASE architecure, better than those obtained with mBERT (pretrained on 60GB of text).[18] This calls into question the need to use a very large corpus such as OSCAR or CCNET when training a monolingual Transformer-based language model such as BERT or RoBERTa. Not only does this mean that the computational (and therefore environmental) cost of training a state-of-the-art language model can be reduced, but it also means that CamemBERT-like models can be trained for all languages for which a Common-Crawl-based corpus of 4GB or more can be created. OSCAR is available in 166 languages, and provides such a corpus for 38 languages. Moreover, it is possible that slightly smaller corpora (e.g. down to 1GB) could also prove sufficient to train high-performing language models. We obtained our results with BASE architectures. Further research is needed to confirm the validity of our findings on larger architectures and other more complex natural language understanding tasks. However, even with a BASE architecture and 4GB of training data, the validation loss is still decreasing beyond 100k steps (and 400 epochs). This suggests that we are still under-fitting the 4GB pretraining dataset, training longer might increase downstream performance.

11.6 Design Choices

11.6.1 Impact of Whole-Word Masking

In Table 11.7, we compare models trained using the traditional subword masking with whole-word masking. Whole-Word Masking positively impacts downstream performances for NLI (although only by 0.5 points of accuracy). To our surprise, this Whole-Word Masking scheme does not benefit much lower level task such as Name Entity Recognition, POS tagging and Dependency Parsing.

[18]The OSCAR-4GB model gets slightly better XNLI accuracy than the full OSCAR-138GB model (81.88 vs. 81.55). This might be due to the random seed used for pretraining, as each model is pretrained only once.

DATASET	MASKING	ARCH.	#PARAM.	#STEPS	UPOS	LAS	NER	XNLI
Masking Strategy								
OSCAR	Subword	BASE	110M	100k	97.78	89.80	**91.55**	81.04
OSCAR	Whole-word	BASE	110M	100k	**97.79**	**89.88**	91.44	**81.55**
Model Size								
CCNET	Whole-word	BASE	110M	100k	97.67	89.46	90.13	82.22
CCNET	Whole-word	LARGE	335M	100k	**97.74**	**89.82**	**92.47**	**85.73**
Dataset								
CCNET	Whole-word	BASE	110M	100k	97.67	89.46	90.13	**82.22**
OSCAR	Whole-word	BASE	110M	100k	**97.79**	**89.88**	**91.44**	81.55
Number of Steps								
CCNET	Whole-word	BASE	110M	100k	**98.04**	89.85	90.13	82.20
CCNET	Whole-word	BASE	110M	500k	97.95	**90.12**	91.30	**83.04**

Table 11.7: Comparing scores on the **Validation sets** of different design choices. POS tagging and parsing datasets are averaged. (average over multiple fine-tuning seeds).

11.6.2 Impact of model size

Table 11.7 compares models trained with the BASE and LARGE architectures. These models were trained with the CCNET corpus (135GB) for practical reasons. We confirm the positive influence of larger models on the NLI and NER tasks. The LARGE architecture leads to respectively 19.7% error reduction and 23.7%. To our surprise, on POS tagging and dependency parsing, having three time more parameters does not lead to a significant difference compared to the BASE model. [Tenney et al., 2019] and [Jawahar et al., 2019] have shown that low-level syntactic capabilities are learnt in lower layers of BERT while higher level semantic representations are found in upper layers of BERT. POS tagging and dependency parsing probably do not benefit from adding more layers as the lower layers of the BASE architecture already capture what is necessary to complete these tasks.

11.6.3 Impact of training dataset

Table 11.7 compares models trained on CCNET and on OSCAR. The major difference between the two datasets is the additional filtering step of CCNET that favors Wikipedia-Like texts. The model pretrained on OSCAR gets slightly better results on POS tagging and dependency parsing, but gets a larger +1.31 improvement on NER. The CCNET model gets

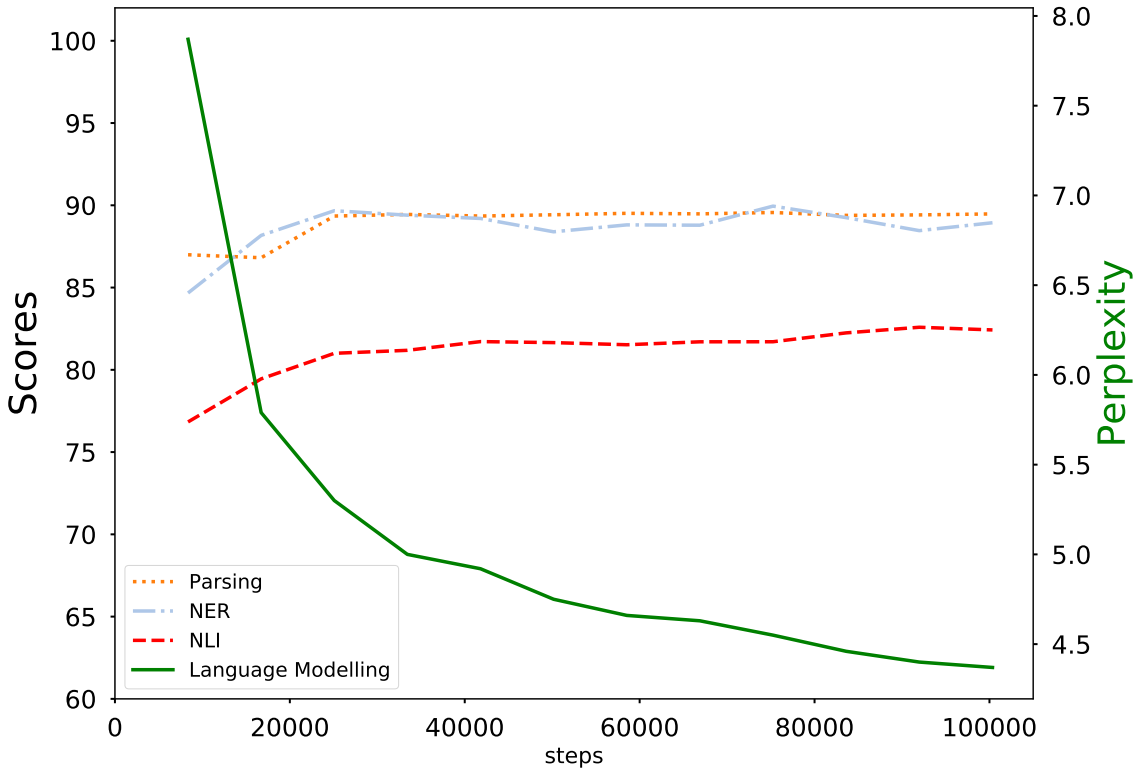


Figure 11.1: Impact of number of pretraining steps on downstream performance for CamemBERT.

.

better performance on NLI (+0.67).

11.6.4 Impact of number of steps

Figure 11.1 displays the evolution of downstream task performance with respect to the number of steps. All scores in this section are averages from at least 4 runs with different random seeds. For POS tagging and dependency parsing, we also average the scores on the 4 treebanks.

We evaluate our model at every epoch (1 epoch equals 8360 steps). We report the masked language modeling perplexity along with downstream performances. Figure 11.1, suggests that the more complex the task the more impactful the number of steps is. We observe an early plateau for dependency parsing and NER at around 22k steps, while for NLI, even if the marginal improvement with regard to pretraining steps becomes smaller, the performance is still slowly increasing at 100k steps.

In Table 11.7, we compare two models trained on CCNET, one for 100k steps and the other for 500k steps to evaluate the influence of the total number of steps. The model trained for 500k steps does not increase the scores much from just training for 100k steps in POS tagging and parsing. The increase is slightly higher for XNLI (+0.84).

Those results suggest that low level syntactic representation are captured early in the language model training process while it needs more steps to extract complex semantic information as needed for NLI.

