The University Of Sheffield.

Automatic Sentence Simplification with Multiple Rewriting Transformations

By:

Fernando Emilio Alva Manchego

A thesis submitted in partial fulfilment of the requirements for the degree of
Doctor of Philosophy

The University of Sheffield
Faculty of Engineering
Department of Computer Science

September 2020

Declaration

I, the author, confirm that the Thesis is my own work. I am aware of the University's Guidance on the Use of Unfair Means (`www.sheffield.ac.uk/ssid/unfair-means`). This work has not been previously been presented for an award at this, or any other, university.

Fernando Emilio Alva Manchego
September 2020

Acknowledgements

Getting a PhD is one of the biggest challenges I have had to tackle. Impostor syndrome and I met early on, so finding my place within the research community (and believing that I belong here) required hard work, adjusting my expectations and learning to trust more in my abilities. Of course, finishing up experiments and writing a Thesis while there is a global pandemic outside my door did not make things any easier by the end. However, despite the difficulties, it was definitely a rewarding experience that made me grow both professionally and personally. And that was, in no small amount, thanks to the support I received from several people.

My supervisors Lucia Specia and Carolina Scarton. Thank you for your patience, guidance and trust during all these years. I am very grateful that on top of knowledge and research experience, you also offered advice and understanding, especially in times of struggle. You inspired me to come all the way from Peru to the UK, and I really appreciate you allowing me to work with and learn from you. Muito obrigado!

My friends and colleagues from SheffieldNLP. Thank you Fred, Alison, Haiyang, Gustavo, Zeerak, James, Felipe, Katerina, George, Hardy, Chiraag, Pranava, Josiah, Thales, Nikos, Loïc and Andreas. I will always value the time we spent together in the lab, whether it was discussing some research papers, or simply chatting while sharing some food.

Many other people I met along the way. Thank you Yorgos, Carl, Cristina, Marcin, Yousif, Helen, Reham, Sahand and Fran for the fun times in the streets, cafés, restaurants and pubs in Sheffield. Thank you to the people from LxMLS 2017 for the great experience that Lisbon was. And a big thank you to Sam, for his patience and support during the last year of PhD. I do not think I would have been able to cope with the stress of Thesis writing and being away from home during a pandemic if I did not have you with me. Thank you for keeping me sane.

And last, but certainly not least, my family. Thank you for the support when I decided to move across the pond, for the video chats and the messages to catch up and make me feel your love when I am not home, and for the hugs and kisses when we are together. When I was a kid I told you a wanted to be a Doctor, and while this may not be exactly what I had in mind at the time, I still managed to become one. I would not have been able to reach this goal without having you by my side. ¡Muchas gracias!

Abstract

Sentence Simplification aims to rewrite a sentence in order to make it easier to read and understand, while preserving as much as possible of its original meaning. In order to do so, human editors perform several text transformations, such as replacing complex terms by simpler synonyms, reordering words or phrases, removing non-essential information, and splitting long sentences. However, executing these rewriting operations automatically while keeping sentences grammatical, preserving their main idea, and generating simpler output, is a challenging and still far from solved problem. Considering that simplifications produced by humans encompass a variety of text transformations, we should expect automatic simplifications to be produced in a similar fashion. However, current data-driven models for the task leverage datasets that do not necessarily contain training instances that exhibit this variety of operations. As such, they tend to copy most of the original content, with only small changes focused on lexical paraphrasing. Furthermore, it is unclear whether this implicit learning of multi-operation simplifications results in automatic outputs with such characteristics, since current automatic evaluation resources (i.e. metrics and test sets) focus on single-operation simplifications. In this Thesis, we tackle these limitations in Sentence Simplification research in four aspects.

First, we develop novel annotation algorithms that are able to identify the simplification operations that were performed by automatic models at word, phrase and sentence levels. We propose to use these algorithms in an operation-based error analysis method, that measures the correctness of executing specific operations based on reference simplifications. This functionality is incorporated into EASSE, our new software package for standard automatic evaluation of simplification systems. We use EASSE to benchmark several simplification systems, and show that our proposed operation-based error analysis serves to better understand the scores computed using automatic metrics.

Second, we introduce ASSET, a new multi-reference dataset for tuning and evaluation of Sentence Simplification models. Reference simplifications in ASSET were produced by human editors applying multiple rewriting transformations. We show that simplifications in ASSET offer more variability than other commonly-used evaluation datasets. In addition, we perform a human evaluation study that demonstrates that multi-operation simplifications are judged simpler than single-operation ones. We also motivate the need to develop new metrics suitable for multi-operation simplification assessment, since we show that judgements on simplicity do not have strong correlations with commonly-used multi-reference metrics computed using multi-operation simplification references.

Third, we carry out the first meta-evaluation of automatic evaluation metrics in Sentence Simplification. We collect a new more reliable dataset for evaluating the behaviour of metrics and human judgements of simplicity. We use this data (and other existing datasets) to analyse the variation of the correlation of automatic metrics and simplicity judgements across three dimensions: the perceived simplicity level, the system type and the set of references used for computation. We show that these three aspects affect the correlations and, in particular, highlight the limitations of commonly-used simplification-specific metrics. Based on our findings, we elaborate a set of recommendations for automatic evaluation of multi-operation simplification, indicating which metrics to compute and how to interpret their scores.

Finally, we implement MulTSS, a multi-operation Sentence Simplification model based on a multi-task learning architecture. We leverage training data from related text rewriting tasks (lexical paraphrasing, extractive compression and split-and-rephrase) to enhance the multi-operation capabilities of a standard simplification model. We show that our multi-task approach can generate better simplifications than strong single-task and pipeline baselines.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Simplicity is a complex concept to define. When reading a text, what is simple to understand for an adult with higher education is not the same for a child, or for a person with a lower literacy level. However, we can generally agree in what makes a text simpler than another. For example, let's consider the following:[1]

(1) Owls are the order Strigiformes, comprising 200 bird of prey species. Owls hunt mostly small mammals, insects, and other birds though some species specialize in hunting fish.

(2) An owl is a bird. There are about 200 kinds of owls. Owls' prey may be birds, large insects (such as crickets), small reptiles (such as lizards) or small mammals (such as mice, rats, and rabbits).

Most people would agree that (2) is easier to understand or **simpler** than (1). Some characteristics that make this possible are:

- Some unusual concepts are explained with examples: insects *(such as crickets)*, small reptiles *(such as lizards)* or small mammals *(such as mice, rats, and rabbits).*
- Some uncommon words are replaced with a more familiar term or phrase: *comprising* in (1) is replaced with *There are about* in (2).
- Some syntactic structures are changed by a simpler pattern. For example, the first clause in (1) (*Owls are the order ... species.*) is split into the first two clauses in (2).

[1] These examples were extracted from a public presentation by Prof. Mirella Lapata available in: `http://videolectures.net/esslli2011_lapata_simplification/`

- Some “unimportant” information is removed: *though some species specialize in hunting fish* in (1) does not appear in (2).

In the example above, (1) is more demanding for a reader without adequate vocabulary or knowledge of animal species, or who finds multi-clause sentences harder to comprehend. In contrast, (2) provides a more accessible version of the same content, which could allow the reader to understand it better. Therefore, we can define **Text Simplification** as the task of **modifying the content and structure** of a text, in order to make it **easier to understand** while **retaining its main idea** and **approximating its original meaning**. The essential goal of Text Simplification is to increase a text’s *readability*, which in turn could improve its *understandability* (Shardlow, 2014). This means that if a text’s grammatical structure and vocabulary make it easy to read (i.e. it is readable), then the reader could probably gain more information from it (i.e. it is understandable).

Different types of readers could benefit from a simplified version of a text. Mason and Kendall (1978) report that splitting a complex sentence into shorter structures can improve comprehension in low-ability readers. Siddharthan (2014) refers to studies on hard-of-hearing children that show their difficulty dealing with complex structures, like coordination, subordination and pronominalisation (Quigley et al., 1977), or passive voice and relative clauses (Robbins and Hatcher, 1981). Shewan (1985) states that adults suffering from aphasia reduce their comprehension of a sentence as its grammatical complexity increases. An eye-tracking study (Rello et al., 2013a) determined that people with dyslexia read faster if more frequent words are used in a sentence, and also that shorter words improve their understanding of the text. Crossley et al. (2007) point out that simplified texts are mostly used for teaching beginners and intermediate English learners.

Motivated by the benefits stated above, some research has focused on developing Text Simplification systems for specific target audiences: writers of simplified texts (Candido Jr. et al., 2009), low-literacy readers (Watanabe et al., 2009), ESL learners (Petersen, 2007), non-native English speakers (Paetzold, 2016), children (De Belder and Moens, 2010), and people suffering from aphasia (Carroll et al., 1998; Devlin and Tait, 1998), dyslexia (Rello et al., 2013b) or autism (Evans et al., 2014). Furthermore, simplifying a text automatically could improve performance on other Natural Language Processing tasks, such as parsing (Chandrasekar et al., 1996), summarisation (Siddharthan et al., 2004; Silveira and Branco, 2012; Vanderwende et al., 2007), information extraction (Beigman Klebanov et al., 2004), relation extraction (Miwa et al., 2010; Niklaus et al., 2016), semantic role labelling (Vickrey and Koller, 2008), and machine translation (Hasler et al., 2017; Mirkin et al., 2013; Mishra et al., 2014; Štajner and Popovic, 2016). We refer the interested reader to Siddharthan (2014) for a more in-depth review of studies on the benefits of simplification for different target audiences and NLP applications.

A text can be simplified at the level of words/phrases, individual sentences or complete documents. **Lexical Simplification**, the first level, consists of replacing words or phrases that are difficult to understand with more common synonyms (Shardlow, 2014) without altering the meaning of the sentence. **Sentence Simplification** adds a syntactic component by also identifying complex grammatical structures in the sentence, and changing them to simpler ones (Shardlow, 2014) by, for example, reordering the elements of the sentence or splitting the sentence into two or more simpler ones. It is also possible to delete "unnecessary" information, or to add explanations for difficult concepts, as shown in the example at the beginning of the Chapter. **Document Simplification**, in turn, adds a discourse component by taking into account the relationships between the sentences in the text, and performing cross-sentence operations, such as joining sentences, reordering (parts of) sentences across the text, or making coreference entities explicit (Alva-Manchego et al., 2019b).

The focus of this Thesis is on Sentence Simplification; in particular, approaches for the task that attempt to learn and perform multiple simplification operations simultaneously, independently of their linguistic nature. Examples of these operations are lexical paraphrasing (i.e. replacing complex terms and minimal word reordering), compression (i.e. deletion of "unnecessary" information) and sentence splitting. We do not include addition of explanations, since performing this simplification operation requires access to external resources (like knowledge bases), turning it into a special task on its own.

Most research approaches treat Sentence Simplification as monolingual Machine Translation (MT), with *original* and *simplified* as *source* and *target* languages, respectively, using phrase-based (Coster and Kauchak, 2011b; Specia, 2010; Wubben et al., 2012), syntax-based (Bach et al., 2011; Xu et al., 2016; Zhu et al., 2010) and neural sequence-to-sequence (Dong et al., 2019; Guo et al., 2018; Kriz et al., 2019; Martin et al., 2020; Nisioi et al., 2017; Scarton and Specia, 2018; Vu et al., 2018; Zhang and Lapata, 2017; Zhao et al., 2018) approaches. Others have used sets of hand-crafted rules (Bott et al., 2012; Candido Jr. et al., 2009; Siddharthan and Mandya, 2014), while some try to extract rewriting rules automatically from parallel corpora (Feblowitz and Kauchak, 2013; Paetzold and Specia, 2013; Woodsend and Lapata, 2011a).

It is unclear to what extent the so-far proposed methods are able to perform multiple simplification operations simultaneously. Manual error analyses on sampled system outputs have shown that Phrase-Based MT models were only successful on executing lexical paraphrasing, unless coupled with very expensive semantic processing (Narayan and Gardent, 2014, 2016). Some Syntax-Based MT models have been tailored to perform sentence splitting, but at the cost of complex training design with no significant gain in performance for syntactic simplifications (Zhu et al., 2010). Neural sequence-to-sequence architectures allow for implementing end-to-end models that could implicitly learn multiple simplification operations. However, so far they have only been evaluated in their ability to perform specific operations like lexical paraphrasing (Xu et al., 2016) or sentence splitting (Sulem et al., 2018a); or used for related tasks involving syntactic changes, like split-and-rephrase (Aharoni and Goldberg, 2018; Botha et al., 2018; Narayan et al., 2017), but without considering simplification as an end goal.

It is important to evaluate the output of these automatic systems. Human judgements on grammaticality, meaning preservation and simplicity are the most reliable assessment method (Štajner et al., 2016b). However, since they are costly to produce, researchers rely on automatic MT-inspired metrics (e.g. BLEU (Papineni et al., 2002)), readability metrics (e.g. Flesch Kincaid (Kincaid et al., 1975)) and simplicity metrics (e.g. SARI (Xu et al., 2016)) for quicker quantitative analysis. These metrics have limitations. In the case of BLEU, Sulem et al. (2018a) determined that it is unsuitable for assessing structural simplicity (i.e. sentence splitting), while Xu et al. (2016) showed that it rewarded conservative systems. Whilst the latter makes BLEU correlate well with judgements of grammaticality and meaning preservation (also pointed out by Wubben et al. (2012); Štajner et al. (2014)), it cannot assess changes to the input sentences that improved their simplicity. In the case of SARI, Xu et al. (2016) only showed that it correlated with human scores of simplicity gain when just lexical paraphrasing was being assessed. Therefore, it remains an open question which automatic metrics will be more appropriate to use in a multi-operation simplification scenario. In addition, the aforementioned metrics evaluate the overall quality of the simplifications, without informing on how the simplification models handle particular simplification operations.

1.2 Objectives

This Thesis studies the problem of automatically simplifying sentences using multiple types of rewriting operations holistically. Overall, the following research questions are addressed by the approaches proposed in this Thesis:

1. **Which simplification operations do automatic systems perform accurately?** The usual method for evaluating Sentence Simplification models uses automatic metrics to estimate the overall quality of the simplifications produced in terms of their similarity to human references. However, these scores do not provide insights about the effectiveness of each approach when performing specific simplification operations. We designed

novel algorithms to automatically identify the operations carried out in parallel original-simplified sentences. These annotations are used as a form of automatic error analysis to benchmark Sentence Simplification models using per-operation scores that help better understand the results provided by standard evaluation metrics.

2. **Do we need multi-reference multi-operation evaluation datasets to support better assessment of Sentence Simplification models?** Sentence Simplification models are evaluated in datasets that are either multi-operation but single-reference, such as Newsela (Xu et al., 2015) and PWKP (Zhu et al., 2010), or multi-reference but single-operation, such as TurkCorpus (Xu et al., 2016) and HSplit (Sulem et al., 2018a). This limits our ability to assess them in multi-operation scenarios using metrics that (preferably) rely on multiple references (e.g. BLEU and SARI). We crowdsourced a new evaluation dataset that contains manual references produced by executing several simplification operations, namely lexical paraphrasing, compression and sentence splitting. We also collected human judgements on the quality of these simplifications that highlight their superiority in terms of simplicity over other datasets. Furthermore, we showed that commonly-used automatic metrics may be unsuitable for assessing multi-operation simplifications.