11.7 Discussion

Since the pre-publication of this work, many monolingual language models have appeared, e.g. [Le et al., 2019, Virtanen et al., 2019, Delobelle et al., 2020], for as much as 30 languages [Nozza et al., 2020]. In almost all tested configurations they displayed better results than multilingual language models such as mBERT [Pires et al., 2019]. Interestingly, [Le et al., 2019] showed that using their FlauBert, a RoBERTa-based language model for French, which was trained on less but more edited data, in conjunction to CamemBERT in an ensemble system could improve the performance of a parsing model and establish a new state-of-the-art in constituency parsing of French, highlighting thus the complementarity of both models.[19] As it was the case for English when BERT was first released, the availability of similar scale language models for French enabled interesting applications, such as large scale anonymization of legal texts, where CamemBERT-based models established a new state-of-the-art on this task [Benesty, 2019], or the first large question answering experiments on a French Squad data set that was released very recently [d'Hoffschmidt et al., 2020] where the authors matched human performance using $\text{CamemBERT}_{\text{LARGE}}$. Being the first pre-trained language model that used the open-source Common Crawl Oscar corpus and given its impact on the community, CamemBERT paved the way for many works on monolingual language models that followed. Furthermore, the availability of all its training data favors reproducibility and is a step towards better understanding such models. In that spirit, we make the models used in our experiments available via our website and via the

[19] We refer the reader to [Le et al., 2019] for a comprehensive benchmark and details therein.

Baselines	SARI ↑
Identity	26.16
Truncate	33.44
Pivot	33.48
MUSS	41.73
CAMEMBERT$_{\text{SIMP}}$	**43.39**

(a) Automatic scores on the ALECTOR dataset. We compare CamemBERT$_{\text{Simp}}$ with MUSS and baselines using the SARI automatic metric.

Best System / Aspect	Fluency	Meaning	Simplicity
MUSS	15.3%	28.8%	27.1%
CamemBERT$_{\text{Simp}}$	13.6%	30.5%	**45.8%**
Similar	**71.2%**	**40.7%**	27.1%

(b) **Pairwise Human Comparisons**. Human judges compare 60 pairs of simplfications either coming from MUSS or CamemBERT$_{\text{Simp}}$ and choose the best according to each aspect in fluency, meaning preservation and simplicity

Table 11.8: **CamemBERT$_{\text{Simp}}$ Results**. Best scores are in bold.

`huggingface` and `fairseq` APIs, in addition to the base CamemBERT model.

11.8 Leveraging CamemBERT for Sentence Simplification

The Cap'FALC project aims at facilitating Sentence Simplification in French. In this section we show how we can leverage CamemBERT to reach this objective, by creating CamemBERT$_{\text{Simp}}$, a state-of-the-art French Sentence Simplification model. CamemBERT$_{\text{Simp}}$ is an encoder-decoder model initialized with the pretrained CamemBERT model. It shares the same transformer architecture as MUSS from Chapter 10 but is only pretrained in masked language modeling in French, whereas MUSS was pretrained with denoising auto-encoding in 25 languages (MBART).

11.8.1 Method

Our method is composed of three steps:

1. Initialize an Encoder-Decoder with CamemBERT$_{\text{LARGE}}$. BERT models have been successfully used to initialize sequence-to-sequence models for generative tasks [Rothe et al., 2020]. We follow the same method and use the CamemBERT$_{\text{LARGE}}$ checkpoint to initialize both the encoder and decoder of a sequence-to-sequence model.

2. Add the ACCESS Controllable Mechanism. Following findings from Chapter 10 with the MUSS model, we combine our encoder-decoder model initialized with CamemBERT with the ACCESS controllable mechanism. Special tokens are prepended as plain text and split using the CamemBERT SentencePiece model. This way we can then use the Encoder-Decoder model for simplification by reducing the length of the input sentence, its lexical complexity, or syntactic complexity.

3. Finetune on mined French Paraphrases. We finally finetune our controllable model on the 1.36 million French paraphrases that we mined in Chapter 10 (see Table 10.1 for more details). This teaches our model to reformulate a given input sentences in a controlled manner, that we can then adapt to simplification.

11.8.2 Evaluation

Automatic Metrics. We evaluate the performance of CamemBERT$_{\text{Simp}}$ using the SARI automatic metrics on the French ALECTOR simplification dataset in Table 11.8a. We compare CamemBERT$_{\text{Simp}}$ with the identity, truncate, and pivot baselines, and with the French MUSS model, using scores reported in Table 10.4 from Chapter 10. CamemBERT$_{\text{Simp}}$ reaches a 2 points higher SARI than the previous best model.

Human Evaluation. Given the unreliability of automatic metrics, we also complement our evaluation with human pairwise comparison in Table 11.8b. We first sample 60 sentences from various complex documents gathered as part of the Cap'FALC project. These documents cover diverse topics including administrative procedures, health guidelines, and legal notices. These sentences are simplified using CamemBERT$_{\text{Simp}}$ and the French MUSS model using the exact same access parameters (0.8 for each control token). We then recruit a French native speaker volunteer to compare simplifications from the two models. For each source sentence, our human annotator is presented with the two simplifications (ordered randomly) and must then answer three questions:

- **Fluency.** Is one sentence more grammatical or natural?

- **Meaning Preservation.** Which simplification expresses the original meaning the best?

- **Simplicity.** Which simplification is the easiest to read and understand?

Results highlight that CamemBERT$_{\text{Simp}}$ is as fluent and meaning preserving as MUSS but often produces output sentences that are simpler to read and understand.

Differences between MUSS and CamemBERT$_{\text{Simp}}$ Both MUSS and CamemBERT$_{\text{Simp}}$ use the exact same architecture (transformer large for the encoder and decoder), use the same controllable mechanism ACCESS, and are finetuned on the same mined paraphrases.

They however differ in their pretraining phases. MUSS is based on the MBART model [Liu et al., 2020b], which was pretrained on 25 languages on denoising auto-encoding similar to masked language modeling on the CC25 dataset. CamemBERT$_{\text{Simp}}$ on the other end was pretrained with masked language modeling on French only on the CCNET dataset.

The fact that CamemBERT is monolingual is probably the reason why it performs better on Sentence Simplification than MUSS which is based on MBART, thus backing up the hypothesis that monolingual models still outperform multilingual models. Our adaptation of CamemBERT for the task of Sentence Simplification is now the new state-of-the-art in the French Language. In future work, we would like to confirm that hypothesis by comparing CamemBERT$_{\text{Simp}}$ with a MUSS model based on a French monolingual BART.

11.9 Summary and Final Remarks

In this work, we investigated the feasibility of training a Transformer-based language model for languages other than English and how it can be used for state-of-the-art Sentence Simplification.

We trained CamemBERT in French, a language model based on RoBERTa. As a prerequisite to the final objectifve of Sentence Simplification, we evaluated CamemBERT on four downstream tasks (part-of-speech tagging, dependency parsing, named entity recognition and natural language inference) in which our best model reached or improved the state of

the art in all tasks considered, even when compared to strong multilingual models such as mBERT, XLM and XLM-R, while also having fewer parameters.

Our experiments also demonstrate that using web crawled data with high variability is preferable to using Wikipedia-based data. In addition we showed that our models could reach surprisingly high performances with as low as 4GB of pretraining data, questioning thus the need for large scale pretraining corpora. This shows that state-of-the-art Transformer-based language models can be trained on languages with far fewer resources than English, whenever a few gigabytes of data are available. This paves the way for the rise of monolingual contextual pre-trained language-models for under-resourced languages.

Finally we used our pretrained model to train CamemBERT$_{\text{Simp}}$, a state-of-the-art Sentence Simplification model in French, that outperforms our previous best model from Chapter 10. We hope that CamemBERT$_{\text{Simp}}$ can pave the way for future work in Sentence Simplification in under-resources languages.

Pretrained on pure open-source corpora, CamemBERT is freely available and distributed with the MIT license via popular NLP libraries (`fairseq` and `huggingface`) as well as on our website `camembert-model.fr`.

Part V

Conclusion and Perspectives

Chapter 12

Conclusion and Perspectives

12.1 Conclusion

In this thesis, we studied the task of Sentence Simplification. We first explored how Sentence Simplification systems can be evaluated, where evaluation falls short and proposed an evaluation library EASSE, a dataset ASSET and an adaptation of recent evaluation metrics to the task of Sentence Simplification. We then explored how to create Sentence Simplification models that can be controlled and reached state-of-the-art performance in English with ACCESS. Finally we extended our research to other languages with scarce Sentence Simplification resource. We introduced MUSS, a controllable Sentence Simplification method that does not require labelled data but that reaches state-of-the-art scores in multiple languages. Then we studied how pretraining in French can help create even stronger unsupervised Sentence Simplification models with CamemBERT.

Evaluation Sentence Simplification Models In Part II we explored how Sentence Simplification evaluation can improved and where it still falls short.