3. **Can current evaluation metrics measure the ability of automatic systems to perform multi-operation simplifications?** It is unclear if commonly-used metrics, such as BLEU, SARI or SAMSA, are reliable when evaluating automatic multi-operation simplifications. In order to better understand their behaviour in this simplification scenario, we conducted the first meta-evaluation of automatic metrics for Sentence Simplification. We studied the variation in the correlation of newly-collected human judgements on simplicity with automatic metrics across three dimensions: the quality of the automatic output, the system type, and the set of references. We determined that all three aspects affect the correlations and, in particular, further analysed the behaviour of simplification-specific metrics SARI and SAMSA. We then elaborated a set of recommendations for evaluating multi-operation simplification models, such as the use of multiple metrics including BERTScore (Zhang et al., 2020), and how to interpret the automatic scores.

4. **Can we devise models that leverage data with more types of rewriting to enhance the multi-operation capabilities of simplification systems?** Most neural sequence-to-sequence models for Sentence Simplification are trained on automatic sentence alignments extracted from related articles in English Wikipedia (EW) and Simple EW, such as WikiLarge (Zhang and Lapata, 2017). These simplification instances are known to be

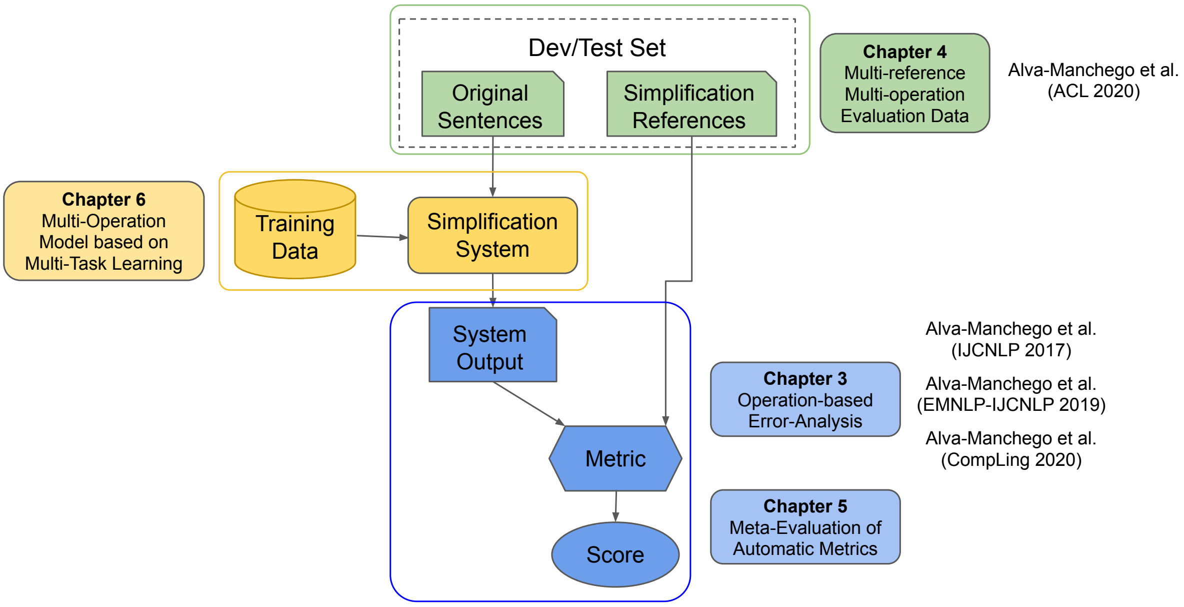

Dev/Test Set
Original Sentences
Simplification References
Chapter 4
Multi-reference Multi-operation Evaluation Data
Alva-Manchego et al. (ACL 2020)
Chapter 6
Multi-Operation Model based on Multi-Task Learning
Training Data
Simplification System
System Output
Metric
Score
Alva-Manchego et al. (IJCNLP 2017)
Chapter 3
Operation-based Error-Analysis
Alva-Manchego et al. (EMNLP-IJCNLP 2019)
Alva-Manchego et al. (CompLing 2020)
Chapter 5
Meta-Evaluation of Automatic Metrics

Figure 1.1 Contributions of this Thesis.

noisy and display limited types of rewriting (Xu et al., 2015). We argue that we could overcome these shortcomings by combining this training data with that from related rewriting tasks through a multi-task framework. We exploited Johnson et al. (2017)'s architecture for multilingual neural MT to jointly train a model in four text generation tasks: Sentence Simplification, Lexical Paraphrasing, Compression and Sentence Splitting. According to automatic metrics, our multi-task model performed better than strong single-task and pipeline baselines for Sentence Simplification. In particular, we showed that training in a multi-task way achieves better results than simply pooling all data together.

1.3 Contributions

The main contributions of this Thesis, depicted in Figure 1.1, span across the whole process of implementing a Sentence Simplification system: training of a model, tuning and testing on adequate evaluation data, and analysis of its performance using automatic metrics.

Operation-based Error Analysis

1. Novel algorithms to annotate the simplification operations performed in aligned original-simplified sentences (Alva-Manchego et al., 2017). Based on word alignments and

constituent-based parse trees, these algorithms identify deletions, replacements, reorderings, additions, and sentence splits at the word, phrase and sentence levels, accordingly.

2. EASSE (Alva-Manchego et al., 2019a), a software package for standard automatic evaluation of sentence-level simplification output. It provides access to (re-)implementations of popular automatic metrics, supplemented with operation-based error analyses, and straightforward access to commonly-used evaluation datasets.

3. A benchmark comparing the simplification capabilities of Sentence Simplification systems (Alva-Manchego et al., 2020b). This study is based on standard automatic metrics, and on identifying which simplification operations each system can execute correctly.

Multi-Reference Multi-operation Evaluation Data

4. A methodology for crowdsourcing multi-operation simplifications that includes detailed annotation guidelines and adequate quality control mechanisms.

5. ASSET (Alva-Manchego et al., 2020a), a new dataset for tuning and evaluation of Sentence Simplification models, whose manual references were produced by executing multiple simplification operations: lexical paraphrasing, compression and sentence splitting. It contains 10 high-quality multi-operation simplification references for 2,359 original sentences (2,000 for the development set and 359 for the test set).

6. An automatic analysis highlighting the diversity of simplification operations in ASSET compared to TurkCorpus and HSplit.

7. Crowdsourced human preference judgements comparing multi-operation simplifications in ASSET and single-operation simplifications in TurkCorpus and HSplit, with respect to fluency, meaning preservation and simplicity. In total, we collected 1,077 judgements (359 sentence pairs * 3 aspects) and showed that ASSET's simplifications are simpler.

8. A preliminary study on the reliability of BLEU and SARI to evaluate system outputs when using simplifications in ASSET as references. We crowdsourced 15 human ratings on fluency, meaning preservation and simplicity for 100 automatic simplifications, totalling 4,500 ratings (100 sentences * 15 ratings * 3 aspects).

Meta-Evaluation of Automatic Metrics

9. A new dataset (Simplicity-DA) for evaluation of automatic metrics comprising human judgements for 600 automatic simplifications generated by six different systems. We

collected 15 ratings per simplification instance on grammaticality, meaning preservation and simplicity, based on the Direct Assessment methodology (Graham et al., 2017).

10. The first meta-evaluation of automatic metrics in Sentence Simplification. We determined that the correlation between metrics and human judgements of simplicity is affected by three conditions: the perceived simplicity level, the system type, and the simplification references used to compute the metrics. For instance, in our Simplicity-DA dataset, metrics can more reliably score low-quality simplifications, are better at scoring system outputs from neural sequence-to-sequence models, and do not have their correlations significantly affected when using more references than those in ASSET.

11. A preliminary experiment on the benefits of selecting the references to use for computing automatic metrics on a sentence-by-sentence basis, according to the operations attempted by the simplification system.

12. A qualitative analysis on the weaknesses of simplification-specific metrics SARI and SAMSA to measure Simplicity Gain and Structural Simplicity, respectively.

13. A set of recommendations for automatic evaluation of simplification models, particularly for multi-operation simplification scenarios.

14. A desiderata for the characteristics of new metrics and resources for automatic evaluation.

Multi-Operation Model based on Multi-task Learning

15. A multi-task model for multi-operation Sentence Simplification. We repurposed Johnson et al. (2017)'s architecture for multilingual neural MT to train a joint model for Sentence Simplification, Lexical Paraphrasing, Compression and Sentence Splitting.

16. Empirical evidence that training a multi-operation model in a multi-task framework is more beneficial than simply pooling together the data from all tasks, or executing single-task models in a pipeline fashion.

Survey

18. A comprehensive survey on data-driven Sentence Simplification that details, compares and discusses resources for the task, evaluation methodologies, and different approaches to build automatic systems (Alva-Manchego et al., 2020b).

1.4 Publications

Some parts of this Thesis have been published and presented in international peer-reviewed venues:

- A version of our literature review on Sentence Simplification (Chap. 2) together with our systems' benchmark (Sec. 3.3.2) was published in the *Computational Linguistics* journal as a survey article (Alva-Manchego et al., 2020b).
- The word-level algorithms for annotation of simplification operations (Sec. 3.2) were part of joint research on modelling Sentence Simplification as sequence labelling, which was published as a long paper in IJCNLP 2017 (Alva-Manchego et al., 2017). The aforementioned algorithms were also included in MASSAlign, a tool for sentence alignment and annotation, which was published as a demo in IJCNLP 2017 (Paetzold et al., 2017).
- Our work on standard automatic evaluation of Sentence Simplification systems (Sec. 3.3.2) was published as part of a demo in EMNLP-IJCNLP 2019 (Alva-Manchego et al., 2019a).
- Our new dataset for tuning and evaluation of Sentence Simplification systems comprising multi-reference multi-operation simplifications (Chap. 4), was published as a long paper in ACL 2020 (Alva-Manchego et al., 2020a).

We have also published and presented contributions related to Text Simplification, but not included in this Thesis:

- Joint work on monolingual and cross-lingual models for Complex Word Identification was published as a short paper in NAACL 2019 (Finnimore et al., 2019).
- Manual and quantitative analyses of cross-sentence simplification operations, together with some preliminary experiments on document-level simplification, were presented as a poster in WiNLP 2019 (Alva-Manchego et al., 2019b).

1.5 Thesis Structure

Chapter 2 presents a literature review on approaches for Sentence Simplification. We focus on methods that attempt to learn how to rewrite sentences from parallel corpora (i.e. data-driven). Without lost of generality, the review is limited to research on texts written in English.

Chapter 3 details our work on operation-based error analysis for Sentence Simplification evaluation. We first introduce algorithms that annotate simplification operations automatically in different granularities. These algorithms are then used to quantify the accuracy in performing the annotated operations by several Sentence Simplification models. Moreover, we benchmark these models with commonly-used evaluation metrics and test sets, and we show that our operation-based analysis complements the information provided by the standard scores.

Chapter 4 introduces ASSET, a new dataset for tuning and evaluation of Sentence Simplification systems. We describe the methodology followed to crowdsource manual multi-operation simplifications, allowing to improve over current publicly-available evaluation datasets that are focused on only one type of operation. Through quantitative and qualitative experiments, we show that ASSET contains simplifications that present more varied sets of rewriting operations, and that are consider simpler than those in other evaluation corpora as judged by humans.

Chapter 5 presents a meta-evaluation of automatic evaluation metrics for Sentence Simplification. We describe the collection of a new dataset with human judgements on the quality of automatic simplifications following the Direct Assessment methodology (Graham et al., 2017). We use this dataset and those from (Xu et al., 2016) and (Sulem et al., 2018c) to analyse the variation of the correlations between metrics and human judgements along three dimensions: the perceived simplicity, the system type and the set of references. Based on our findings, we elaborate a set of recommendations for evaluating automatic multi-operation simplifications.

Chapter 6 describes our multi-task approach for multi-operation Sentence Simplification. We train a joint model with data from Sentence Simplification and from related rewriting tasks (Lexical Paraphrasing, Compression and Sentence Splitting) that could collaborate to improve the simplicity of sentences. We evaluate our proposed approach with automatic metrics, mainly, and show that it outperforms single-task and pipeline baselines.

Finally, Chapter 7 summarises our findings and provides directions for future research.

Chapter 2

Data-driven Sentence Simplification

When simplifying sentences, different rewriting operations are performed, ranging from replacing complex words or phrases for simpler synonyms, to changing the syntactic structure of the sentence (e.g. splitting or reordering components). In this Thesis, we are interested in models that simplify in a holistic process, i.e. that attempt to perform all types of simplification operations integrally, and not just lexical or syntactic alterations individually.[1] Modern Sentence Simplification approaches are **data-driven**; that is, they attempt to learn these operations using parallel corpora of aligned original-simplified sentences. This results in general simplification models that could be use for any specific type of audience depending on the data used during training. While significant progress has been made in this direction, current models are not yet able to execute the task fully automatically with the performance levels required to be directly useful for end users. As such, we believe it is important to review current research in the field, and to analyse it critically to better identify areas that could be improved. Without loss of generality, we focus on work done for sentences in English. Most of the content of this Chapter was published as part of a survey in (Alva-Manchego et al., 2020b).

We first present studies on human simplification to motivate the focus on a multi-operation approach, and explain how some of these operations are manifested in related text rewriting tasks (Sec. 2.1). Then, we detail the most commonly-used resources for Sentence Simplification research, with emphasis on corpora used to train automatic models (Sec. 2.2). After that, we explain how the output of a simplification model is generally evaluated (Sec. 2.3). Furthermore, we present a critical summary of the different approaches that have been used to train data-driven sentence-level models (Sec. 2.4). Finally, based on our analysis of the literature, we highlight the gaps in current research that are addressed in this Thesis (Sec. 2.5).

[1] For a detailed review on Lexical Simplification, we recommend Paetzold and Specia (2017b). For work focused on Syntactic Simplification, we recommend Siddharthan (2014) and Scarton et al. (2017).

2.1 Sentence-Level Simplification Operations

In this Section, we first review work on human simplification that focus on the diversity of text transformations that are performed. Based on these studies, we then comment on the relationships between Sentence Simplification and other text rewriting tasks.

2.1.1 Rewriting Operations in Manual Simplifications

A few corpus studies have been carried out to determine how humans simplify sentences. These studies shed some light on the simplification operations that an automatic Sentence Simplification model should be expected to perform.

Petersen and Ostendorf (2007) analysed a corpus of 104 original and manually-simplified news articles in English to understand how professional editors performed the simplifications, so they can later propose ways to automate the process. For their study, every sentence in the simplified version of an article was manually aligned to a corresponding sentence (or sentences) in the original version. Each original-simplified alignment was then categorised as dropped (1 to 0), split (1 to $\geq$ 2), total (1 to 1) or merged (2 to 1). Their analysis then focused on the split and dropped alignments. The authors determined that the decision to split an original sentence depends on some syntactic features (e.g. number of nouns, pronouns, verbs, etc.) and, most importantly, its length. On the other hand, the decision to drop a sentence may be influenced by its position in the text and how redundant is the information it contains.

Aluísio et al. (2008) studied six corpora of simple texts (different genres) and a corpus of non-simple news text in Brazilian Portuguese. Their analysis included counting *simple* words and discourse markers, calculating average sentence lengths, counting prepositional phrases, adjectives, adverbs, clauses, and other features. As a result, a manual for Brazilian Portuguese Sentence Simplification was elaborated, containing a set of rules to perform the task (Specia et al., 2008). In addition, as part of the same project, Caseli et al. (2009) implemented a tool to aid manual Sentence Simplification considering the following operations: non-simplification, simple rewriting, strong rewriting (similar content but very different writing), subject-verb-object reordering, passive to active voice transformation, clause reordering, sentence splitting, sentence joining, and full or partial sentence dropping.