Chapter 5 examined which linguistic features are best correlated with human judgement of sentence simplification when no reference simplification is available. Results indicate that length based features correlate the most with simplicity, while term-based comparisons between the source and simplification (e.g. BLEU) correlate the most with fluency and meaning preservation. It would still be interesting to confirm these findings on another

larger and more diversified corpus. Further, more elaborate features could also yield better correlation with simplicity judgements such as what was explored in the more recent work [Brunato et al., 2018], or with meaning preservation such as methods based on neural networks [Zhang et al., 2020].

In Chapter 6, we proposed to streamline the evaluation of Sentence Simplification by regrouping traditional metrics in a library called EASSE. Our library fixes bugs and standardizes implementations of metrics, but also include additional quality estimation tools based on what we explored in Chapter 5. This library has played a key role in the evaluation of models proposed in this thesis and has since been used in various papers studying Sentence Simplification.

We then showed in Chapter 7 that current evaluation datasets for Sentence Simplification lack in diversity and in overall simplicity. As a result we built a new dataset, called ASSET that features more diverse simplifications operations, similar to how humans would simplify sentences. This dataset is deemed simpler than previous dataset by human judges, and leads to better correlations with human judgements for automatic metrics. However we raise the concern that correlations with human judgements of traditional metrics are still very low, thus calling for new evaluation metrics to be proposed. Since the publication of this work, ASSET has been integrated in the general purpose GEM benchmark [Gehrmann et al., 2021].

The low correlation of automatic metrics was investigated further in Chapter 8. We showed that existing correlations of traditional metrics with human judgements might be due to spurious correlations when evaluating imperfect system simplifications. Indeed, when evaluating human-written simplifications with automatic metrics, these correlations disappear. We also adapted to recent neural-based evaluation method, namely QUESTEVAL and BERTScore, and showed that they can lead to better results. Evaluation of Sentence Simplification is still an open question and we raise a warning that more research is needed to create accurate evaluation metrics for Sentence Simplification. We therefore think that human evaluation is still necessary when proposing new simplification models, even though human evalution can suffer from low annotator agreement in Sentence Simplification.

Controllable Sentence Simplification in English After discussing evaluation of Sentence Simplification systems, we explored how to create such Sentence Simplification models in English in Part III.

In Chapter 9 we motivate the need for more flexible Sentence Simplification systems, that can be adapted to the needs of different end audiences, and that take into account the wide variety of rewriting operations of Sentence Simplification highlighted in Chapter 7. We proposed ACCESS, a sequence-to-sequence model that is conditionned on mutliple features specific to Sentence Simplification: length, lexical complexity, syntactic complexity, and amount of rewriting. Our model can then be adapted to fit specific types of simplifications and reaches state-of-the-art results in English. Even though our model produces simplifications with better automatic scores, we did not evaluate it using human evaluation, which would allow confirming whether our model is actually performs better. One limit of ACCESS is that it uses fixed control values for a given dataset, sometimes limiting the model too much. For instance it will always try to reduce the length of the input to the exact value provided, sometimes making the generation ungrammatical or removing important information when the source sentence did not need to be shortened.

From English Sentence Simplification to Other Languages In Part IV, we proposed methods to create strong Sentence Simplification models in languages other than English where training data is scarce.

The largest bottleneck for Sentence Simplification in languages other than English is training data. In Chapter 10, we propose a method to mine parallel data from the web in the form of paraphrases, and then use the ACCESS controllable mechanism to condition on simplification specific features and perform Sentence Simplification at test time. Our unsupervised method, MUSS, reaches state-of-the-art results in English, French and Spanish even compared to supervised models, and we further improve results by incorporating labelled simplification data. With this work we discovered that the ACCESS controllable mechanism benefits from more varied data, even if it is not in the form of simplification data. However the paraphrases that we mined still present a low amount of sentence rewriting and similar syntactic structure between the source and target. We identify the mining of even

more varied paraphrase data as the most important possible area of improvement for MUSS. This could be achieved by using other types of sentence representation for instance.

In Chapter 11, we study another approach to Sentence Simplification in French, the language of the Cap'FALC project. This method relies on unsupervised pretraining of a masked language model. We thus proposed CamemBERT, the first pretrained masked language model based on BERT, in a language other than English. CamemBERT reaches state-of-the-art performance on various French downstream tasks. Our monolingual model outperforms strong multilingual models, confirming the need for language-specific models. We also show that pretraining on diverse heterogeneous data from the web is paramount to good performance. Finally we use CamemBERT for the task of Sentence Simplification and show that it can reach even better results than MUSS, most likely due to the fact that it is pretrained in a single language. Further work would be needed to confirm whether the better performance of CamemBERT comes from pretraining on a single language or from other differences such as using masked language modeling.

12.2 Perspectives

In this section we emphasize some perspectives and area of future work following our research.

First Sentence Simplification is hard to define exactly and as such hard to find a good way to evaluate it. Its inherent diversity makes traditional evaluations difficult with low correlation with human judgements even when multiple human references are used.

Evaluating Sentence Simplification models on a Simplicity-Meaning Trade-off Curve We showed in Part II that Sentence Simplification is hard to evaluate. One of the reasons for the difficult evaluation is that for a given source sentence, a wide variety of simplifications are acceptable. In particular, a given sentence can be simplified with a varying degree of simplicity by removing more or less content. Simplicity and meaning preservation are indeed inversely correlated: removing content makes a sentence easier to read but less meaning preserving [Schwarzer and Kauchak, 2018]. The amount of content that should

be removed is application-dependent and cannot be fixed. Therefore, models that have different amounts of content deletion cannot be compared in a fair manner: some will be more meaning preserving but less simple, while others will be less meaning preserving but simpler. Assuming that grammaticality is a prerequisite and can be evaluated independently, future research should investigate a way to compare models that take this simplicity-meaning trade-off into account. Given a meaning preservation score (x-axis) and a simplicity score (y-axis), models could be evaluated in a 2-dimensional trade-off scatter plot. Controllable models could then be evaluated in the form of a pareto curve, by varying the amount of meaning preservation, thus allowing for more pertinent comparisons between approaches.

Finding Better Simplification Controls In Chapter 9 we showed how we could make models controllable by conditioning on predefined simplification-specific features. However those features were chosen using expert knowledge and by trial and error. In future work, it would be interesting to investigate wether the controls could be automatically learned. Instead of computing controls manually (e.g. measuring the length of sentence), an additional model would learn to produce controls in a latent control space. At train time the control model would take the input sentence and target sentence and create a latent vector of very small size (e.g. 4 floats to keep it similar to the number of control tokens that we used in ACCESS). Then those latent controls can be fed as input to the actual Sentence Simplification model along with the source sentence. This differentiable process would teach the control model to integrate useful task-specific information in the latent controls. An additional discriminator could prevent the control model to produce instance-specific semantic information. This method could even be used for other text rewriting tasks and allow out-of-the-box multi-task learning, where the latent would encode task-specific information.

Using MUSS for other tasks Without resorting to learning latent controls, our proposed simplification models ACCESS and MUSS can already be used for other monolingual text rewriting tasks such as paraphrasing or style transfer (e.g. detoxification). They would only need the addition of other task specific controls and adapting the mining process of parallel

sentences. MUSS is actually already capable of performing paraphrasing out-of-the-box given that it was trained on paraphrases.

Document-level Simplification While we focused on sentence-level simplification in this thesis, the field will most likely transition towards document-level simplification. Indeed, most simplification applications are at the document-level (e.g. simplifying news articles, legal documents, or administrative documents). Treating the task at the document-level will involve many new challenges such as anaphora resolution, sentence fusion, summarizing multiple sentences, reordering ideas, generating examples... Given the even larger space of acceptable document simplifications for a given source document, document-level simplification will be even harder to evaluate. This is why reference-less neural methods such as QUESTEVAL combined with quality estimation features for simplicity could prove useful for evaluating document-level simplification without having to resort to using reference simplifications.

Factual Consistency During our experiments, we frequently observed meaning distortion and hallucination in generated simplifications, especially related to named-entities hallucinations. Models often modify the source sentence in a way that alters the original meaning. Improving the factual consistency of Sentence Simplification models is an important direction for future studies. Evaluation models such as QUESTEVAL could be a way to quantify the amount of factual errors in generated simplifications, and thus help improve factual correctness.

12.3 Towards French FALC Simplification

Initially, we planned on having a model that could simplify texts directly in FALC in the 3 years of this thesis. We have made progress on Sentence Simplification but would require more work to reach a state where models can be used seamlessly in FALC document simplification. We highlight future directions relative to FALC in this section.

From Sentence Simplification Models to FALC Our French MUSS model can perform Sentence Simplification in French, but it is still far from the simplifications that human editors create with the FALC guidelines. Indeed, most FALC simplifications are complete rewritings of the input text, discarding the majority of the original phrasing and reformulating from scratch. While our models can perform lexical simplification, minor rephrasing, content deletion, and some sentence splitting, these modifications are still very superficial. Most of the input sentence is kept unchanged. Future work could explore ways to create Sentence Simplification models with higher level of rewriting. Such line of work would need training data with high level of rephrasing as well, but this is currently hard to find. Even our mining process that uses dense semantic sentence representations, still mines sentences that are very structurally similar. This might be due to the type of sentence embeddings used (i.e. LASER). Better semantic sentence embeddings could improve the amount of rephrasing.

Towards fully-fledged FALC Document Creation Creating FALC documents does not only involve simplification at the document-level but also improvements in readability using visual cues. FALC documents improve the document layout, fonts, colors, and adds pictograms to illustrate ideas. All of those aspects are not taken into account in text simplification research, and it would be interesting to bridge existing research in text simplification with research in human-computer interaction to allow for automatic document-simplification in this broader sense.

Integration of our Work in the Cap'FALC Tool Our work is currently being integrated in the Cap'FALC tool and will serve as a first version of assistive automatic simplification. The Cap'FALC tool will take the form of a web interface where various editing tools will help the editor in transcribing complex documents in FALC documents. It is illustrated in Figure 12.1. Our French MUSS model will provide candidate simplifications for each source sentence on editor demand. The editor can then select one of the provided simplification and reformulate it further if needed. The tool also integrates other features that we developped using expertise acquired along this thesis, such as complex word identification and long sentences detection. We hope that our work can facilitate the production of FALC documents,

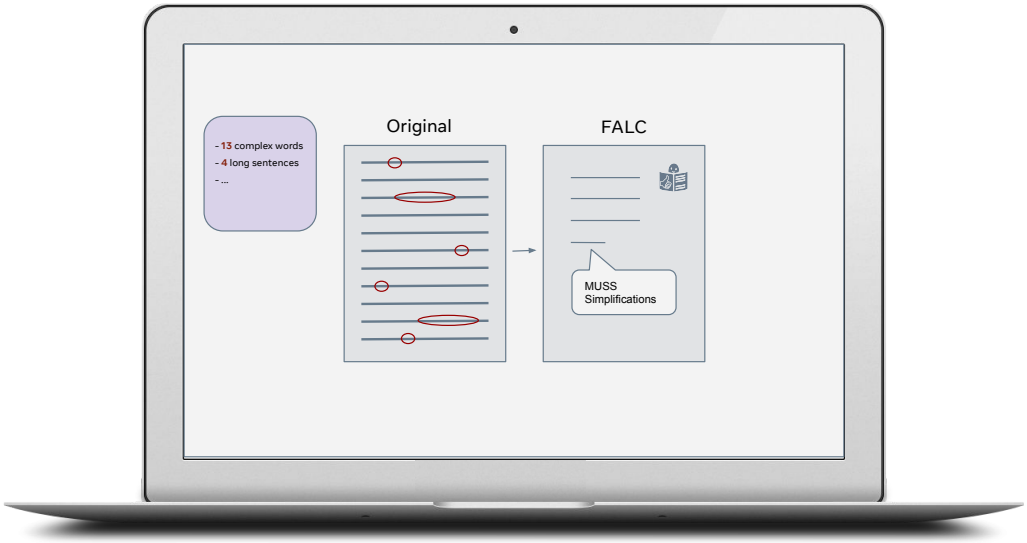


Figure 12.1: Illustration of the Cap'FALC tool interface. MUSS will propose candidate simplifications on-demand that the editor can then adapt, reformulate, or reject. Features such as complex word identification or long sentence identification will also facilitate the transcription.

and allow editors to focus on emphasizing important notions, reordering ideas, or other core aspects of FALC transcription.

Bibliography

Anne Abeillé, Lionel Clément, and François Toussenel. *Building a Treebank for French*, pages 165–187. Kluwer, Dordrecht, 2003.

Omri Abend and Ari Rappoport. Universal conceptual cognitive annotation (ucca). In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 228–238, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P13-1023.

Roee Aharoni and Yoav Goldberg. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2114. URL https://www.aclweb.org/anthology/P18-2114.

Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In Emily M. Bender, Leon Derczynski, and Pierre Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 1638–1649. Association for Computational Linguistics, 2018. ISBN 978-1-948087-50-6. URL https://www.aclweb.org/anthology/C18-1139/.

Oliver Alonzo, Matthew Seita, Abraham Glasser, and Matt Huenerfauth. Automatic text simplification tools for deaf and hard of hearing adults: Benefits of lexical simplification and providing users with autonomy. In *Proceedings of the 2020 CHI Conference on*

Human Factors in Computing Systems, CHI '20, page 1–13, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367080. doi: 10.1145/3313831.3376563. URL https://doi.org/10.1145/3313831.3376563.

Oliver Alonzo, Jessica Trussell, Becca Dingman, and Matt Huenerfauth. Comparison of methods for evaluating complexity of simplified texts among deaf and hard-of-hearing adults at different literacy levels. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, CHI '21, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380966. doi: 10.1145/3411764.3445038. URL https://doi.org/10.1145/3411764.3445038.

Sandra Aluísio and Caroline Gasperin. Fostering digital inclusion and accessibility: The PorSimples project for simplification of Portuguese texts. In *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*, pages 46–53, Los Angeles, California, June 2010. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W10-1607.

Sandra M. Aluísio, Lucia Specia, Thiago A. S. Pardo, Erick G. Maziero, Helena M. Caseli, and Renata P. M. Fortes. A corpus analysis of simple account texts and the proposal of simplification strategies: First steps towards text simplification systems. In *Proceedings of the 26th Annual ACM International Conference on Design of Communication*, SIGDOC '08, pages 15–22, Lisbon, Portugal, 2008. ACM. ISBN 978-1-60558-083-8. doi: 10.1145/1456536.1456540. URL http://doi.acm.org/10.1145/1456536.1456540.

Sandra M Aluísio, Lucia Specia, Thiago AS Pardo, Erick G Maziero, and Renata PM Fortes. Towards Brazilian Portuguese automatic text simplification systems. In *Proceedings of the eighth ACM symposium on Document engineering*, pages 240–248, 2008.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 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, November 2017.

Asian Federation of Natural Language Processing. URL https://www.aclweb.org/anthology/I17-1030.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 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, 2019a.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy, August 2019b. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. Data-driven sentence simplification: Survey and benchmark. *Computational Linguistics*, 46(1):135–187, 2020. doi: 10.1162/coli_a_00370. URL https://doi.org/10.1162/coli_a_00370.

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853, 2005. URL http://jmlr.org/papers/v6/ando05a.html.

Alessio Palmero Aprosio, Sara Tonelli, Marco Turchi, Matteo Negri, and Mattia A Di Gangi. Neural text simplification in low-resource conditions using weak supervision. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 37–44, 2019.

Mikel Artetxe and Holger Schwenk. Margin-based parallel corpus mining with multilingual sentence embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy, July 2019a. Association for Computational Linguistics. doi: 10.18653/v1/P19-1309. URL https://www.aclweb.org/anthology/P19-1309.

Mikel Artetxe and Holger Schwenk. Massively multilingual sentence embeddings for zero-

shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610, 2019b.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1399. URL https://www.aclweb.org/anthology/D18-1399.

Nguyen Bach, Qin Gao, Stephan Vogel, and Alex Waibel. Tris: A statistical sentence simplifier with log-linear models and margin-based discriminative training. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 474–482, Chiang Mai, Thailand, November 2011. Asian Federation of Natural Language Processing. URL http://www.aclweb.org/anthology/I11-1053.

Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Gianni Barlacchi and Sara Tonelli. Ernesta: A sentence simplification tool for children's stories in italian. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 476–487. Springer, 2013.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://www.aclweb.org/anthology/W19-5301.

Regina Barzilay and Noemie Elhadad. Sentence Alignment for Monolingual Comparable Corpora. In *Proceedings of the 2003 Conference on Empirical Methods in Nat-*

ural Language Processing, EMNLP '03, pages 25–32, Stroudsburg, PA, USA, 2003. Association for Computational Linguistics. doi: 10.3115/1119355.1119359. URL http://dx.doi.org/10.3115/1119355.1119359.