Bott and Saggion (2011) 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 operations performed: change (e.g. difficult words, pronouns, voice of verb), delete (words, phrases or clauses), insert (word or phrases), split (relative clauses, coordination, etc.), proximization (add locative phrases, change from third to second person), reorder, select, and join (sentences). The first four operations were the most common.

Even though each study used texts in a different language, and each team of researchers devised their own taxonomy of simplification operations, some commonalities can be identified. At the lexical level, both Aluísio et al. (2008) and Bott and Saggion (2011) included operations that change words or phrases with simpler synonyms. At the sentence level, those two groups included the change of voice and reordering of components, while all three teams considered sentence splitting as an important structural change. Going beyond the boundaries of a sentence, all three teams included joining sentences. Finally, all three teams also considered the deletion of parts or complete sentences as a valid simplification operation.

2.1.2 Related Text Rewriting Tasks

In Natural Language Processing (NLP) research, there are several text rewriting tasks that could be considered related to (or subtasks of) Sentence Simplification.

Simplification could easily be confused with **summarisation**. As Shardlow (2014) points out, summarisation focuses on reducing length and content by removing unimportant or redundant information. As shown in the previous section, some deletion of content can also be performed while simplifying sentences. However, that is not the main goal: deletion should only happen when it results in a simpler text. In addition, we could replace words by more explanatory phrases, make coreferences explicit, add connectors to improve fluency, among other changes. As a consequence, a simplified text could end up being longer than its original version, while still improving the readability of the text. To sum up, although summarisation and simplification are related, they pursue different objectives.

Another related task is **sentence compression**, which consists of reducing the length of a sentence without losing its main idea and keeping it grammatical (Jing, 2000). Most approaches focus on deleting unnecessary words. Therefore, this could be considered as a subtask of the simplification process, which also encompasses more complex operations that can change the grammatical structure of the sentence. **Abstractive sentence compression** (Cohn and Lapata, 2013), on the other hand, can include operations like substitution, reordering and insertion. However, the goal is still to reduce content without necessarily improving readability.

Split-and-rephrase (Narayan et al., 2017) focuses on splitting a sentence into several shorter ones, and making the necessary rephrasings to preserve meaning and grammaticality. As shown in the previous section, sentence splitting is an important simplification operation (among others) that improves the structural simplicity of a sentence. As such, split-and-rephrase could be considered as another possible rewriting operation within simplification.

These text rewriting tasks have particular goals (e.g. reducing the length of a text, or performing a specific structural change) that differ from the main objective in Simplification: to improve readability. However, they are closely-related to the simplification operations that previous research has identified. As such, they can be seen as substasks in Sentence Simplification that, when executed jointly, can generate simpler texts.

2.2 Corpora for Simplification

A data-driven Sentence Simplification model is one that learns to simplify from examples in corpora. In particular, for learning sentence-level operations, a model requires instances of original sentences and their corresponding simplified versions. In this Section, we present the most commonly-used resources for Sentence Simplification that provide these examples, including parallel corpora and dictionary-like databases. For each parallel corpus, especially, we outline the motivations behind it, how the much-necessary sentence alignments were extracted, and report on studies about the suitability of the resource for Sentence Simplification research.

As presented in Sec. 2.1.1, an original sentence could be aligned to one (1-to-1) or more (1-to-N) simplified sentences. At the same time, several original sentences could be aligned to a single simplified one (N-to-1). The corpora we describe in this Section contain many of these types of alignments. In the remaining of this Chapter, we use the term **simplification instance** to refer to any type of sentence alignment in a general way.

2.2.1 Main - Simple English Wikipedia

The Simple English Wikipedia (SEW)[2] is a version of the online English Wikipedia (EW)[3] primarily aimed at English learners, but that can also be beneficial for students, children and adults with learning difficulties (Simple Wikipedia, 2017b). With this purpose, articles in SEW use fewer words and simpler grammatical structures. For example, writers are encouraged to use the list of words of Basic English (Ogden, 1930), which contains 850 words presumed to be sufficient for everyday life communication. Authors also have guidelines on how to create syntactically simple sentences by, for example, giving preference to the subject-verb-object order for their sentences, and avoiding compound sentences (Simple Wikipedia, 2017a).

[2] https://simple.wikipedia.org

[3] https://wikipedia.org

Simplification Instances

Much of the popularity of using Wikipedia for research in Sentence Simplification comes from publicly-available automatically-collected alignments between sentences of *equivalent* articles in EW and SEW. Several techniques have been explored to produce such alignments with reasonable quality.

A first approach consists of aligning texts according to their tf-idf cosine similarity. For the **PWKP** corpus, Zhu et al. (2010) measured this directly at sentence-level between all sentences of each article pair, and sentences whose similarity was above a certain threshold were aligned. For the **C&K-1** (Coster and Kauchak, 2011b) and **C&K-2** (Kauchak, 2013) corpora, the authors first aligned paragraphs with tf-idf cosine similarity, and then found the best overall sentence alignment with the dynamic programming algorithm proposed by Barzilay and Elhadad (2003). This algorithm takes context into consideration: the similarity between two sentences is affected by their proximity to pairs of sentences with high similarity. Finally, Woodsend and Lapata (2011a) also adopt the two step process of Coster and Kauchak (2011b), using tf-idf when compiling the **AlignedWL** corpus.

Another approach is to take advantage of the revision histories in Wikipedia articles. When editors change the content of an article, they need to comment on what the change was and the reason for it. For the **RevisionWL** corpus, Woodsend and Lapata (2011a) looked for keywords *simple*, *clarification* or *grammar* in the revision comments of articles in SEW. Then, they used Unix commands `diff` and `dwdiff` to identify modified sections and sentences, respectively, to produce the alignments. This approach is inspired by Yatskar et al. (2010), who used a similar method to extract high-quality lexical simplifications (e.g. *collaborate* → *work together*).

More sophisticated techniques for measuring sentence similarity have also been explored. For their **EW-SEW** corpus, Hwang et al. (2015) implemented an alignment method using word-level semantic similarity based on Wiktionary[4]. They first created a graph using synonym information and word-definition co-occurrence in Wiktionary. Then, similarity is measured based on the number of shared neighbours between words. This word-level similarity metric is then combined with a similarity score between dependency structures. This final similarity rate is used by a greedy algorithm that forces 1-to-1 matches between original and simplified sentences. Kajiwara and Komachi (2016) propose several similarity measures based on word embeddings alignments. Given two sentences, their best metric (1) finds, for each word in one sentence, the word that is most similar to it in the other sentence, and (2) averages the similarities for all words in the sentence. For symmetry, this measure is calculated twice

[4] Wiktionary is a free dictionary in the format of a wiki so that everyone can add and edit words' definitions. Available in `https://en.wiktionary.org`

(simplified → original, original → simplified) and their average is the final similarity measure between the two sentences. This metric was used to align original and simplified sentences from articles in a 2016 Wikipedia dump, and produce the **sscorpus**. It contains 1-to-1 alignments from sentences whose similarity was above a certain threshold.

The alignment methods described have produced different versions of parallel corpora from EW and SEW, which are currently used for research in Sentence Simplification. Table 2.1 summarises some of their characteristics.

Table 2.1 Summary of parallel corpora extracted from EW and SEW. An original sentence can be aligned to one (1-to-1) or more (1-to-N) unique simplified sentences. A (*) indicates that some aligned simplified sentences may not be unique.

Corpora	Instances	Alignment Types
PWKP (Zhu et al., 2010)	108K	1-to-1, 1-to-N
C&K-1 (Coster and Kauchak, 2011b)	137K	1-to-1, 1-to-N, N-to-1
RevisionWL (Woodsend and Lapata, 2011a)	15K	1-to-1*, 1-to-N*, N-to-1*
AlignedWL (Woodsend and Lapata, 2011a)	142K	1-to-1, 1-to-N
C&K-2 (Kauchak, 2013)	167K	1-to-1, 1-to-N, N-to-1
EW-SEW (Hwang et al., 2015)	392K	1-to-1
sscorpus (Kajiwara and Komachi, 2016)	493K	1-to-1
WikiLarge (Zhang and Lapata, 2017)	286K	1-to-1*, 1-to-N*, N-to-1*

RevisionWL is the smallest parallel corpus listed and its instances may not be as clean as those of the others. A 1-to-1* alignment means that an original sentence can be aligned to a simplified one that appears more than once in the corpus. A 1-to-N* alignment means that an original sentence can be aligned to several simplified sentences, but some (or all of them) repeat more than once in the corpus. Lastly, a N-to-1* alignment means that several original sentences can be aligned to one simplified sentence that repeats more than once in the corpus. This sentence repetition is indicative of misalignments, which makes this corpus noisy.

EW-SEW and sscorpus provide the largest number of instances. These corpora also specify a similarity score per aligned sentence pair, which can help filter out instances with less confidence to reduce noise. Unfortunately, they only contain 1-to-1 alignments. Despite being smaller in size, PWKP, C&K-1, C&K-2, and AlignedWL also offer 1-to-N alignments, which is desirable if we want a SS model to learn how to split sentences.

Finally, **WikiLarge** (Zhang and Lapata, 2017) joins instances from four Wikipedia-based datasets: PWKP, C&K-2, AlignedWL and RevisionWL. It is the most common corpus used for training neural sequence-to-sequence models for Sentence Simplification (see Sec. 2.4.4). However, it is not the biggest in size currently available, and can contain noisy alignments.

Suitability for Simplification Research

Several studies have been carried out to determine the characteristics that make Wikipedia-based corpora suitable (or unsuitable) for the simplification task.

Some research has focused on determining if SEW is actually simple. Yasseri et al. (2012) conducted a statistical analysis on a dump of the whole corpus from 2010 and concluded that, even though SEW articles use fewer complex words and shorter sentences, their syntactic complexity is basically the same as EW (as compared by part-of-speech n-gram distribution).

Other studies target the automatic alignments used to train Sentence Simplification models. Coster and Kauchak (2011b) found that in their C&K-1 corpus, the majority (65%) of simple paragraphs do not align to an original one, and even between aligned paragraphs not every sentence is aligned. Also, around 27% of instances are identical, which could induce Sentence Simplification models to learn to not modify an original sentence, or to perform very conservative rewriting transformations. Xu et al. (2015) analysed 200 randomly selected instances of the PWKP corpus and found that around 50% of the alignments are not real simplifications. Some of them (17%) correspond to misalignments and, on the others (33%), the simple sentence presents the same level of complexity as its counterpart. Instances formed by identical sentence pairs could provide models with a signal of when simplification is not needed. However, this is only valuable if the original sentence was already simple-enough. Otherwise, these instances (together with the misalignments) add noise to the training data, and prevent models from learning how to perform the necessary simplification operations.

Another line of research tries to determine the simplification operations realised in available parallel data. Coster and Kauchak (2011b) used word alignments on C&K-1 and found rewordings (65%), deletions (47%), reorders (34%), merges (31%), and splits (27%). Amancio and Specia (2014) extracted 143 instances also from C&K-1, and manually annotated the simplification operations performed: sentence splitting, paraphrasing (either single word or whole sentence), drop of information, sentence reordering, information insertion, and misalignment. They found that the most common operations were paraphrasing (39.8%) and drop of information (26.76%). Xu et al. (2015) categorised the *real* simplifications they encountered in PWKP according to the simplification performed, and found: deletion only (21%), paraphrase only (17%) and deletion+paraphrase (12%). These results show a tendency towards lexical simplification and compression operations. Also, Xu et al. (2015) state that the simplifications found are not ideal, since many of them are minimal: just a few words are simplified (replaced or dropped) and the rest is left unchanged.

These studies evidence problems with instances in corpora extracted from EW and SEW alignments. Noisy data in the form of misalignments as well as lack of variety of simplification operations can lead to suboptimal Sentence Simplification models that learn to simplify from these corpora. However, their scale and public availability are strong assets and simplification models have been shown to learn to perform some simplifications (albeit still with mistakes) from this data. Therefore, this is still an important resource for research in the area. One promising direction is to devise ways to mitigate the effects of the noise in the data.

2.2.2 Newsela Corpus

In order to tackle some of the problems identified in EW and SEW alignments, Xu et al. (2015) introduced the Newsela corpus. It contains 1,130 news articles with up to five simplified versions each: the original text is version 0 and the most simplified version is 5. The target audience considered was children with different education grade levels. These simplifications were produced manually by professional editors, which is an improvement over SEW where volunteers performed the task. A manual analysis of 50 random automatically aligned sentence pairs (reproduced in Figure 2.1) shows a better presence and distribution of simplification operations in the Newsela corpus.

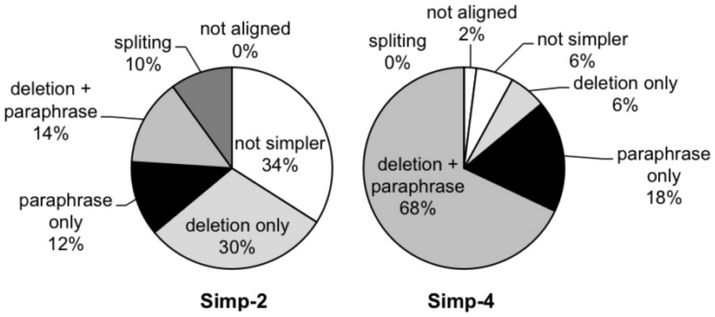


Figure 2.1 Manual categorisation of simplification operations in sampled sentences from two simplified versions in the Newsela corpus. Simp-N means sentences from the original article (version 0) automatically-aligned to sentences in version-N of the same article. Extracted from Xu et al. (2015).

.

The statistics of Figure 2.1 show that there is still a preference towards compression and lexical substitution operations, rather than more complex syntactic alterations. However, splitting starts to appear in early simplification versions. In addition, just like with EW and SEW, there are sentences which are not simpler than their counterparts in the previous version. This is likely to be because they did not need any further simplifications to comply with the readability requirements of the grade level of the current version.

Xu et al. (2015) also presented an analysis of the most frequent syntax patterns in original and simplified texts for PWKP and Newsela. These patterns correspond to *parent node (head* *node)* → *children node(s)* structures. Overall, the Wikipedia corpus has a higher tendency to retain complex patterns in their simple counterpart than Newsela. Finally, the authors present a study on discourse connectives that are important for readability according to Siddharthan (2003). They report that simple cue words are more likely to appear in Newsela's simplifications, and that complex connectives have a higher probability to be retained in Wikipedia's. This could enable research on how discourse features influence simplification.

Simplification Instances

Newsela is a corpus that can be obtained for free for research purposes[5], but it cannot be redistributed. As such, it is not possible to produce and release sentence alignments for the research community in Sentence Simplification. This is certainly a disadvantage, since it is difficult to compare models developed using this corpus without a common split of the data and the same document, paragraph, and sentence alignments.

Xu et al. (2015) align sentences between consecutive versions of articles in the corpus using Jaccard similarity (Jaccard, 1912) based on overlapping word lemmas. Alignments with the highest similarity become simplification instances.