Regina Barzilay and Lillian Lee. Learning to paraphrase: an unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL-HLT 2003*, pages 16–23, 2003.

Rachel Bawden, Marie-Amélie Botalla, Kim Gerdes, and Sylvain Kahane. Correcting and validating syntactic dependency in the spoken French treebank rhapsodie. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2320–2325, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/766_Paper.pdf.

Hanna Béchara, Hernani Costa, Shiva Taslimipoor, Rohit Gupta, Constantin Orasan, Gloria Corpas Pastor, and Ruslan Mitkov. Miniexperts: An svm approach for measuring semantic textual similarity. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 96–101, 2015.

Michaël Benesty. Ner algo benchmark: spacy, flair, m-bert and camembert on anonymizing french commercial legal cases, December 2019. URL https://towardsdatascience.com/benchmark-ner-algorithm-d4ab01b2d4c3?source=friends_link&sk=5bffa2cb19997d1658479f18ce8cf6bb.

Or Biran, Samuel Brody, and Noemie Elhadad. 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*, pages 496–501, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-2087.

Steven Bird and Edward Loper. NLTK: The natural language toolkit. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217, Barcelona, Spain,

July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P04-3031.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017. doi: 10.1162/tacl_a_00051. URL https://www.aclweb.org/anthology/Q17-1010.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 272–303, Belgium, Brussels, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6401. URL https://www.aclweb.org/anthology/W18-6401.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1080. URL https://www.aclweb.org/anthology/D18-1080.

Stefan Bott and Horacio Saggion. An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 20–26, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics. ISBN 9781937284053. URL http://dl.acm.org/citation.cfm?id=2107679.2107682.

Stefan Bott and Horacio Saggion. Spanish text simplification: An exploratory study. *Procesamiento del Lenguaje Natural*, 47:87–95, 2011b.

Stefan Bott, Horacio Saggion, and Simon Mille. Text simplification tools for Spanish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1665–1671, Istanbul, Turkey, May 2012. European

Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2012/pdf/762_Paper.pdf.

Laetitia Brouwers, Delphine Bernhard, Anne-Laure Ligozat, and Thomas François. Syntactic sentence simplification for french. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 47–56, 2014.

Peter F. Brown, Vincent J. Della Pietra, Peter V. de Souza, Jennifer C. Lai, and Robert L. Mercer. Class-based n-gram models of natural language. *Computational Linguistics*, 18 (4):467–479, 1992.

Dominique Brunato, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. Design and Annotation of the First Italian Corpus for Text Simplification. In *Proceedings of the 9th Linguistic Annotation Workshop*, LAW IX, pages 31–41, Denver, Colorado, 2015. Association for Computational Linguistics.

Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi. Is this sentence difficult? do you agree? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2690–2699, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1289. URL https://www.aclweb.org/anthology/D18-1289.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. Concreteness ratings for 40 thousand generally known English word lemmas. *Behavior research methods*, 46(3): 904–911, 2014.

Marie Candito and Benoit Crabbé. Improving generative statistical parsing with semi-supervised word clustering. In *Proc. of IWPT'09*, Paris, France, 2009.

Marie Candito and Djamé Seddah. Le corpus sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical (the sequoia corpus : Syntactic annotation and use for a parser lexical domain adaptation method) [in french]. In *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN, Grenoble, France, June*

4-8, 2012, pages 321–334, 2012. URL https://www.aclweb.org/anthology/F12-2024/.

Marie Candito, Guy Perrier, Bruno Guillaume, Corentin Ribeyre, Karën Fort, Djamé Seddah, and Éric Villemonte de la Clergerie. Deep syntax annotation of the sequoia french treebank. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asunción Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation, LREC 2014, Reykjavik, Iceland, May 26-31, 2014.*, pages 2298–2305. European Language Resources Association (ELRA), 2014. URL http://www.lrec-conf.org/proceedings/lrec2014/summaries/494.html.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. Practical simplification of english newspaper text to assist aphasic readers. In *Proceedings of AAAI-98 Workshop on Integrating Artificial Intelligence and Assistive Technology*, pages 7–10, Madison, Wisconsin, 1998.

Branden Chan, Timo Möller, Malte Pietsch, Tanay Soni, and Chin Man Yeung. German bert. https://deepset.ai/german-bert, 2019.

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

Ping Chen, Fei Wu, Tong Wang, and Wei Ding. A semantic qa-based approach for text summarization evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Sumit Chopra, Michael Auli, and Alexander M Rush. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98, 2016.

James Clarke and Mirella Lapata. Global inference for sentence compression: An integer

linear programming approach. *Journal of Artificial Intelligence Research*, 31:273–381, 2008.

Jacob Cohen. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220, 1968.

Trevor Cohn and Mirella Lapata. An abstractive approach to sentence compression. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 4(3):1–35, 2013.

Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: evaluating cross-lingual sentence representations. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2475–2485. Association for Computational Linguistics, 2018. ISBN 978-1-948087-84-1. URL https://www.aclweb.org/anthology/D18-1269/.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Édouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, 2020.

William Coster and David Kauchak. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 665–669, Stroudsburg, PA, USA, 2011a. Association for Computational Linguistics. ISBN 978-1-932432-88-6. URL http://dl.acm.org/citation.cfm?id=2002736.2002865.

William Coster and David Kauchak. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 1–9, Portland, Oregon, 2011b. ACL. ISBN 9781937284053. URL http://dl.acm.org/citation.cfm?id=2107679.2107680.

Andrew M. Dai and Quoc V. Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pages 3079–3087, 2015. URL http://papers.nips.cc/paper/5949-semi-supervised-sequence-learning.

Jan De Belder and Marie-Francine Moens. Text Simplification for Children. In *Proceedings of the SIGIR 2010 Workshop on Accessible Search Systems*, pages 19–26, Geneva, 2010. ACM.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. Robbert: a dutch roberta-based language model. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3255–3265, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Multilingual bert. https://github.com/google-research/bert/blob/master/multilingual.md, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

Siobhan Devlin and John Tait. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173, 1998.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 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, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1331.

Timothy Dozat and Christopher D. Manning. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Hk95PK9le.

Yoann Dupont. Exploration de traits pour la reconnaissance d'entit'es nomm'ees du français par apprentissage automatique. In *24e Conf'erence sur le Traitement Automatique des Langues Naturelles (TALN)*, page 42, 2017.

Martin d'Hoffschmidt, Wacim Belblidia, Quentin Heinrich, Tom Brendlé, and Maxime Vidal. Fquad: French question answering dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 1193–1208, 2020.

Richard Evans, Constantin Orasan, and Iustin Dornescu. 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*, PIT 2014, pages 131–140, Gothenburg, Sweden, 2014. Association for Computational Linguistics.

Angela Fan, David Grangier, and Michael Auli. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-2706. URL https://www.aclweb.org/anthology/W18-2706.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*, 2019.

Thibault Fevry and Jason Phang. Unsupervised sentence compression using denoising auto-encoders. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 413–422, 2018.

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104, 2017.

Rudolph Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221, 1948.

Thomas François, Adeline Müller, Eva Rolin, and Magali Norré. AMesure: A web platform to assist the clear writing of administrative texts. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 1–7, Suzhou, China, December 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.aacl-demo.1`.

Núria Gala, Anaïs Tack, Ludivine Javourey-Drevet, Thomas François, and Johannes C Ziegler. Alector: A Parallel Corpus of Simplified French Texts with Alignments of Misreadings by Poor and Dyslexic Readers. In *Proceedings of LREC 2020*, 2020.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `http://cs.jhu.edu/~ccb/publications/ppdb.pdf`.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pages 1243–1252. PMLR, 2017.

Sebastian Gehrmann, Tosin Adewumi, Karmanya Aggarwal, Pawan Sasanka Ammanamanchi, Aremu Anuoluwapo, Antoine Bosselut, Khyathi Raghavi Chandu, Miruna Clinciu, Dipanjan Das, Kaustubh D Dhole, et al. The gem benchmark: Natural language generation, its evaluation and metrics. *arXiv preprint arXiv:2102.01672*, 2021.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, 2017.

Isao Goto, Hideki Tanaka, and Tadashi Kumano. Japanese news simplification: Task design, data set construction, and analysis of simplified text. In *Proceedings of*

Machine Translation Summit XV, Vol. 1: MT Researchers' Track, pages 17–31, Miami, FL, USA, November 2015. Association for Machine Translation in the Americas. URL http://amtaweb.org/wp-content/uploads/2015/10/MTSummitXV_ResearchTrack.pdf.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. Continuous measurement scales in human evaluation of machine translation. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 33–41, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W13-2305.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation, LREC 2018, Miyazaki, Japan, May 7-12, 2018*. European Language Resources Association (ELRA), 2018. URL http://www.lrec-conf.org/proceedings/lrec2018/summaries/627.html.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/C18-1039.

Kenneth Heafield. KenLM: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197, 2011.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. A transition-based directed acyclic graph parser for ucca. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1127–1138, Vancouver,

Canada, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-1104.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1587–1596. JMLR. org, 2017.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 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, May–June 2015. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N15-1022.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does BERT learn about the structure of language? In Korhonen et al. [2019], pages 3651–3657. ISBN 978-1-950737-48-2. doi: 10.18653/v1/p19-1356. URL https://doi.org/10.18653/v1/p19-1356.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. Neural CRF model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960, Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.acl-main.709.

Hongyan Jing. Sentence reduction for automatic text summarization. In *Sixth Applied Natural Language Processing Conference*, pages 310–315, 2000.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

A Joulin, E Grave, P Bojanowski, M Douze, H Jégou, and T Mikolov. Fasttext: compressing text classification models. *arXiv preprint ArXiv:1612.03651*, 2016.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 427–431, April 2017.

Tomoyuki Kajiwara and M Komachi. Text simplification without simplified corpora. *The Journal of Natural Language Processing*, 25:223–249, 2018.

Tomoyuki Kajiwara and Mamoru Komachi. Building a Monolingual Parallel Corpus for Text Simplification Using Sentence Similarity Based on Alignment between Word Embeddings. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1147–1158, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee. URL `http://aclweb.org/anthology/C16-1109`.

Akihiro Katsuta and Kazuhide Yamamoto. Improving text simplification by corpus expansion with unsupervised learning. In *2019 International Conference on Asian Language Processing (IALP)*, pages 216–221. IEEE, 2019.

David Kauchak. 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)*, pages 1537–1546, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P13-1151`.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338, 2016.

J.P. Kincaid, R.P. Fishburne, R.L. Rogers, and B.S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical Report 8-75, Chief of Naval Technical Training: Naval Air Station Memphis., February 1975. 49 p.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302, 2015.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA, 2007. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1557769.1557821.

Philipp Koehn, Huda Khayrallah, Kenneth Heafield, and Mikel L. Forcada. Findings of the WMT 2018 shared task on parallel corpus filtering. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 726–739, October 2018. doi: 10.18653/v1/W18-6453. URL https://www.aclweb.org/anthology/W18-6453.

Philipp Koehn, Francisco Guzmán, Vishrav Chaudhary, and Juan Pino. Findings of the WMT 2019 shared task on parallel corpus filtering for low-resource conditions. In *Proceedings of the Fourth Conference on Machine Translation*, pages 54–72, 2019.

Daniel Kondratyuk. 75 languages, 1 model: Parsing universal dependencies universally. *CoRR*, abs/1904.02099, 2019. URL http://arxiv.org/abs/1904.02099.

Anna Korhonen, David R. Traum, and Lluís Màrquez, editors. *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 2019. Association for Computational Linguistics. ISBN 978-1-950737-48-2. URL https://www.aclweb.org/anthology/volumes/P19-1/.

Reno Kriz, Eleni Miltsakaki, Marianna Apidianaki, and Chris Callison-Burch. Simplification using paraphrases and context-based lexical substitution. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 207–217. Association for Computational Linguistics, June 2018. doi: 10.18653/v1/N18-1019. URL https://www.aclweb.org/anthology/N18-1019.

Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 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, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1317. URL https://www.aclweb.org/anthology/N19-1317.

Wojciech Kryscinski, Bryan McCann, Caiming Xiong, and Richard Socher. Evaluating the factual consistency of abstractive text summarization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9332–9346, 2020.

Taku Kudo. Subword regularization: Improving neural network translation models with multiple subword candidates. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75. Association for Computational Linguistics, 2018. ISBN 978-1-948087-32-2. doi: 10.18653/v1/P18-1007. URL https://www.aclweb.org/anthology/P18-1007/.

Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computa-

tional Linguistics. doi: 10.18653/v1/D18-2012. URL https://www.aclweb.org/anthology/D18-2012.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.707. URL https://www.aclweb.org/anthology/2020.acl-main.707.

Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, 1995.

Anne Lacheret, Sylvain Kahane, Julie Beliao, Anne Dister, Kim Gerdes, Jean-Philippe Goldman, Nicolas Obin, Paola Pietrandrea, and Atanas Tchobanov. Rhapsodie: a prosodic-syntactic treebank for spoken French. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 295–301, Reykjavik, Iceland, May 2014. European Language Resources Association (ELRA). URL http://www.lrec-conf.org/proceedings/lrec2014/pdf/381_Paper.pdf.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In Carla E. Brodley and Andrea Pohoreckyj Danyluk, editors, *Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Williams College, Williamstown, MA, USA, June 28 - July 1, 2001*, pages 282–289. Morgan Kaufmann, 2001. ISBN 1-55860-778-1.

Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. URL http://arxiv.org/abs/1901.07291.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016, The 2016 Conference of*

the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016, pages 260–270. The Association for Computational Linguistics, 2016. ISBN 978-1-941643-91-4. URL https://www.aclweb.org/anthology/N16-1030/.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*, 2018a.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018b.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. ALBERT: A lite BERT for self-supervised learning of language representations, 2019. URL http://arxiv.org/abs/1909.11942. arXiv preprint 1909.11942.

Hang Le, Loïc Vial, Jibril Frej, Vincent Segonne, Maximin Coavoux, Benjamin Lecouteux, Alexandre Allauzen, Benoît Crabbé, Laurent Besacier, and Didier Schwab. Flaubert: Unsupervised language model pre-training for french, 2019. arXiv : 1912.05372.

VI Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707, 1966.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 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, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W04-1013.

Xianggen Liu, Lili Mou, Fandong Meng, Hao Zhou, Jie Zhou, and Sen Song. Unsupervised paraphrasing by simulated annealing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 302–312, 2020a.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized BERT pretraining approach, 2019. URL http://arxiv.org/abs/1907.11692. arXiv preprint 1907.11692.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020b.

Jonathan Mallinson and Mirella Lapata. Controllable sentence simplification: Employing syntactic and lexical constraints. *arXiv e-prints*, art. arXiv:1910.04387, 2019.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Sentence compression for arbitrary languages via multilingual pivoting. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2453–2464, 2018.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. Zero-shot crosslingual sentence simplification. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5109–5126, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.415. URL https://www.aclweb.org/anthology/2020.emnlp-main.415.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In

Association for Computational Linguistics (ACL) System Demonstrations, pages 55–60, 2014. URL http://www.aclweb.org/anthology/P/P14/P14-5010.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazare, Éric Villemonte de la Clergerie, Antoine Bordes, and Benoît Sagot. Reference-less quality estimation of text simplification systems. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 29–38, 2018.

Louis Martin, Éric Villemonte De La Clergerie, Benoît Sagot, and Antoine Bordes. Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698, 2020a.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. Muss: Multilingual unsupervised sentence simplification by mining paraphrases. *arXiv preprint arXiv:2005.00352*, 2020b.

Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric Villemonte De La Clergerie, Djamé Seddah, and Benoît Sagot. Camembert: a tasty french language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7203–7219, 2020c.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P13-2017.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In Christopher J. C. Burges, Léon Bottou, Zoubin Ghahramani, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26:*

27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States., pages 3111–3119, 2013. URL http://papers.nips.cc/paper/5021-distributed-representations-of-words-and-phrases-and-their-composi

Tomáš Mikolov, Édouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Dragos Stefan Munteanu and Daniel Marcu. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504, 2005. doi: 10.1162/089120105775299168. URL https://www.aclweb.org/anthology/J05-4003.

Ramesh Nallapati, Bowen Zhou, Cicero dos Santos, Çağlar Gulçehre, and Bing Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 280–290, 2016.

Shashi Narayan and Claire Gardent. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 435–445, Baltimore, Maryland, June 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P14-1041.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1064. URL https://www.aclweb.org/anthology/D17-1064.