Štajner et al. (2017) explore three similarity metrics and two alignment methods to produce paragraph and sentence alignments in Newsela. The first similarity metric uses a character 3-gram model (Mcnamee and Mayfield, 2004) with cosine similarity. The second metric averages the word embeddings (trained in EW) of the text snippet and then uses cosine similarity. The third metric computes the cosine similarity between all word embeddings in the text snippet (instead of the average). Regarding the alignment methods, the first one uses any of the previous metrics to compute the similarity between all possible sentence pairs in a text and chooses the pair of highest similarity as the alignment. The second method uses the previous strategy first, but instead of choosing the pair with highest similarity, assumes that the order of sentences of the original text is preserved in its simplified version, and thus chooses the sequence of sentence alignments that best supports this assumption. The produced instances were evaluated based on human judgements for 10 original texts with three of their corresponding simplified versions. Their best method measures similarity between text snippets with the character 3-gram model and aligns using the first strategy. Even though the alignments are not publicly available, the algorithms and metrics to produce them can be found in CATS (Štajner et al., 2018)[6].

The vicinity-driven algorithms of Paetzold and Specia (2016c) are used in (Alva-Manchego et al., 2017; Scarton et al., 2018b; Scarton and Specia, 2018) to generate paragraph and sentence

[5] `https://newsela.com/data/`

[6] `https://github.com/neosyon/SimpTextAlign`

alignments between consecutive versions of articles in Newsela. Given two documents/paragraphs, their method first creates a similarity matrix between all paragraphs/sentences using tf-idf cosine similarity. Then, it selects a coordinate in the matrix that is closest to the beginning [0,0] and that corresponds to a pair of text snippets with a similarity score above a certain threshold. From this point on, it iteratively searches for good alignments in a hierarchy of vicinities: V1 (1-1, 1-N, N-1 alignments), V2 (skipping one snippet) and V3 (long-distance skips). They first align paragraphs and then sentences within each paragraph. The extracted sentence alignments correspond to 1-to-1, 1-to-N, and N-to-1 instances. The alignment algorithms are publicly-available as part of the MASSAlign toolkit (Paetzold et al., 2017).[7]

Since articles in the Newsela corpus have different simplified versions that correspond to different grade levels, models using paragraph or sentence alignments between consecutive versions (e.g. 0→1, 1→2, 2→3, etc.) may learn different simplification operations than those using non-consecutive versions (e.g. 0→2, 1→3, 2→4, etc.). In addition, using non-consecutive versions with the same original side (e.g. 0→1, 0→2, 0→3, etc.) results in having instances where the same original sentence has simplifications in different grade levels. This could confuse models with respect to the right type of operations that should be performed. All these nuances are important to keep in mind when learning from automatic alignments of this corpus.

Suitability for Simplification Research

Scarton et al. (2018b) studied automatically-aligned sentences from the Newsela corpus in order to determine its suitability for Sentence Simplification. They first analysed the corpus in terms of readability and psycholinguistic metrics, determining that each version of an article is indeed simpler than the previous one. They then used the sentences to train models for four tasks: complex vs. simple classification, complexity prediction, lexical simplification, and sentence simplification. The dataset proved useful for the first three tasks, and helped achieve the highest reported performance for a state-of-the-art lexical simplifier. Results for the last task were inconclusive, indicating that more in-depth studies need to be performed, and that research intending to use Newsela for Sentence Simplification needs to be mindful about the types of sentence alignments to use for training models.

2.2.3 Other Resources for English

In this Section, we describe some additional resources that are used for Sentence Simplification in English with very specific purposes: tuning and testing of models in general purpose

[7] https://github.com/ghpaetzold/massalign

(TurkCorpus) and domain-specific (SimPA) data, evaluation of sentence splitting (HSplit), readability assessment (OneStopEnglish), training and testing of split-and-rephrase (WEBSPLIT and WikiSplit), and learning paraphrases (PPDB and SPPDB).

TurkCorpus

Just like with other text rewriting tasks, there is no single correct simplification possible for a given original sentence. As such, Xu et al. (2016) asked workers on Amazon Mechanical Turk to simplify 2,359 sentences extracted from the PWKP corpus to collect eight references for each one. This corpus was then randomly split into two sets: one with 2,000 instances intended to be used for system tuning, and one with 359 instances for measuring the performance of Sentence Simplification models using metrics that rely on multiple references (see SARI in Sec. 2.3.2). However, the instances chosen from PWKP are those that focus on lexical paraphrasing (1-to-1 alignments with almost similar lengths), thus limiting the range of simplification operations that models can be evaluated on using this multi-reference corpus. This corpus is the most commonly used to evaluate and compare Sentence Simplification systems trained on EW data.

HSplit

Sulem et al. (2018a) created a multi-reference corpus specifically for assessing sentence splitting. They took the sentences from the test set of TurkCorpus, and manually simplified them in two settings: (1) split the original sentence as much as possible, and (2) split only when it simplifies the original sentence. Two annotators carried out the task in both settings. The authors report that, in the whole corpus, around 72% of the input original sentences were split into an average of 2.02 sentences in their simplified output.

SimPA

Scarton et al. (2018a) introduce a corpus that differs from the previous ones in two aspects: (1) it contains sentences from the Public Administration domain instead of the more general (Wikipedia) and news (Newsela) “domains”, and (2) lexical and syntactic simplifications were performed independently. The former could be useful for validation and/or evaluation of Sentence Simplification models in a different domain, while the latter allows the analysis of the performance of models in the two subtasks in isolation. The current version of the corpus contains 1,100 original sentences, each with three references of lexical simplifications only, and one reference of syntactic simplification. This syntactic simplification was performed starting from a randomly-selected lexical simplification reference for each original sentence.

OneStopEnglish

Vajjala and Lučić (2018) compiled a parallel corpus of 189 news articles that were rewritten by teachers to three levels of adult ESL (English as a Second Language) learners: elementary, intermediate, and advanced. In addition, they used cosine similarity to automatically align sentences between articles in all the levels, resulting in 1,674 instances for ELE-INT, 2,166 for ELE-ADV, and 3,154 for INT-ADV. The initial motivation for creating this corpus was to aid in automatic readability assessment at document and sentence levels. However, OneStopEnglish could also be used for testing the generalisation capabilities of models trained on bigger corpora with different target audiences.

WebSplit

Narayan et al. (2017) introduced split-and-rephrase, and created a dataset for training and testing of models attempting this task. Extracting information from the WebNLG dataset (Gardent et al., 2017), they collected WebSplit. Each entry in the dataset contains: (1) a meaning representation (MR) of an original sentence, which is a set of RDF triplets (*subject|property|object*); (2) the original sentence to which the meaning representation corresponds; and (3) several MR-sentence pairs that represent valid splits ("simple" sentences) of the original sentence. After its first release, Aharoni and Goldberg (2018) found that around 90% of unique "simple" sentences in the development and test sets also appeared in the training set. This resulted in trained models performing well due to memorisation rather than learning to split properly. Therefore, Aharoni and Goldberg proposed a new split of the data ensuring that (1) every RDF relation is represented in the training set, and that (2) every RDF triplet appears in only one of the data splits. Later, Narayan et al. released an updated version of their original dataset, with more data and following constraint (2) before.

WikiSplit

Botha et al. (2018) created a corpus for the split-and-rephrase task based on EW edit histories. In the dataset, each original sentence is only aligned to two simpler ones. A simple heuristic was employed for the alignment: the trigram prefix and trigam suffix of the original sentence should match, respectively, the trigram prefix of the first simple sentence and the trigam suffix of the second simple sentence. The two simple sentence should not have the same trigam suffix either. The BLEU score between the aligned pairs was also used to filter out misalignments according to an empirical threshold. The final corpus contains one million instances.

Paraphrase Database

Ganitkevitch et al. (2013) released the Paraphrase Database (PPDB), which contains 220 million paraphrases in English. These paraphrases are lexical (one token), phrasal (multiple tokens), and syntactic (tokens and non-terminals). To extract the paraphrases, they used bilingual corpora with the following intuition: "two strings that translate to the same foreign string can be assumed to have the same meaning". The authors employed the synchronous context-free grammar formalism to collect paraphrases. Using Machine Translation technology, they extracted grammar rules from foreign-to-English corpora. Then, the paraphrase is created from rule pairs where the left-hand side and foreign string match. Each paraphrase in PPDB has a similarity score, which was calculated using monolingual distributional similarity.

Simple Paraphrase Database

Pavlick and Callison-Burch (2016) created Simple PPDB, a subset of the PPDB tailored for Sentence Simplification. They used machine learning models to select paraphrases which generate a simplification and preserve its meaning. First, they selected 1,000 words from PPDB which also appear in the Newsela corpus. Then, they selected up to 10 paraphrases for each word, and crowd-sourced the manual evaluation of these paraphrases in two stages: (1) rate their meaning preservation in a scale of 1 to 5, and (2) label the ones with rates higher than 2 as simpler or not. After that, this data was used to train a multi-class logistic regression model to predict if a paraphrase would produce simpler, more complex or non-sense output. Finally, they applied this model to PPDB and extracted 4.5 million simplifying paraphrase rules. Each rule in Simple PPDB is composed of: (1) a *paraphrase score* that indicates the quality of the output simplified phrase; (2) a *simplification score* indicating the confidence of the regression model; (3) the *syntactic category* of the paraphrase; (4) the *input phrase* to the simplified; and (5) the *output phrase*, i.e. the simplification of the input.

2.2.4 Discussion

In this Section, we have described the most important datasets used for training and testing Sentence Simplification systems. In particular, we detailed how their simplification instances were collected, and reviewed studies that analysed their suitability for research in the area.

Simplification instances from EW-SEW and Newsela have their own advantages and disadvantages. However, the biggest shortcoming that both share is that these original-simplified sentence pairs were extracted automatically and, therefore, are noisy. In some cases, the original and simplified sides are exactly the same (or have very minimal changes) even though more simplification operations should have been performed. This encourages models to be conservative. In other cases, for the same original sentence there are multiple possible simplifications that are very different from each other. This could confuse models about the right set of simplification operations to apply for a given original sentence. Together with misalignments, these deficiencies prevent models to learn to perform the necessary operations that improve the readability of sentences. Therefore, it becomes necessary to build simplification systems that are robust to this noisy data. Even better, it would be desirable to collect simplification instances that are naturally-aligned at the sentence-level.

These noisy alignments can be even more problematic when used to evaluate the quality of automatic simplifications. As will be explained in the next section, most evaluation metrics are based on comparing system outputs to simplification references. Therefore, ensuring that these references adequately reflect how humans simply is of paramount importance. Some work has been done for collecting high-quality manual sentence simplifications, such us for TurkCorpus and HSplit. More efforts like those should be pursued to improve the reliability of references.

2.3 Evaluation of Simplification Models

The main goal in Sentence Simplification is to improve the readability and understandability of the original sentence. Independently of the technique used to simplify a sentence, the evaluation methods we use should allow us to determine how good the simplification output is for that end goal. In this Section we explain how the outputs of automatic Sentence Simplification models are typically evaluated, based on human ratings and/or using automatic metrics.

2.3.1 Human Assessment

The most reliable method to determine the quality of a simplification consists of asking human judges to rate it. It is common practice to evaluate a model's output on three criteria: grammaticality, meaning preservation and simplicity (Štajner et al., 2016b).

For **grammaticality** (or fluency), evaluators are presented with a sentence and asked to rate it with a Likert scale of 1-3 or 1-5 (most common). The lowest score indicates that the sentence is completely ungrammatical, while the highest score means that it is completely grammatical. Native or highly-proficient speakers of the language are ideal judges for this criterion.

For **meaning preservation** (or adequacy), evaluators are presented with a pair of sentences (the original and the simplification), and asked to rate (also using a Likert scale) the similarity of the meaning of the sentences. A low score denotes that the meaning is not preserved, while a high score suggests that the sentence pair share the same meaning.

For **simplicity**, evaluators are presented with an original-simplified sentence pair and are asked to rate how much simpler (or easier to understand) the simplified version is in comparison to the original version, also using a Likert scale. Xu et al. (2016) differs from this standard, asking judges to evaluate **simplicity gain**, which means to count the correct lexical and syntactic paraphrases performed. Sulem et al. (2018b) introduce the notion of **structural simplicity**, which ignores lexical simplifications and focuses on structural changes with the question: *Is the output simpler than the input, ignoring the complexity of the words?* A different approach uses labels instead of numerical scores, like in the Shared Task on Quality Assessment for Text Simplification (Štajner et al., 2016b): *bad* = very difficult to understand, *ok* = somewhat difficult to understand, and *good* = easy to understand; or in Surya et al. (2019) with a binary label to indicate whether the system output is a simplification of the original sentence or not.

These three criteria are heavily related, and the perfect simplification does not necessarily score the highest in all of them. For instance, Schwarzer and Kauchak (2018) showed that there is a trade-off between simplicity and meaning preservation. The authors crowdsourced 10 human ratings for 500 simplification instances, and determined that there is a negative correlation between these two aspects.

2.3.2 Automatic Metrics

Human evaluation is the preferred method for assessing the quality of simplifications, but its is costly to produce and may require expert annotators or end-users of a specific target audience. As such, they are impractical at the development stage of a simplification system, when we need to easily compare different architectures or tune models' parameters. Therefore, researchers turn to automatic measures as a means of obtaining faster and cheaper evaluation results. Some of these metrics are based on comparing the automatic simplifications to manually-produced references; others compute the readability of the text based on psycholinguistic metrics; while others are trained on specially annotated data so as to learn to predict the quality or usefulness of the simplification being evaluated.

String Similarity Metrics

These metrics are mostly borrowed from the Machine Translation literature, since Sentence Simplification can be seen as translating a text from complex to simple. The most commonly-used are BLEU and TER.

BLEU (**B**i**L**ingual **E**valuation **U**nderstudy), proposed by Papineni et al. (2002), is a precision-oriented metric, which means that it depends on the number of n-grams in the candidate translation that match with n-grams of the reference, independently of position. BLEU values range from 0 to 1 (or to 100); *the higher the better*.

BLEU calculates a modified n-gram precision: (i) count the maximum number of times that a n-gram occurs in any of the references, (ii) clip the total count of each candidate n-gram by its maximum reference count (i.e, $Count_{clip} = \min(Count, MaxRefCount)$), and (iii) add these clipped counts up, and divide by the total (unclipped) number of candidate words. Short sentences (compared to the lengths of the references) could inflate this modified precision. As such, BLEU uses a Brevity Penalty (BP) factor, calculated as in Equation 2.1, where c is the length of the candidate translation, r is the reference corpus length, and r/c is used in a decaying exponential (in this case, c is the total length of the candidate translation corpus).

$$BP = \begin{cases} 1 & \text{if } c > r \\ e^{(1-r/c)} & \text{if } c \leq r \end{cases} \tag{2.1}$$

The final BLEU score is computed as in Equation 2.2. Traditionally, $N = 4$ and $w_n = 1/N$.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log(p_n)\right) \tag{2.2}$$

In simplification research, several studies (Wubben et al., 2012; Xu et al., 2016; Štajner et al., 2014) show that BLEU has high correlation with human assessments of grammaticality and meaning preservation, but not simplicity. Also, Sulem et al. (2018a) show that this correlation is low or non-existent when sentence splitting has been performed. As such, BLEU should not be used as the only metric for evaluation and comparison of Sentence Simplification models. In addition, due to its definition, this metric is more useful with simplification corpora that provides multiple references for each original sentence.