Christina Niklaus, André Freitas, and Siegfried Handschuh. Minwikisplit: A sentence splitting corpus with minimal propositions. In *Proceedings of the 12th International*

Conference on Natural Language Generation (INLG), Tokyo, Japan, October 2019. Artificial Intelligence Research Center of Japan. URL https://www.inlg2019.com/assets/papers/67_Paper.pdf.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 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, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-2036. URL https://www.aclweb.org/anthology/P19-2036.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 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, July 2017. Association for Computational Linguistics. URL http://aclweb.org/anthology/P17-2014.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. Why we need new evaluation metrics for NLG. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2241–2252, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1238. URL https://www.aclweb.org/anthology/D17-1238.

Debora Nozza, Federico Bianchi, and Dirk Hovy. What the [mask]? making sense of language-specific BERT models. *CoRR*, abs/2003.02912, 2020. URL https://arxiv.org/abs/2003.02912.

Charles Kay Ogden. *Basic English: A General Introduction with Rules and Grammar*. Kegan Paul, Trench, Trubner & Co, 1930.

Pedro Javier Ortiz Suárez, Benoît Sagot, and Laurent Romary. Asynchronous pipeline for processing huge corpora on medium to low resource infrastructures. *Challenges in the Management of Large Corpora (CMLC-7) 2019*, page 9, 2019.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-4009. URL https://www.aclweb.org/anthology/N19-4009.

Gustavo Paetzold and Lucia Specia. 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, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/S16-1085. URL https://www.aclweb.org/anthology/S16-1085.

Gustavo H. Paetzold and Lucia Specia. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 3761–3767, Phoenix, Arizona, 2016b. AAAI Press.

Gustavo H. Paetzold, Fernando Alva-Manchego, and Lucia Specia. Massalign: Alignment and annotation of comparable documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4, Tapei, Taiwan, November 2017. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/I17-3001.

Chris D Paice. Constructing literature abstracts by computer: techniques and prospects. *Information Processing & Management*, 26(1):171–186, 1990.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania, 2002. ACL. doi: 10.3115/1073083.1073135. URL http://dx.doi.org/10.3115/1073083.1073135.

Ellie Pavlick and Chris Callison-Burch. 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, August 2016. Association for Computational Linguistics. URL http://anthology.aclweb.org/P16-2024.

Ellie Pavlick and Joel Tetreault. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74, 2016. doi: 10.1162/tacl_a_00083. URL https://www.aclweb.org/anthology/Q16-1005.

Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 425–430, 2015.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

David Pellow and Maxine Eskenazi. An open corpus of everyday documents for simplification tasks. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 84–93, Gothenburg, Sweden, April 2014a. Association for Computational Linguistics. doi: 10.3115/v1/W14-1210. URL https://www.aclweb.org/anthology/W14-1210.

David Pellow and Maxine Eskenazi. Tracking human process using crowd collaboration to enrich data. In *Human Computation and Crowdsourcing: Works in Progress and Demonstration Abstracts. An Adjunct to the Proceedings of the Second AAAI Conference on Human Computation and Crowdsourcing*, pages 52–53, 2014b. URL https://www.aaai.org/ocs/index.php/HCOMP/HCOMP14/paper/viewFile/9021/9005.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 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, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1162. URL https://www.aclweb.org/anthology/D14-1162.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn A. Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics, 2018. ISBN 978-1-948087-27-8. URL https://www.aclweb.org/anthology/N18-1202/.

Sarah E. Petersen. *Natural Language Processing Tools for Reading Level Assessment and Text Simplification for Bilingual Education*. PhD thesis, University of Washington, Seattle, WA, USA, 2007. AAI3275902.

Sarah E. Petersen and Mari Ostendorf. Text simplification for language learners: a corpus analysis. In *Proceedings of the SLaTE Workshop on Speech and Language Technology in Education*, pages 69–72, Farmington, PA, USA, 2007. Carnegie Mellon University and ISCA Archive.

Slav Petrov, Dipanjan Das, and Ryan T. McDonald. A universal part-of-speech tagset. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation, LREC 2012, Istanbul, Turkey, May 23-25, 2012*, pages 2089–2096. European Language Resources Association (ELRA), 2012. ISBN 978-2-9517408-7-7. URL http://www.lrec-conf.org/proceedings/lrec2012/summaries/274.html.

Maxime Peyrard. Studying summarization evaluation metrics in the appropriate scoring

range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1502. URL https://www.aclweb.org/anthology/P19-1502.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? arXiv:1906.01502, 2019.

Matt Post. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium, 2018. Association for Computational Linguistics. URL http://aclweb.org/anthology/W18-6319.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, 2016.

J. Rapin and O. Teytaud. Nevergrad - A gradient-free optimization platform. https://GitHub.com/FacebookResearch/Nevergrad, 2018.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. Simplify or help?: text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, page 15. ACM, 2013.

Miguel Rios and Serge Sharoff. Large scale translation quality estimation. In *The Proceedings of the 1st Deep Machine Translation Workshop*, 2015.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280, 2020.

Alexander M Rush, Sumit Chopra, and Jason Weston. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389, 2015.

Horacio Saggion and Thierry Poibeau. Automatic text summarization: Past, present and future. In *Multi-source, multilingual information extraction and summarization*, pages 3–21. Springer, 2013.

Horacio Saggion, Elena Gómez Martínez, Esteban Etayo, Alberto Anula, and Lorena Bourg. Text simplification in simplext. making text more accessible. *Procesamiento del lenguaje natural*, 47:341–342, 2011.

Horacio Saggion, Sanja Štajner, Stefan Bott, Simon Mille, Luz Rello, and Biljana Drndarevic. Making it simplext: Implementation and evaluation of a text simplification system for spanish. *ACM Transactions on Accessible Computing*, 6(4):1–36, 2015.

Benoît Sagot, Marion Richard, and Rosa Stern. Annotation référentielle du corpus arboré de Paris 7 en entités nommées (referential named entity annotation of the paris 7 french treebank) [in french]. In Georges Antoniadis, Hervé Blanchon, and Gilles Sérasset, editors, *Proceedings of the Joint Conference JEP-TALN-RECITAL 2012, volume 2: TALN, Grenoble, France, June 4-8, 2012*, pages 535–542. ATALA/AFCP, 2012. URL `https://www.aclweb.org/anthology/F12-2050/`.

Manuela Sanguinetti and Cristina Bosco. PartTUT: The Turin University Parallel Treebank. In Roberto Basili, Cristina Bosco, Rodolfo Delmonte, Alessandro Moschitti, and Maria Simi, editors, *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*, volume 589 of *Studies in Computational Intelligence*, pages 51–69. Springer, 2015. ISBN 978-3-319-14205-0. doi: 10.1007/978-3-319-14206-7_3. URL `https://doi.org/10.1007/978-3-319-14206-7_3`.

Carolina Scarton and Lucia Specia. 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, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2113. URL https://www.aclweb.org/anthology/P18-2113.

Carolina Scarton, Alessio Palmero Aprosio, Sara Tonelli, Tamara Martín Wanton, and Lucia Specia. MUSST: A multilingual syntactic simplification tool. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 25–28, Tapei, Taiwan, November 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/I17-3007.

Carolina Scarton, Gustavo H. Paetzold, and Lucia Specia. Simpa: A sentence-level simplification corpus for the public administration domain. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4333–4338, Miyazaki, Japan, may 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE, 2012.

Max Schwarzer and David Kauchak. Human evaluation for text simplification: The simplicity-adequacy tradeoff. In *Southern California Natural Language Processing Symposium*, 2018. URL https://pdfs.semanticscholar.org/76d7/f28362f81be856ef38c142ec9b78d154088b.pdf.

Holger Schwenk, Guillaume Wenzek, Sergey Edunov, Edouard Grave, and Armand Joulin. Ccmatrix: Mining billions of high-quality parallel sentences on the web. *arXiv:1911.04944*, 2019.

Thomas Scialom, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Answers unite! unsupervised metrics for reinforced summarization models. In *2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3237–3247. Association for Computational Linguistics, 2019.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Coldgans: Taming language gans with cautious sampling strategies. *Advances in Neural Information Processing Systems 33*, 2020a.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. Discriminative adversarial search for abstractive summarization. *The Thirty-Seven International Conference on Machine Learning*, 2020b.