TER (**T**ranslation **E**dit **R**ate), by Snover et al. (2006), measures the minimum number of necessary edits to change a candidate translation so that it matches one of the references, normalised by the average length of the references. Only the reference that is closest (according to TER) is considered for the final score. The considered edits are insertions, deletions, substitutions of single words, and shifts (positional changes) of word sequences. TER is an edit-distance metric (Equation 2.3), with values between 0 and 100; *lower values are better.*

$$TER = \frac{\text{\# of edits}}{\text{average \# of reference words}} \tag{2.3}$$

To calculate the number of shifts, TER follows a two step process: (i) use dynamic programming to count insertions, deletions and substitutions; and use a greedy search to find the set of shifts the minimises the number of those three operations; then (ii) calculate the optimal remaining edit distance using minimum-edit-distance and dynamic programming.

For Sentence Simplification research, TER's intermediate calculations (i.e. the edits counts) have been used to show the simplification operations that a model is able to perform (Zhang and Lapata, 2017). However, this is not a general practice and no studies have been conducted to verify that the edits correlate with simplification operations.

iBLEU is a variant of BLEU introduced by Sun and Zhou (2012) as a way to measure the quality of a candidate paraphrase. The metric balances the *semantic similarity* between the candidate and the reference, with the dissimilarity between the candidate and the source. Given a candidate paraphrase c, human references r_s and input text s, iBLEU is computed as in Equation 2.4, with values ranging from 0 to 1 (or to 100); *higher values are better*.

$$\text{iBLEU}(s, r_s, c) = \alpha \times \text{BLEU}(c, r_s) - (1 - \alpha) \times \text{BLEU}(c, s) \tag{2.4}$$

After empirical evaluations, the authors recommend to use a value of $\alpha \in [0.7, 0.9]$. For instance, Mallinson et al. (2017) experiment with $\alpha = 0.8$, whilst Xu et al. (2016) sets $\alpha = 0.9$.

Readability Metrics

Flesch Reading Ease (FRE, Flesch, 1948) is a metric that attempts to measure how easy a text is to understand. It is based on average sentence length and average word length. Longer sentences could imply the use of more complex syntactic structures (e.g. subordinated clauses), which makes reading harder. The same analogy applies to words: longer words contain prefixes and suffixes that present more difficulty to the reader. This metric (Equation 2.5) gives a score between 0 and 100, with *lower values indicating a higher level of difficulty*.

$$FRE = 206.835 - 1.015\left(\frac{\text{number of words}}{\text{number of sentences}}\right) - 84.6\left(\frac{\text{number of syllables}}{\text{number of words}}\right) \tag{2.5}$$

Flesch–Kincaid Grade Level (FKGL, Kincaid et al., 1975) is a recalculation of FRE, so as to correspond to grade levels in the United States (Equation 2.6). The coefficients were derived from multiple regression procedures in reading tests of 531 Navy personnel. The lowest possible value is -3.40 with no upper bound. The obtained score should be interpreted in an inverse way as for FRE, now *lower values indicate a lower level of difficulty*.

$$FKGL = 0.39\left(\frac{\text{number of words}}{\text{number of sentences}}\right) + 11.8\left(\frac{\text{number of syllables}}{\text{number of words}}\right) - 15.59 \tag{2.6}$$

FKBLEU (Xu et al., 2016) combines iBLUE and FKGL to ensure grammaticality and simplicity in the generated text. Given an output simplification O, a reference R and an input original sentence I, FKBLEU is calculated according to Equation 2.7; *higher values mean better simplifications.*

$$\begin{aligned} \text{FKBLEU} &= \text{iBLEU}(I,R,O) \times \text{FKGLdiff}(I,O) \\ \text{FKGLdiff} &= \text{sigmod}(\text{FKGL}(O) - \text{FKGL}(I)) \end{aligned} \tag{2.7}$$

Because of the way these Flesch-based metrics are computed, short sentences could get good scores, even if they are ungrammatical or non meaning preserving. As such, their values could be used to measure superficial simplicity, but not as an overall evaluation or for comparison of Sentence Simplification models (Wubben et al., 2012). Many other metrics could be used for more advanced readability assessment (McNamara et al., 2014); however, these are not commonly-used in simplification research.

Simplification Metrics

SARI (**S**ystem output **A**gainst **R**eferences and **I**nput sentence) was introduced by Xu et al. (2016) as a means to measure "how good" the words added, deleted and kept by a simplification model are. This metric compares the output of a Sentence Simplification model against multiple simplification references and the original sentence.

The intuition behind SARI is to reward models for adding n-grams that occur in any of the references but not in the input, to reward keeping n-grams both in the output and in the references, and to reward not over-deleting n-grams. SARI is the arithmetic mean of n-gram precisions and recalls for add, keep, and delete; *the higher the final value, the better*. Xu et al. (2016) show that SARI correlates with human judgements of simplicity gain. As such, this metric has become the standard measure for evaluating and comparing models' outputs.

Considering a model output O, the input sentence I, references R, and $\#_g(\cdot)$ as a binary indicator of occurrence of n-grams g in a given set, we first calculate n-gram precision $p(n)$ and recall $r(n)$ for the three operations listed (add, keep and delete):

$$p_{add}(n) = \frac{\sum_{g\in O}\min(\#_g(O\cap\bar{I}),\#_g(R))}{\sum_{g\in O}\#_g(O\cap\bar{I})} \quad ,\#_g(O\cap\bar{I}) = \max(\#_g(O)-\#_g(I),0)$$
$$r_{add}(n) = \frac{\sum_{g\in O}\min(\#_g(O\cap\bar{I}),\#_g(R))}{\sum_{g\in O}\#_g(R\cap\bar{I})} \quad ,\#_g(R\cap\bar{I}) = \max(\#_g(R)-\#_g(I),0)$$

$$p_{keep}(n) = \frac{\sum_{g\in I}\min(\#_g(I\cap O),\#_g(I\cap R'))}{\sum_{g\in I}\#_g(I\cap O)} \quad ,\#_g(I\cap O) = \min(\#_g(I),\#_g(O))$$
$$r_{keep}(n) = \frac{\sum_{g\in I}\min(\#_g(I\cap O),\#_g(I\cap R'))}{\sum_{g\in I}\#_g(I\cap R')} \quad ,\#_g(I\cap R') = \min(\#_g(I),\#_g(R)/r)$$

$$p_{del}(n) = \frac{\sum_{g\in I}\min(\#_g(I\cap\overline{O}),\#_g(I\cap\overline{R'}))}{\sum_{g\in I}\#_g(I\cap\overline{O})} \quad \begin{array}{l}\#_g(I\cap\overline{O}) = \max(\#_g(I)-\#_g(O),0)\\ \#_g(I\cap\overline{R'}) = \max(\#_g(I)-\#_g(R)/r,0)\end{array}$$

For keep and delete, R' marks n-gram counts over R with fractions. For example, if a unigram occurs 2 out of the total r references, then its count is weighted by $2/r$ when computing precision and recall. Recall is not calculated for deletions to avoid rewarding over-deleting. Finally, SARI is calculated as shown in Equation 2.8.

$$SARI = d_1F_{add} + d_2F_{keep} + d_3F_{del} \tag{2.8}$$

where $d_1 = d_2 = d_3 = 1/3$ and

$$P_{operation} = \frac{1}{k}\sum_{n=[1,..,k]} p_{operation}(n) \qquad R_{operation} = \frac{1}{k}\sum_{n=[1,..,k]} r_{operation}(n)$$
$$F_{operation} = \frac{2\times P_{operation}\times R_{operation}}{P_{operation}+R_{operation}} \qquad operation \in [del, keep, add]$$

An advantage of SARI is considering both the input original sentence and the references in its calculation. This is different from BLEU that only ponders the similarity of the output with the references. While iBLEU also uses both input and references, it compares the output against them independently, combining these scores in a way that rewards outputs that are similar to the references, but not so similar to the input. In contrast, SARI compares the output against the input sentence and references simultaneously, and rewards outputs that modify the input in ways that are expressed by the references. In addition, not all n-gram matches are considered equal: the more references "agree" with keeping/deleting certain *n*-gram, the higher the importance of the match in the score computation.

One disadvantage of SARI is the limited number of simplification operations taken into account, restricting the evaluation to only 1-to-1 lexically-paraphrased sentences. As such, it needs to be used in conjunction with other metrics or evaluation procedures when measuring the performance of a Sentence Simplification model. Also, if only one reference exists that is identical to the original sentence, and the model's output does not change the original sentence, SARI would over penalise it and give a low score. Therefore, SARI requires multiple references that are different from the original sentence to be reliable.

SAMSA (**S**implification **A**utomatic evaluation **M**easure through **S**emantic **A**nnotation) was introduced by Sulem et al. (2018b) to tackle some of the shortcomings of reference-based simplicity metrics, i.e SARI. The authors show that SARI has low correlation with human judgements when the simplification of a sentence involves structural changes, specifically sentence splitting. The new metric, on the other hand, correlates with meaning preservation and structural simplicity. Consequently, SARI and SAMSA should be used in conjunction to have a more complete evaluation of different simplification operations.

To calculate SAMSA, the original (source) sentence is semantically-parsed using the UCCA scheme (Abend and Rappoport, 2013), either manually or by the automatic parser TUPA (Hershcovich et al., 2017). The resulting graph contains the *Scenes* in the sentence (e.g. actions), as well as their corresponding *Participants*. SAMSA's premise is that a correct splitting of an original sentence should create a separate simple sentence for each UCCA *Scene* and its *Participants*. To verify this, SAMSA uses the word alignment between the original and the simplified output to count how many *Scenes* and *Participants* hold the premise. This process does not require simplification references (unlike SARI), and since the semantic parsing is only performed in the original sentence, it prevents adding parser errors of (possibly) grammatically incorrect simplified sentences produced by the Sentence Simplification model being evaluated.

Prediction-based Metrics

If reference simplifications are not available, a possible approach is to evaluate the simplicity of the simplified output sentence by itself, or compare it to the one from the original sentence.

Most approaches in this line of research attempt to **classify** a given sentence into categories that define its simplicity by extracting several features from the sentence and training a classification model. For instance, Napoles and Dredze (2010) used lexical and morpho-syntactic features to predict if a sentence was more likely to be from Main or Simple English Wikipedia.

Later on, inspired by work on Quality Estimation (QE) for Machine Translation,[8] Štajner et al. (2014) proposed to train classifiers to predict the quality of simplified sentences, with respect to grammaticality and meaning preservation. In this case, the features extracted correspond to values from metrics such as BLEU or (components of) TER. The authors proposed two tasks: (1) to classify, independently, the grammaticality and meaning preservation of sentences into three classes: bad, medium and good; and (2) to classify the overall quality of the simplification using either a set of three classes (OK, needs post-editing, discard) or two classes (retain, discard). The same approach was the main task in the *1st Quality Assessment for Text Simplification Workshop* (QATS, Štajner et al., 2016b), but automatic judgements of simplicity were also considered. There were promising results with respect to predicting grammaticality and meaning preservation, but not for simplicity or an overall quality evaluation metric. Afterwards, Martin et al. (2018) extended Štajner et al. (2014)'s work with features from Štajner et al. (2016a) to analyse how different feature groups correlate with human judgements on grammaticality, meaning preservation and simplicity using data from QATS. Using QE research for reference-less evaluation in simplification is still an area not sufficiently explored, mainly because it requires human annotations on example instances that can be used as training data, which can be expensive to collect.

Another group of approaches investigates how to **rank** sentences according to their predicted reading levels. Vajjala and Meurers (2014a,b) showed that, in the PWKP (Zhu et al., 2010) dataset and and earlier version of the OneStopEnglish (Vajjala and Lučić, 2018) corpus, even if all simplified sentences were simpler than their aligned original counterpart, some sentences in the "simple" section had a higher reading level than some in the "original" section. As such, attempting to use binary classification approaches to determine if a sentence is simple or not may not be the appropriate way to model the task. Consequently, Vajjala and Meurers (2015) proposed to use pair-wise ranking to assess the readability of simplified sentences. They used the same features of the document-level model of Vajjala and Meurers (2014a), but now they attempt to learn to predict which of two given sentences is simpler than the other. Ambati et al. (2016) tested the usefulness of syntactic features extracted from an incremental parser for the task, while Howcroft and Demberg (2017) explored using more psycholinguistic features, such as idea density, surprisal, integration cost, and embedding depth.

[8] In Quality Estimation, the goal is to evaluate an output translation without comparing it to a reference. For a comprehensive review of this area of research, please refer to (Specia et al., 2018).

2.3.3 Discussion

In this Section we have described how the outputs of Sentence Simplification models are evaluated using both human judgements and automatic metrics. We have attempted not to only explain these methods, but to also point out their advantages and disadvantages.

In the case of human evaluation, one important but often overlooked aspect is that it should be carried out by individuals from the same target audience of the data on which the Sentence Simplification model was trained. This is specially relevant when collecting simplicity judgements due to its subjective nature: what a non-native proficient adult speaker considers "simple" may not hold for a native-speaking primary school student, for example. Even within the same target group, differences in simplicity needs and judgements may arise. This is why some researchers have started to focus on developing and evaluating models for personalised simplification (Bingel et al., 2018a,b). In addition, we should think carefully whether the quality of a simplified text is better judged as an intrinsic feature, or if we should assess it depending on its usefulness to carry out another task. Nowadays, quality judgements focus on assessing the automatic output for what it is: *is it grammatical?*, *does it still express the same idea?*, *is it easier to read?* However, the goal of simplification is to modify a text so that a reader can understand it better. With that in mind, a more functional evaluation of the generated text could be more informative of the understandability and usefulness of the output. An example of such type of task-based assessment is presented by Mandya et al. (2014), where human judges had to use the automatically-simplified texts in a reading comprehension test with multiple-choice questions. Afterwards, the accuracy of their responses is used to qualify the helpfulness of the simplified texts in the particular comprehension task. This type of human evaluation could be more goal-oriented, but they are costly to create and execute.

Automatic metrics are useful for quickly assessing models and comparing different architectures. They could even be considered more objective than humans since personal biases do not play a role. However, the metrics used in Sentence Simplification research are flawed. BLEU has been found to only be reliable for assessment in Machine Translation but not other Natural Language Generation tasks (Reiter, 2018), and it is not adequate for most simplification operations (Sulem et al., 2018a). SARI is only useful as a proxy for simplicity gain assessment, limited to lexical simplifications and short-distance reordering despite more simplification operations being possible. Commonly-used Flesch metrics were developed to assess complete documents and not sentences, which is the focus of most simplification research nowadays. Therefore, when evaluating models using automatic scores, it is essential to consider their particular limitations, to always look at all possible metrics and try to interpret them accordingly.

2.4 Data-driven Approaches to Sentence Simplification

In this Section, we review research aiming at learning sentence simplifications from examples. More specifically, approaches that involve learning simplification operations from parallel corpora of aligned original-simplified sentences in English. Compared to approaches based on hand-crafted rules, data-driven approaches can perform multiple simplification operations simultaneously, as well as learn very specific and complex rewriting patterns. As a result, they make it possible to model interdependencies among different operations more naturally. Therefore, we do not include approaches to Sentence Simplification based on sets of hand-crafted rules, such as rules for splitting and reordering sentences (Bott et al., 2012; Candido Jr. et al., 2009; Siddharthan, 2011), nor approaches that only learn lexical simplifications, that is, which target one-word replacements.

We classify data-driven approaches as relying on statistical machine translation techniques (Sec. 2.4.1), induction of synchronous grammars (Sec. 2.4.2), semantics-assisted (Sec. 2.4.3), and neural sequence-to-sequence models (Sec. 2.4.4). For each group, we describe some of the most representative approaches by detailing the models implemented and the resources they used for training and evaluating their systems. At the end of each subsection, we also compare the models in the corresponding group in terms of automatic metrics using their authors' self-reported results. Whilst direct comparisons are not always possible due to authors using different test datasets or automatic metrics, it serves to get a general idea of the quality of the simplifications of each groups' models. A fairer comparison using the same test datasets and metrics is later presented in Sec. 3.3.2.

2.4.1 Monolingual Statistical Machine Translation

Several approaches treat Sentence Simplification as a monolingual Machine Translation task, with *original* and *simplified* as *source* and *target* languages, respectively. While other translation methods exist, in this Section we focus on Statistical Machine Translation (SMT). Given a sentence f in the source language, the goal of an SMT model is to produce a translation e in the target language. This is modelled using the noisy channel framework (Equation 2.9).

$$e^* = \underset{e \in E}{\arg\max}\, p(e|f) = \underset{e \in E}{\arg\max}\, p(f|e)p(e) \tag{2.9}$$

This framework relies on a **translation model** $p(f|e)$ and a **language model** $p(e)$. In addition, a **decoder** is in charge of producing the most probable e given an f. The language model is monolingual, and thus "easier" to generate. There are different approaches for implementing the translation model and the decoder. In practice, they all rely on a linear combination of these and additional features, which are directly optimised to maximise translation quality, rather than on the generative noisy channel model. In what follows, we review the most popular approaches and explain their applications for Sentence Simplification.

Phrase-Based MT Approaches

The intuition behind Phrase-Based SMT (PBSMT) is to use phrases (sequences of words) as the fundamental unit of translation. Therefore, the translation model $p(f|e)$ depends on the normalised count of the number of times each possible phrase-pair occurs. These counts are extracted from parallel corpora and automatic phrase alignments that are obtained from word alignments. Decoding is a search problem: find the sentence that maximises the translation and language model probabilities. It could be solved using a best-first search algorithm, like A*, but exploring the entire search space of possible translations is expensive. Therefore, decoders use beam-search to only retain, at every step, the most promising states to continue the search.

Moses (Koehn et al., 2007) is a popular PBSMT system, freely available.[9] It provides tools for easy training, tuning and testing of translation models based on this SMT approach. Specia (2010) was the first to use this toolkit, with no adaptations, for the simplification task. Experiments were carried out on a parallel corpus of original and manually-simplified newspaper articles in Brazilian Portuguese (Caseli et al., 2009). The trained model mostly executes lexical simplifications and simple rewritings. However, as expected, it is overcautious and cannot perform long distance operations like subject-verb-object reordering or splitting.

Moses-Del: Coster and Kauchak (2011b) also used Moses as-is, and trained it on their C&K corpus, obtaining slightly better results when compared to not doing any simplification. In Coster and Kauchak (2011a), the authors modified Moses to allow complex phrases to be aligned to NULL, thus implementing deletions during simplification. To accomplish this, they modify the word alignments before the phrase alignments are learned: (1) any complex unaligned word is now aligned to NULL, and (2) if several complex words in a set align to only one simple word, and one of the complex words is equal to the simple word, then the other complex words in the set are aligned to NULL. Their model achieves better results than a standard Moses implementation.

PBSMT-R: Wubben et al. (2012) also used Moses but added a post-processing step. They ask the decoder to generate the 10 best simplifications (where possible), and then rank them according to their dissimilarity to the input sentence (measured by edit distance). The most dissimilar sentence is chosen as the final output, and ties are resolved using the decoder

[9] http://www.statmt.org/moses/

score. This trained model achieves a better BLEU score than more sophisticated approaches (Woodsend and Lapata, 2011a; Zhu et al., 2010), explained in Sec. 2.4.2. When compared to such approaches using human evaluation, PBSMT-R is better in grammaticality and meaning preservation. However, the results are limited to paraphrasing transformations.

Table 2.2 summarises the performance of PBSMT-based Sentence Simplification models, as reported in the original papers. The BLEU values are not directly comparable, since each approach used a different corpus for testing. From a operation capability point of view, PBSMT-based simplification models are able to perform substitutions, short distance reorderings and deletions, but fail to learn more sophisticated operations (e.g. splitting) that may require more information on the structure of the sentences and relationships between their components.

Table 2.2 Self-reported performance of PBSMT-based Sentence Simplification models.

Model	Train Corpus	Test Corpus	BLEU ↑	FKGL ↓
Moses (Brazilian Portuguese)	PorSimples	PorSimples	60.75	
Moses (English)	C&K	C&K	59.87	
Moses-Del	C&K	C&K	60.46	
PBSMT-R	PWKP	PWKP	43.00	13.38

Syntax-Based MT Approaches

In Syntax-Based SMT (SBSMT), the basic units for translation are syntactic components in parse trees. In PBSMT, the language model and phrase alignments act as features that inform the model about how likely the generated translation is (simplification, in our case). In SBSMT we can extract more informed features, based on the structures of the parallel parse trees.

TSM: Zhu et al. (2010) proposed a **T**ree-based **S**implification **M**odel that can perform four simplification operations: splitting, dropping, reordering, and substitution (of words and phrases). Given an original sentence c, the model attempts to find a simplification s using Equation 2.10, with a language model $P(s)$ and a direct translation model $P(s|c)$.

$$s = \arg\max_{s} P(s|c)P(s) \tag{2.10}$$

For estimating $P(s|c)$, the method traverses the original sentence parse tree from top to bottom, extracting features from each node and for each of the four possible transformations. These features are operation-specific and are stored in feature tables for each operation. For each feature combination in each table, probabilities are calculated during training. We will use the splitting operation to explain this process in more detail. A similar method is used for the other three operations.

In TSM, sentence splitting is decomposed into two operations: SEGMENTATION (if a sentence is to be split or not) and COMPLETION (to make the splits grammatical). The probability of a SEGMENTATION operations is calculated using Equation 2.11, where w is a word in the complex sentence c, and $SFT(w|c)$ is the probability of w in the Segmentation Feature Table (SFT).

$$P(seg|c) = \prod_{w:c} SFT(w|c) \tag{2.11}$$

COMPLETION implies deciding if the border word in the second split needs to the dropped, and which parts of the first split need to be copied into the second. The probability of this operation is calculated as in Equation 2.12, where s are the split sentences, bw is a border word in s, w is a word in s, dep is a dependency of w which is out of the scope of s, $BDFT$ is the Border Drop Feature Table (BDFT), and CFT is the Copy Feature Table.

$$P(com|seg) = \prod_{bw:s} BDFT(bw|s) \prod_{w:s} \prod_{dep:w} CFT(dep) \tag{2.12}$$

Finally, once similar computations are done for the other three operations, all probabilities are combined to calculate the translation model. In Equation 2.13 where $P(dp|node)$, $P(ro|node)$ and $P(sub|node)$ correspond to dropping, reordering, and substituting non-terminal nodes, and $P(sub|w)$ is for substitutions of terminal nodes.

$$\begin{aligned} P(s|c) = \sum_{\theta:Str(\theta(c))=s} \Big(& P(seg|c)P(com|seg) \\ & \prod_{node} P(dp|node)P(ro|node)P(sub|node) \\ & \prod_{w} (sub|w) \Big) \end{aligned} \tag{2.13}$$

The model is trained using the Expectation-Maximisation algorithm proposed by Yamada and Knight (2001). This algorithm builds a training tree to calculate $P(s|c)$, which corresponds to the probability of the root of the training tree. For decoding, the inside and outside probabilities are calculated for each node in the decoding tree, which is constructed in a similar fashion as the training tree. To simplify a new original sentence, the algorithm starts from the root and greedily selects the branch with highest outside probability.

The proposed approach used the PWKP corpus for training and testing, obtaining higher readability scores (Flesch) than other baselines considered (Moses and the sentence compression system of Filippova and Strube (2008) with variations). Overall, TSM showed good performance for word substitution and splitting.

TriS: Bach et al. (2011) proposed a method for splitting an original sentence into several simpler ones. A sentence is considered simple if it is in the subject-verb-object order (SVO), with one subject, one verb and one object. Given a sentence c that needs to be split into a set S of simple sentences, the objective is to select the set with the highest probability (Equation 2.14).

$$\hat{S}(c) = \underset{\forall S}{\arg\max}\, P(S|c) \tag{2.14}$$

The language and translation models are combined using a log-linear model as in Equation 2.15, where $f_m(S,c)$ are feature functions on each sentence, and w_m are model parameters to be learned.

$$p(S|c) = \frac{\exp\left(\sum_{m=1}^{M} w_m f_m(S,c)\right)}{\sum_{S'} \exp\left(\sum_{m=1}^{M} w_m f_m(S',c)\right)} \tag{2.15}$$

For decoding, their method starts by listing all noun phrases and verbs of the original sentence and generating simple sentences combining the lists in SVO form. Then, it proceeds with a k-best stack decoding algorithm: it starts with a stack of 1-simple-sentence hypotheses (a hypothesis is a complete simplification of multiple simple sentences), and at each step it pops a hypothesis of the stack and expands it (to 2-simple-sentence in the first iteration and so on) and puts the new hypotheses in another stack, prunes them (according to some metric) and updates the original hypotheses stack. After all steps (which corresponds to the number of verbs in the sentence), it selects the k-best hypotheses in the stack. For training, they use the Margin Infused Relaxed Algorithm (Crammer and Singer, 2003). For modelling, 177 feature functions were designed to capture intra and inter sentential information (simple counts, distance in the parse tree, readability measures, etc.).

To test their approach, a corpus of 854 sentences extracted from The New York Times and Wikipedia was created, with one manual simplification each. The authors evaluate on 100 unseen sentences and compare against the rule-based approach of Heilman and Smith (2010). Their model achieves better Flesch-Kincaid Grade Level and ROUGE scores.

SBSMT (PPDB + <Metric>): Xu et al. (2016) proposed to optimise a SBSMT framework with rule-based features and tuning metrics specific for lexical simplification (FKBLEU and SARI, described in Sec. 2.3.2). The proposed simplification model also relies on paraphrasing rules available in the Paraphrase Database (PPDB), which are expressed as a Synchronous

Context-Free Grammar (SCFG). The authors also added nine new features to each rule in the PPDB (each rule already contains 33). These new features are simplification-specific, for example: length in characters, length in words, number of syllables, among others.

These modifications were implemented in the SBSMT toolkit Joshua (Post et al., 2013) and performed experiments using TurkCorpus (described in Sec. 2.2.3) on three versions of the SBSMT system, changing the tuning metric (BLEU, FKBLEU and SARI). Evaluations using human judgements show that all three models achieved better grammatically, meaning preservation and simplicity gain than PBSMT-R (Wubben et al., 2012).

Table 2.3 summarises the performance of the syntax-based models trained using the Sentence Simplification approaches described. These values are not directly comparable, since each approach used a different corpus for testing. In the case of the models based on Joshua and PPDB, not surprisingly, each achieves the highest score according to the metric it was optimised for. However, SBSMT (PPDB + SARI) seems to be the overall best. From an operations capability point of view, TSM and TriS are capable of performing splitting, which is an advantage over SBSMT variations that only generate paraphrases.

Table 2.3 Self-reported performance of SBSMT-based Sentence Simplification models.

Model	Train Corpus	Test Corpus	BLEU ↑	FKGL ↓	SARI ↑
TSM	PWKP	PWKP	38.00		
TriS	Own	Own		7.9	
SBSMT(PPDB+BLEU)	TurkCorpus	TurkCorpus	99.05	12.88	26.05
SBSMT(PPDB+FKBLEU)	TurkCorpus	TurkCorpus	74.48	10.75	34.18
SBSMT(PPDB+SARI)	TurkCorpus	TurkCorpus	72.36	10.90	37.91

2.4.2 Grammar Induction

In this approach, Sentence Simplification is modelled as a tree-to-tree rewriting problem. Approaches typically follow a two step process: (1) use parallel corpora of aligned original-simplified sentence pairs to extract a set of tree transformation rules, and then (2) learn how to select which rule(s) to apply to unseen sentences to generate the best simplified output. This is analogous to how a SBSMT approach works: the rules would be the features, and the decoder applies the learnt model deciding how to use these rules. In what follows, we first provide some brief preliminary explanations on synchronous grammars, and then proceed to explain how grammar-induction-based approaches have implemented each of the aforementioned steps.

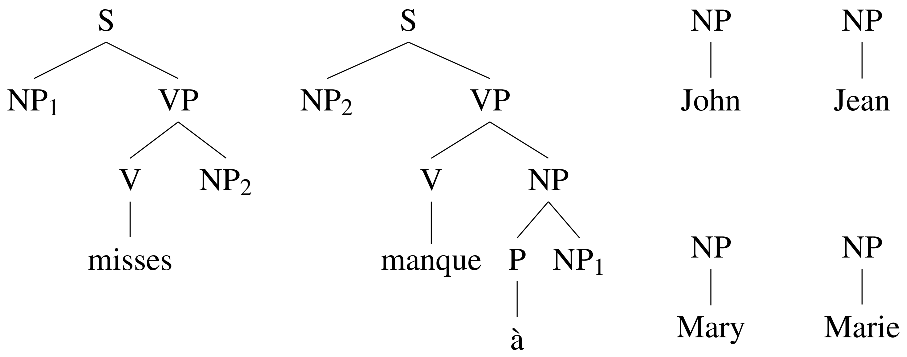
Preliminaries

A Context-Free Grammar (CFG) is a set of productions or rewriting rules that describe how to generate strings in a formal language. Synchronous Context-Free Grammars (SCFGs) are a generalisation of CFGs that generate pairs of related strings and not just single strings (Chiang, 2006). In a SCFG, each production has source and target sides that are related. For example, we show a SCFG for a sentence in English and its translation to Japanese (Chiang, 2006):

$$\begin{aligned}
\mathrm{S} &\rightarrow \langle \mathrm{NP}_1\mathrm{VP}_2 | \mathrm{NP}_1\mathrm{VP}_2 \rangle \\
\mathrm{VP} &\rightarrow \langle \mathrm{V}_1\mathrm{NP}_2 | \mathrm{NP}_2\mathrm{V}_1 \rangle \\
\mathrm{NP} &\rightarrow \langle i | watashi\ wa \rangle \\
\mathrm{NP} &\rightarrow \langle the\ box | hako\ wo \rangle \\
\mathrm{V} &\rightarrow \langle open | akemasu \rangle
\end{aligned}$$

The numbers in the non-terminals serve as links between nodes in the source and target. These links are 1-to-1 and every non-terminal is always linked to another.

SCFGs have the limitation of only being able to relabel and reorder sibling nodes. In contrast, **Synchronous Tree Substitution Grammars** (STSGs, Eisner, 2003) are able to perform more long-distance swapping. In a STSG, productions are pairs of elementary trees, which are tree fragments whose leaves can be non-terminal or terminal symbols:



SCFGs impose an isomorphism constraint between the aligned trees that STSGs relax. However, to account for all the different movement patterns that could exist in a language would require powerful and, perhaps, slow grammars (Smith and Eisner, 2006). **Quasi-synchronous Grammars** (QS, Smith and Eisner, 2006) relax the isomorphism constraint further, following the intuition that one of the parallel trees is *inspired* by the other. This means that any node in one tree can be linked to any other node on the other tree. Observe that in STSGs, even though the linked nodes can be in any part of the frontier of the trees, they still need to have the same syntactic tag. This is not the case in QGs since *anything can align to anything*.

Simplification Models

The formalisms explained in previously have been used to automatically-extract rules that convey the rewriting operations required to simplify sentences. Using these transformation rules, grammar-induction-based models then decide, given an original sentence, which rule(s) to apply and how to generate the final simplification output (often referred to as *decoding*).