Thomas Scialom, Paul-Alexis Dray, Patrick Gallinari, Sylvain Lamprier, Benjamin Piwowarski, Jacopo Staiano, and Alex Wang. Questeval: Summarization asks for fact-based evaluation. *arXiv preprint arXiv:2103.12693*, 2021.

Abigail See, Peter J Liu, and Christopher D Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, 2017.

Abigail See, Stephen Roller, Douwe Kiela, and Jason Weston. What makes a good conversation? how controllable attributes affect human judgments. 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 1702–1723, 2019.

Amit Seker, Amir More, and Reut Tsarfaty. Universal morpho-syntactic parsing and the contribution of lexica: Analyzing the onlp lab submission to the conll 2018 shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 208–215, 2018.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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, June 2016a. Association for Computational Linguistics. doi: 10.18653/v1/N16-1005. URL https://www.aclweb.org/anthology/N16-1005.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 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, 2016b.

Matthew Shardlow. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing 2014*, 4(1), 2014. doi: 10.14569/SpecialIssue.2014.040109. URL http://dx.doi.org/10.14569/SpecialIssue.2014.040109.

Advaith Siddharthan. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109, 2006.

Matthew G Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. Ter-plus: paraphrase, semantic, and alignment enhancements to translation edit rate. *Machine Translation*, 23(2-3):117–127, 2009.

Lucia Specia, Sujay Kumar Jauhar, and Rada Mihalcea. Semeval-2012 task 1: English lexical simplification. In ** SEM 2012: The First Joint Conference on Lexical and Computational Semantics–Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 347–355, 2012.

Lucia Specia, Kashif Shah, Jose GC Souza, and Trevor Cohn. Quest-a translation quality estimation framework. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 79–84, 2013.

Lúcia Specia, Sandra Maria Aluísio, and Thiago A. Salgueiro Pardo. Manual de simplificação sintática para o português. Technical Report NILC-TR-08-06, NILC–ICMC–USP, São Carlos, SP, Brasil, 2008. Available in http://www.nilc.icmc.usp.br/nilc/download/NILC_TR_08_06.pdf.

Sanja Štajner, Hannah Béchara, and Horacio Saggion. A deeper exploration of the standard pb-smt approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 823–828, 2015a.

Sanja Štajner, Iacer Calixto, and Horacio Saggion. Automatic text simplification for Spanish: Comparative evaluation of various simplification strategies. In *Proceedings of the international conference recent advances in natural language processing*, 2015b.

Milan Straka. Udpipe 2.0 prototype at conll 2018 ud shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207, 2018.

Milan Straka, Jana Straková, and Jan Hajic. Evaluating contextualized embeddings on 54 languages in POS tagging, lemmatization and dependency parsing, 2019. URL http://arxiv.org/abs/1908.07448. arXiv preprint 1908.07448.

Jana Straková, Milan Straka, and Jan Hajic. Neural architectures for nested NER through linearization. In Korhonen et al. [2019], pages 5326–5331. ISBN 978-1-950737-48-2. URL https://www.aclweb.org/anthology/P19-1527/.

Elior Sulem, Omri Abend, and Ari Rappoport. Semantic structural evaluation for text simplification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 685–696, New Orleans, Louisiana, 2018a. Association for Computational Linguistics. URL http://aclweb.org/anthology/N18-1063.

Elior Sulem, Omri Abend, and Ari Rappoport. 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, 2018b. Association for Computational Linguistics. URL http://aclweb.org/anthology/D18-1081.

Md Sultan, Steven Bethard, and Tamara Sumner. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230, 2014. ISSN 2307-387X. URL https://transacl.org/ojs/index.php/tacl/article/view/292.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy, July 2019. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/P19-1198.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism Bulletin*, 30(4):415–433, 1953.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://www.aclweb.org/anthology/P19-1452.

Jörg Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, LREC'12, pages 2214–2218, Istanbul, Turkey, 2012a.

Jörg Tiedemann. Parallel Data, Tools and Interfaces in OPUS. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings*

of the Eight International Conference on Language Resources and Evaluation, 2012b. ISBN 978-2-9517408-7-7.

Sara Tonelli, Alessio Palmero Aprosio, and Francesca Saltori. SIMPITIKI: a Simplification corpus for Italian. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Napoli, Italy, 2016. URL http://ceur-ws.org/Vol-1749/paper52.pdf.

Sara Tonelli, Alessio Palmero Aprosio, and Marco Mazzon. The impact of phrases on italian lexical simplification. In *Proceedings of the Fourth Italian Conference on Computational Linguistics (CLiC-it 2017)*, pages 316–320, 2017.

Jenine Turner and Eugene Charniak. Supervised and unsupervised learning for sentence compression. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 290–297, 2005.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: Bert for finnish, 2019. arXiv preprint 1912.07076.

Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 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, 2018. Association for Computational Linguistics. URL http://aclweb.org/anthology/N18-2013.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. Facilita: Reading assistance for low-literacy readers. In *Proceedings of the 27th ACM International Conference on Design of Communication*, SIGDOC '09, pages 29–36, Bloomington, Indiana, USA, 2009. ACM. ISBN 978-1-60558-559-8. doi: 10.1145/1621995.1622002. URL http://doi.acm.org/10.1145/1621995.1622002.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Édouard Grave. Ccnet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4003–4012, 2020.

John Wieting and Kevin Gimpel. ParaNMT-50M: Pushing the limits of paraphrastic sentence embeddings with millions of machine translations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 451–462, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1042. URL https://www.aclweb.org/anthology/P18-1042.

Adina Williams, Nikita Nangia, and Samuel Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1101. URL https://www.aclweb.org/anthology/N18-1101.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256, May 1992. ISSN 0885-6125. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew.

Huggingface's transformers: State-of-the-art natural language processing, 2019. arXiv preprint 1910.03771.

Kristian Woodsend and Mirella Lapata. Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/D11-1038.

Shijie Wu and Mark Dredze. Beto, bentz, becas: The surprising cross-lingual effectiveness of BERT. 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 833–844, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1077. URL https://www.aclweb.org/anthology/D19-1077.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 1015–1024, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=2390524.2390660.

Menglin Xia, Ekaterina Kochmar, and Ted Briscoe. Text readability assessment for second language learners. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 12–22, 2016.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297, 2015. URL http://www.cis.upenn.edu/~ccb/publications/publications/new-data-for-text-simplification.pdf.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. *Transactions of the Association*

for Computational Linguistics, 4:401–415, 2016. ISSN 2307-387X. URL https://www.transacl.org/ojs/index.php/tacl/article/view/741.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237, 2019. URL http://arxiv.org/abs/1906.08237.

Taha Yasseri, András Kornai, and János Kertész. A practical approach to language complexity: A wikipedia case study. *PLOS ONE*, 7(11):1–8, 11 2012. doi: 10.1371/journal.pone.0048386. URL https://doi.org/10.1371/journal.pone.0048386.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 365–368, Los Angeles, California, June 2010. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/N10-1056.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Neurips*, 2020.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkeHuCVFDr.

Xingxing Zhang and Mirella Lapata. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/D17-1063.

Yaoyuan Zhang, Zhenxu Ye, Yansong Feng, Dongyan Zhao, and Rui Yan. A constrained sequence-to-sequence neural model for sentence simplification. *CoRR*, abs/1704.02312, 2017. URL https://arxiv.org/abs/1704.02312.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 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, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1355. URL https://www.aclweb.org/anthology/D18-1355.

Yanbin Zhao, Lu Chen, Zhi Chen, and Kai Yu. Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, 2020.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 1353–1361, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics. URL http://dl.acm.org/citation.cfm?id=1873781.1873933.

Sanja Štajner, Ruslan Mitkov, and Horacio Saggion. 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)*, pages 1–10, Gothenburg, Sweden, April 2014. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W14-1201.

Sanja Štajner, Maja Popović, and Hannah Béchera. Quality estimation for text simplification. In *Proceeding of the Workshop on Quality Assessment for Text Simplification - LREC 2016*, QATS 2016, pages 15–21, Portorož, Slovenia, 2016a. European Language Resources Association (ELRA).

Sanja Štajner, Maja Popović, Horacio Saggion, Lucia Specia, and Mark Fishel. Shared task on quality assessment for text simplification. In *Proceeding of the Workshop on Quality Assessment for Text Simplification - LREC 2016*, QATS 2016, pages 22–31, Portorož, Slovenia, 2016b. European Language Resources Association (ELRA).

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. CATS: A tool for customized alignment of text simplification corpora. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3895–3903, Miyazaki, Japan, may 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.