QG+ILP: Woodsend and Lapata (2011a) use QGs to induce rewrite rules that can perform sentence splitting, word substitution and deletion. From word alignments between source and target sentences, they first align non-terminal nodes where more than one child node aligns. From these constituent alignments, they extract syntactic simplification rules. However, if a pair of trees have the same syntactic structure but differ only because of a lexical substitution, a more general rule is extracted considering only the words and their part-of-speech tags. To create the rules for sentence splitting, a source sentence is aligned with two consecutive target sentences (named *main* and *auxiliary*). They then select a split node, which is a source node that "contributes" (i.e. has aligned children nodes) to both *main* and *auxiliary* targets. This results in a rule with three components: the source node, the node in the target *main* sentence, and the phrase structure in the entire *auxiliary* sentence. Some examples of these rules are:

- Lexical: $\langle$[VBN discovered]$\rangle \rightarrow \langle$ [VBD was] [VBN found]$\rangle$
- Syntactic: $\langle$NP, ST$\rangle \rightarrow \langle$[NP PP$_1$], [ST$_1$]$\rangle$
- Split: $\langle$NP, NP, ST$\rangle \rightarrow \langle$[NP$_1$ SBAR$_2$], [NP$_1$], [ST$_2$]$\rangle$

With these transformation rules, Woodsend and Lapata (2011a) then use Integer Linear Programming (ILP) to find the best candidate simplification. During decoding, the original sentence's parse tree is traversed from top to bottom, applying at each node all the simplification rules that match. If more than one rule matches, each candidate simplification is added to the target tree. As a result, a "super tree" is created, which contains all possible simplifications of each node of the source sentence. Then, an ILP program decides which nodes should be kept and which would be removed. The objective function of the ILP considers a penalty for substitutions and rewrites (favours the more common transformations with less penalty), and tries to reduce the number of words and syllables. The ILP has constraints to ensure grammaticality (if a phrase node is chosen, the node it depends on is also chosen), coherence (if one partition of a split sentence is chosen, the other partition is also chosen), and always one (and only one) simplification per sentence. The authors trained two models: one extracting rules from the AlignedLP corpus (AlignILP) and other using the RevisionWL corpus (RevILP).

For evaluation, they used the test split from the PWKP instances. RevILP was their best model, achieving the closest scores to the references using both Flesh-Kincaid and human judgements on simplicity, grammaticality and meaning preservation.

T3+Rank: Paetzold and Specia (2013) extract candidate tree rewriting rules using T3 (Cohn and Lapata, 2009), an abstractive sentence compression model that employs STSGs for deletion, reordering and substitution. Using word-aligned parallel sentences, the model maps the word alignment into a constituent level alignment between the source and target trees by adapting the alignment template method of Och and Ney (2004). This constituent alignments are then generalised (i.e. aligned nodes are replaced with links) to extract rules. This generalisation is performed by a recursive algorithm that attempts to find the minimal most general set of synchronous rules. The recursive depth allowed by the algorithm determines the specificity of the rules. Once the rules have been extracted, Paetzold and Specia (2013) divide them into two sets: one with purely syntactic transformations, and other with purely lexical transformations. This process has two goals: (1) to filter out rules that are not simplification transformations according to some pre-established criteria on what type of information the rule contains, and (2) to be able to explore syntactic and lexical simplification individually. Given an original sentence, the proposed approach applies both sets of rewriting rules separately. The candidate simplifications are then ranked, in each set, in order to determine the best syntactic and the best lexical simplifications. For ranking, they measure the perplexity of the output using a language model built from SEW. Evaluation results using human judgements on simplicity, grammaticality and meaning preservation, were only encouraging for lexical simplification, were almost half of the automatically simplified sentences were considered simpler, over 80% grammatical, and a little over 50% as meaning preserving.

SimpleTT: Feblowitz and Kauchak (2013) proposed an approach similar to T3 (Cohn and Lapata, 2009) using STSGs. They modified the rule extraction process to reduce the number of candidate rules that need to be generalised. Also, instead of controlling the rules' specificity with the recursion depth, SimpleTT augments the rules with more information, such as head's lexicalisation and part-of-speech tag. During decoding, the model starts by trying to match the more specific rules up to the most general. If no rule matches, then the source parse tree is just copied. The model generates the 10,000 most probable simplifications for a given sentence according to the STSG grammar, and the best one is determined using a log-linear combination of features, such as rule probability and output length. For training and testing, the authors used the C&K-1 corpus, and compared their model against PBSMT-R and Moses-Del. Using human evaluation, SimpleTT gets the highest scores for simplicity and grammaticality among the tested models, with values comparable to those for human simplifications. However, it gets the lowest score in meaning preservation, presumably because SimpleTT tends to simplify by deleting sentence's elements. This approach does not explicitly model sentence splitting.

Table 2.4 summarises the performance of the models trained with the Sentence Simplification approaches described. Results are not directly comparable. Overall, grammar-induction-based approaches, because of their pipeline architecture, offer more flexibility on how the rules are learned and how they are applied, as compared to end-to-end approaches. Even though Woodsend and Lapata (2011a) were the only ones that attempted to model splitting, the other approaches could be modified in a similar way, since the formalisms allow it.

Table 2.4 Self-reported performance of grammar-based Sentence Simplification models. A (*) indicates a test set different from the standard.

Model	Train Corpus	Test Corpus	BLEU ↑	FKGL ↓
QG+ILP	AlignedWL	PWKP	34.0	12.36
	RevisionWL	PWKP	42.0	10.92
T3+Rank	C&K	C&K*	34.2	
SimpleTT	C&K	C&K	56.4	

2.4.3 Semantics-Assisted

Narayan and Gardent (2014) argue that the simplification transformation of splitting is semantics-driven. In many cases, splitting occurs when an entity takes part in two (or more) distinct events described in a single sentence. For example, in Sentence 3 below, *bricks* is involved in two events: "being resistant to cold" and "enabling the construction of permanent buildings".

(3) **Original:** Being more resistant to cold, bricks enabled the construction of permanent buildings.

(4) **Simplified:** Bricks were more resistant to cold. Bricks enabled the construction of permanent buildings.

Even though deciding when and where to split can be determined by syntax (e.g. sentences with relative or subordinate clauses), constructing the second sentence in the split by adding the shared element with the first split should be accomplished by semantic information, since we need to identify the entity involved in both events.

Hybrid: Narayan and Gardent (2014) combine semantics-driven splitting and deletion, with PBSMT-based substitution and reordering. The general idea is to use semantic roles to identify the events in a given original sentence, and those events would determine how to split the sentence. Deletion would also be directed by this information, since mandatory arguments for each identified verbal predicate should not be deleted. For substitution of complex words/phrases and reordering, the authors rely on a PBSMT-based model.

The proposed method first uses Boxer (Curran et al., 2007) to obtain the semantic representation of the original sentences. From there, candidate splitting pairs are selected from events that share a common core semantic role (e.g. agent and patient). The probability of a candidate being a split is determined by the semantic roles associated with it. The probability of deleting a node is determined by its semantic relations to the split events. And, the probabilities for substitution and reordering are determined by a PBSMT system. For training, they used the Expectation-Maximisation algorithm of Yamada and Knight (2001), same as Zhu et al. (2010), but calculating probabilities over Boxer's semantic graph instead of a syntactic parse tree.

The Sentence Simplification model is trained and tested using the PWKP corpus. Sentences for which Boxer failed to extract a semantic representation were excluded during training, and directly passed to the PBSMT system in testing. For evaluation, the model is compared against QG+ILP, PBSMT-R, and TSM. Hybrid performs splits closer in proportion to those of the references. It also achieves the highest BLEU score and smaller edit distance to references. With human evaluation, Hybrid obtains the highest score in simplicity, and is a close second to PBSMT-R for grammaticality and meaning preservation.

UNSUP: Narayan and Gardent (2016) propose a method that does not require aligned original-simplified sentences to train a Sentence Simplification model. Their approach first uses a context-aware lexical simplifier (Biran et al., 2011) that learns simplification rules from articles of EW and SEW. Given an original sentence, these rules are applied and the best combination of simplifications is found using dynamic programming. Then, they use Boxer to extract the semantic representation of the sentence and identify the events/predicates. After that, they estimate the maximum likelihood of the sequences of semantic role sets that would result after each possible split (i.e. subsequence of events). To compute these probabilities, they only rely on data from SEW. Finally, they use an ILP to determine which phrases in the sentence should be deleted, similarly to the compression model of Filippova and Strube (2008).

For evaluation, they use the test set of PWKP, and compare against TSM, QG+ILP, PBSMT-R, and Hybrid. The proposed unsupervised pipeline achieves results comparable to those of the other models, but it is only better than TSM in terms of BLEU. It produces more splits than Hybrid and than the reference, but there is no analysis about the correctness of these splits. Using human evaluation, UNSUP achieves the highest values in simplicity and grammaticality.

EvLex: Štajner and Glavaš (2017) introduce a Sentence Simplification model that can perform sentence splitting, content reduction and lexical simplifications. For the first two operations, they build on Glavaš and Štajner (2013) identifying events in a given sentence. Each identified event and its arguments constitute a new simpler sentence (splitting). Information that does not correspond to any of the events is discarded (content reduction). For event identification, they use EvGraph (Glavaš and Šnajder, 2015), which relies on a supervised classifier to identify events and a set of manually-constructed rules to identify the arguments. Finally, the pipeline also incorporates an unsupervised lexical simplification model (Glavaš and Štajner, 2015). The authors carried out tests to determine whether the order of the components affects the resulting simplifications. They found that the differences were minimal.

The proposed architecture is compared to QG+ILP (Woodsend and Lapata, 2011a), testing in two datasets, one with news stories (News) and one with Wikipedia sentences (Wiki). EvLex achieves the highest FRE score in the News dataset, while QG+ILP is the best in this metric in Wiki. Regarding human evaluations, their model achieves the best grammaticality and simplicity scores for both datasets.

Table 2.5 summarises the results of the models trained with the Sentence Simplification approaches presented in this Section. Not all models are directly comparable because they were tested in different corpora. The semantics-aided models presented resemble, in part, the research of Bach et al. (2011) explained in Sec. 2.4.1, focused on sentence splitting. In that work, splitting is based on preserving a SVO order in each split, which could be considered as an agent-verb-patient structure. These findings suggest that the splitting operation requires a more tailored modelling, different from standard MT-based sequence-to-sequence approaches.

Table 2.5 Self-reported performance of semantics-assisted Sentence Simplification models.

Model	Train Corpus	Test Corpus	BLEU ↑	FRE ↑
EvLex		Wiki		59.8
		News		74.7
Hybrid	PWKP	PWKP	53.60	
UNSUP	PWKP	PWKP	38.47	

Split-and-Rephrase

Narayan et al. (2017) introduce a new task called split-and-rephrase, focused on splitting a sentence into several others, and making the necessary changes to ensure grammaticality. No deletions should be performed so as to preserve meaning. The authors use the WebSplit dataset (described in Sec. 2.2.3) to train and test five models for the split-and-rephrase task: (1) Hybrid (Narayan and Gardent, 2014); (2) Seq2Seq, which is an encoder-decoder with local-p attention (Luong et al., 2015); (3) MultiSeq2Seq, which is a multi-source sequence-to-sequence model (Zoph and Knight, 2016) that takes as input the original sentence and its MR triples; and (4) one that models the problem in two steps: first learn to split, and then learn to rephrase. In this last model, the splitting step uses the original sentence and its MR to split the latter into several MR sets. However, two variations are explored for the rephrasing step: (1) Split-MultiSeq2Seq learns to rephrase from the split MRs and the original sentence in a multi-source fashion, while (2) Split-Seq2Seq only uses the split MRs and rephrases based on a sequence-to-sequence model.

All models were automatically evaluated using multi-reference BLEU, the average number of simple sentences per complex sentence, and the average number of output words per output simple sentence. Split-Seq2Seq achieved the best scores in all the metrics. This result supports the idea that the split-and-rephrase task is better treated in a pipeline. One could also hypothesize that the semantic information in the MRs helps in the splitting step. However, according to the paper, the authors *"strip off named-entities and properties from each triple and only keep the tree skeleton"* when learning to split. This suggests that it may not be the semantic information itself (i.e. the properties), but the groupings of semantically-related elements in each triple that helps perform the splits.

Aharoni and Goldberg (2018) focus on the text-to-text setup of the task, that is, without using the MR information. They first propose a new split of the dataset after determining that around 90% of unique simple sentences in the original development and test sets also appeared in the training set. With the new data splits, they train a vanilla sequence-to-sequence model with attention, and incorporate a copy mechanism inspired by work in abstractive text summarisation (Gu et al., 2016; See et al., 2017). This model achieves the best performance in both the original and new dataset splits, but with a low BLEU score of 24.97.

Botha et al. (2018) argue that poor performance on the task may be due to WEBSPLIT not being suitable for training models. According to the authors, since WEBSPLIT was derived from the WEBNLG dataset, it only contains artificial sentence splits created from the RDF triples. As such, they introduce WikiSplit, a new dataset for split-and-rephrase based on EW edit histories (see Sec. 2.2). Botha et al. use the same model as Aharoni and Goldberg (2018) to experiment with different combinations of training data: WEBSPLIT only, WikiSplit only, and Both. The evaluation is performed on WEBSPLIT. Results show that training using WikiSplit only or Both improves performance on the task in around 30 BLEU points.

2.4.4 Neural Sequence-to-Sequence

Sentence Simplification is modelled as a sequence-to-sequence problem, and tackled normally with an attention-based encoder-decoder architecture (Bahdanau et al., 2014). The encoder projects the source sentence into a set of continuous vector representations from which the decoder generates the target sentence. A major advantage of this approach is that it allows to train end-to-end models without needing to extract features or estimate individual model components, such as the language model. In addition, all simplification operations can be learnt simultaneously, instead of developing individual mechanisms as in previous research.

RNN-based Architectures

Most models are based on Recurrent Neural Networks (RNNs) with Long Short Term Memory units (LSTMs, Hochreiter and Schmidhuber, 1997). Given an original source sentence $X = (x_1, x_2, ..., x_{|X|})$, the model learns to predict its simplified version, $Y = (y_1, y_2, ..., y_{|Y|})$. It uses an encoder that transforms the source sentence X into a sequence of hidden states (h_1^S, h_2^S, ..., $h_{|X|}^S$), from which the decoder generates one word y_{t+1} at the time in target Y. The generation process is conditioned on all the words generated so far $y_{1:t}$ and a dynamic context vector c_t, which also encodes the source sentence:

$$P(Y|X) = \prod_{t=1}^{|Y|} P(y_t|y_{1:t-1}, X) \tag{2.16}$$

$$P(y_{t+1}|y_{1:t}, X) = \text{softmax}(g(h_t^T, c_t)) \tag{2.17}$$

where $g(\cdot)$ is a neural network with one hidden layer and parametrised as follows:

$$g(h_t^T, c_t) = W_o \tanh(U_h h_t^T + W_h c_t) \tag{2.18}$$

where $W_o \in \mathbb{R}^{|V| \times d}$, $U_h \in \mathbb{R}^{d \times d}$, and $W_h \in \mathbb{R}^{d \times d}$; $|V|$ is the output vocabulary size and d is the hidden unit size. h_t^T is the hidden state of the decoder LSTM which summarises $y_{1:t}$ (what has been generated so far):

$$h_t^T = \text{LSTM}(y_t, h_{t-1}^T) \tag{2.19}$$

The dynamic context vector c_t is a weighted sum of the hidden states of the source sentence, whose weights α_{t_i} are determined by an attention mechanism:

$$c_t = \sum_{i=1}^{|X|} \alpha_{t_i} h_i^S \qquad \alpha_{t_i} = \frac{\exp(h_t^T \cdot h_i^S)}{\sum_i \exp(h_t^T \cdot h_i^S)} \tag{2.20}$$

NTS: Nisioi et al. (2017) introduced the first **N**eural **T**ext **S**implification approach using the encoder-decoder with attention architecture provided by OpenNMT (Klein et al., 2017). They experimented with using the default system, and also with combining pre-trained word2vec word embeddings (Mikolov et al., 2013) with locally trained ones. They also generated two candidate hypotheses for each beam size, and used BLEU and SARI to determine which hypothesis to choose from the n-best list of candidates. EW-SEW was used for training, and TurkCorpus for validation and testing. When compared to PBSMT-R and SBSMT (PPDB+SARI), NTS with its default features achieved the highest grammaticality and meaning preservation scores in human evaluation. SBSMT (PPDB+SARI) was still the best using SARI scores. Overall, NTS is able to perform simplifications limited to lexical paraphrasing and deletion operations. It is also appears that choosing the second hypothesis results in less conservative simplifications.

NSELSTM: Vu et al. (2018) used Neural Semantic Encoders (NSEs, Munkhdalai and Yu, 2017) instead of LSTMs for the encoder. At any encoding time step, a NSE has access to all the tokens in the input sequence, and is thus able to capture more context information while encoding the current token, instead of only relying on the previous hidden state. Their approach is tested on PWKP, TurkCorpus and Newsela. Two models are presented, one tuned using BLEU (NSELSTM-B) and one using SARI (NSELSTM-S). When compared against other models, NSELSTM-B achieved the best BLEU scores in the Newsela and TurkCorpus datasets, while NSELSTM-S was second-best on SARI scores in Newsela and PWKP. According to human evaluation, NSELSTM-B has the best grammaticality for Newsela and PWKP, while NSELSTM-S is the best in meaning preservation and simplicity for PWKP and TurkCorpus.

Modifying the Training Method

Without significantly changing the standard RNN-based architecture described before, some research has experimented with alternative learning algorithms to train the models.

DRESS: Zhang and Lapata (2017) use the standard attention-based encoder-decoder as an agent within a Reinforcement Learning (RL) architecture (Figure 2.2). An advantage of this approach is that the model can be trained end-to-end using specific evaluation metrics for Sentence Simplification.

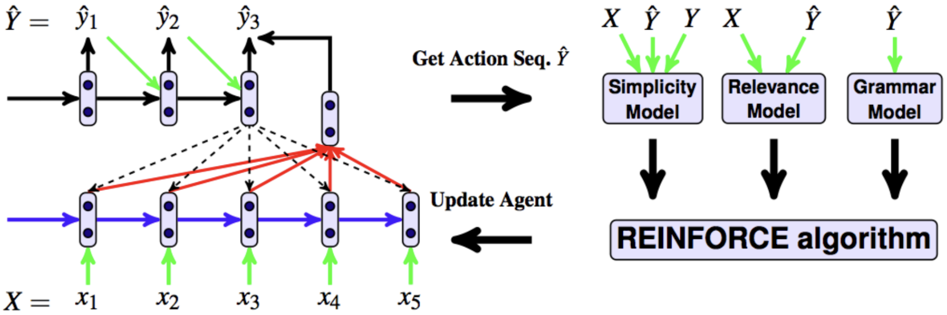

$\hat{Y} = \hat{y}_1 \quad \hat{y}_2 \quad \hat{y}_3$
Get Action Seq. $\hat{Y}$
Update Agent
$X = x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5$
$X \ \hat{Y} \ Y$
Simplicity Model
$X \ \hat{Y}$
Relevance Model
$\hat{Y}$
Grammar Model
REINFORCE algorithm

Figure 2.2 Model architecture for DRESS. Extracted from Zhang and Lapata (2017).

The agent reads the original sentence and takes a series of actions (words in the vocabulary) to generate the simplified output. After that, it receives a reward that scores the output according to its simplicity, relevance (meaning preservation) and fluency (grammaticality). To reward simplicity, they calculate SARI in both the expected direction and in reverse (using the output as reference, and the reference as output) to counteract the effect of having noisy data and a single reference; the reward is then the weighted sum of both values. To reward relevance, they compute the cosine similarity between the vector representations (obtained using a LSTM) of the source sentence and the predicted output. To reward fluency, they calculate the probability of the predicted output using an LSTM language model trained on simple sentences.

For learning, the authors used the REINFORCE algorithm (Williams, 1992), whose goal is to find an agent that maximises the expected reward. As such, the training loss is given by the negative expected reward:

$$\mathscr{L}(\theta) = -\mathbb{E}_{(\hat{y}_1,\ldots,\hat{y}_{|\hat{Y}|})\sim P_{RL}(\cdot|X)}[r(\hat{y}_1,\ldots,\hat{y}_{|\hat{Y}|})] \tag{2.21}$$

where P_{RL} is the policy, given in our case by the distribution produced by the encoder-decoder (Equation 2.17) and $r(\cdot)$ is the reward function. The authors followed Ranzato et al. (2016) in first pre-training the agent by minimising the negative log-likelihood of the training source-target pairs, in order to avoid starting the process with a random policy. Then, for each target sequence, they used the same learning process to train the first L tokens, and applied the RL algorithm in the remaining token.

Even though the model as-is can learn lexical simplifications, the authors state that these are not always correct. As such, the model is modified to learn them explicitly. An encoder-decoder is trained in a parallel original-simplified corpus to get probabilistic word alignments (attention scores α_t) that help determine if a word should or should not be simplified. For these lexical simplifications to take context into consideration, they are integrated into the RL model using linear interpolation following Equation 2.22, where P_{LS} is the probability of simplifying a word.

$$P(y_t|y_{1:t-1},X) = (1-\eta)P_{RL}(y_t|y_{1:t-1},X) + \eta P_{LS}(y_t|X,\alpha_t) \tag{2.22}$$

The model that only uses the RL algorithm (DRESS) and the one that also incorporates the explicit lexical simplifications (DRESS-LS) are trained and tested in three different datasets: PWKP, WikiLarge (with TurkCorpus for testing), and Newsela. The authors compared their models against PBSMT-R, Hybrid, SBSMT (PPDB+SARI), and a standard encoder-decoder with attention (EncDecA). In PWKP, DRESS has the lowest FKGL followed by DRESS-LS. In the TurkCorpus, DRESS and DRESS-LS are second best in FKGL and third best in SARI. In Newsela, DRESS-LS achieves the highest BLEU score. Overall, DRESS and DRESS-LS obtained better scores than EncDecA, with DRESS-LS being the best of the three. It is worth noting that even though there were examples of sentence splitting in the training corpora (e.g. PWKP), the authors do not report on their models being able to perform it.

PointerCopy+MTL: Guo et al. (2018) worked with Sentence Simplification within a Multi-Task Learning (MTL) framework. Considering Sentence Simplification as the main task, they incorporated two auxiliary tasks to improve model's performance: paraphrase generation and entailment generation. The former helps with inducing word and phrase replacements, reorderings and deletions; while the latter ensures that the generated simplified output logically follows the original sentence. The proposed MTL architecture implements multi-level soft sharing (Figure 2.3). Based on observations by Belinkov et al. (2017), lower-level layers in the encoder/decoder (i.e. that are closer to the input/output) are shared among tasks focused on word representations and syntactic-level information (i.e. Sentence Simplification and paraphrasing); while higher-level layers are shared among tasks focused on semantics and meaning (i.e. Sentence Simplification and entailment). In addition, their RNN-based model is enhanced with a pointer-copy mechanism (See et al., 2017), which allows deciding at decoding time whether to copy a token from the input or generate one.

While training main and auxiliary tasks within an MTL framework, a concern is how to determine the appropriate number of iterations on each task relative to the others. Guo et al. (2018) proposed to learn this mixing ratio dynamically using a multi-armed bandits based controller. Basically, at each round, the controller selects a task based on some noise value estimates, observes "rewards" for the selected task (in their case, the reward was the negative validation loss of the main task), and switches accordingly.

The proposed model was trained and tested using PWKP, WikiLarge (with TurkCorpus as test set) and Newsela for Sentence Simplification, the SNLI (Bowman et al., 2015) and the

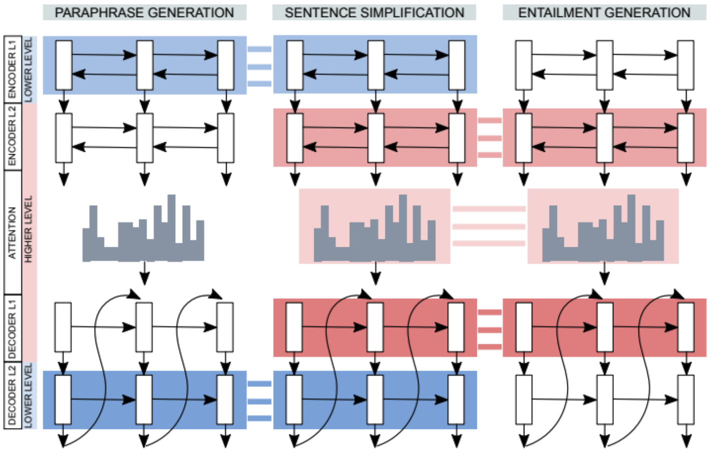


Figure 2.3 Model architecture for PointerCopy+MTL. Extracted from Guo et al. (2018).

MultiNLI (Williams et al., 2018) corpora for entailment generation, and ParaNMT (Wieting and Gimpel, 2018) for paraphrase generation. Using automatic metrics, PointerCopy+MTL achieved the highest SARI score only in the Newsela corpus. With human judgements, their model scored as the best in simplicity.

Adding Side Constraints

Sennrich et al. (2016) proposed to add source-side tokens to the training data of Neural MT models as a way to mark *side constraints* that control the level of politeness of the translation. This idea was further exploited by Johnson et al. (2017) to implement a multilingual Neural MT model capable of performing zero-shot translation. Based on these successes, some work has taken advantage of this simple approach to control specific aspects of the simplification output.

targeTS: Scarton and Specia (2018) enriched the encoder's input with information about the target audience and the (predicted) simplification operations to be performed. Concretely, an artificial token was prepended to the input sentences indicating (1) the grade level of the target simplification, and/or (2) one of four possible simplification operations: identical, elaboration, splitting, or joining. At test time, the operation token is either predicted (using a simple features-based Naive Bayes classifier) or an oracle label is used. They experimented using the standard neural architecture available in OpenNMT and data from the Newsela corpus. Results showed improvements in BLEU, SARI and Flesch when using this extra information. Nishihara et al. (2019) extended this model by adding weights to the loss function based on the frequency or point-wise mutual information of the words in the target grade level. Another extension is presented in (Scarton et al., 2020), where the grade-level and operation information is incorporated as one-hot vectors that initialise the encoder and/or decoder.

ACCESS: Martin et al. (2020) followed the same idea to condition the simplification output on four characteristics: length, amount of paraphrasing, lexical complexity and syntactic complexity. In this case, the artificial tokens contained ratios (target side over source side) for four textual features, respectively: character length (i.e. compression level), normalised character-level Levenshtein similarity, a custom computation of word frequency named WordRank, and the maximum depth of the dependency tree. They implemented this approach using an encoder-decoder Transformer architecture trained on WikiLarge. When compared to previous work on the TurkCorpus test set using automatic metrics, ACCESS achieved the highest SARI score with a version of the model that only relied on the first three attributes.

Adding External Knowledge

The previously described models attempted to learn how to simplify only using information from the training datasets. Zhao et al. (2018) argued that the relatively-small size of these datasets prevents models to generalise well, considering the vast amount of possible simplification operations that exist. Therefore, they proposed to include human-curated paraphrasing rules from Simple Paraphrase Database (SPPDB, Pavlick and Callison-Burch, 2016) into a neural encoder-decoder architecture. This intuition is similar to Xu et al. (2016), who incorporated those rewriting rules into a SBSMT-based model. In addition, the authors moved from the RNN-based architecture to one based on the **Transformer** (Vaswani et al., 2017).

The rewriting rules from SPPD were incorporated into the model using two mechanisms. In Deep Critic Sentence Simplification (DCSS), the model uses a new loss function that maximises the probability of generating the simplified form of a word, while minimising the probability of generating its original one. In Deep Memory Augmented Sentence Simplification (DMASS), the model has a built-in memory that stores the rules from SPPD in the form context vectors calculated from the hidden states of the encoder, and corresponding generated outputs.

The model was trained only using WikiLarge, and tested on TurkCorpus and Newsela. The authors evaluated using both mechanisms, DCSS and DMASS, independently, as well as in conjunction. Then compared to other models, DMASS+DCSS achieved the highest SARI score in both test sets. They also estimated the correctness of rule utilisation based on ground-truth from SPPDB, and showed that their models also improved compared to previous work.

Unsupervised Architectures

Surya et al. (2019) proposed an unsupervised approach for developing simplification systems. Their motivation was to design an architecture that could be exploited to train Sentence

Simplification models for languages or domains that do not have large resources of parallel original-simplified instances. Their proposal is based on a modified auto encoder that uses a shared encoder E and two dedicated decoders: one for generating complex sentences (G_d) and one for simple sentences (G_s). In addition, their model relies on Discriminator and Classifier modules. The Discriminator determines if a given context vector sequence (from either complex or simple sentences) is close to one extracted from simple sentences in the dataset. It interacts with G_s using an adversarial loss function $\mathscr{L}_{adv}$, in a similar fashion as GANs (Goodfellow et al., 2014). The Classifier is in charge of diversification by ensuring, through loss function $\mathscr{L}_{div}$, that both G_d and G_s attend differently to the hidden representations generated by the shared encoder. Two additional loss function, $\mathscr{L}_{rec}$ and $\mathscr{L}_{denoi}$, are used for reconstructing sentences and denoising, respectively. The full architecture can bee seen in Figure 2.4.

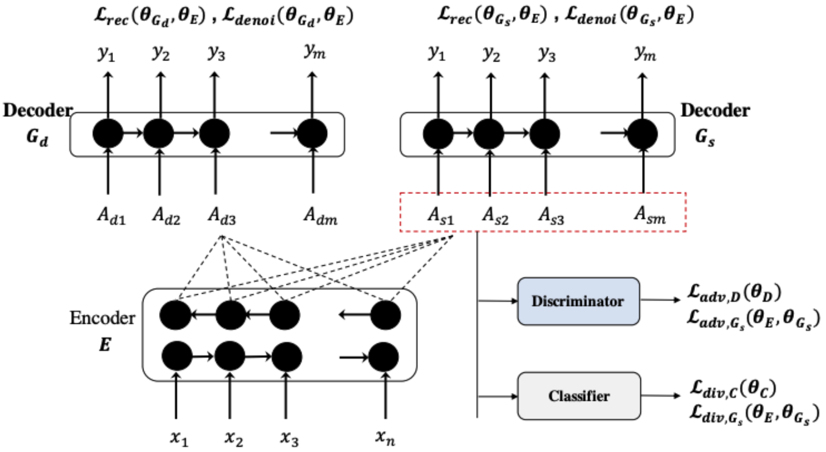

$\mathcal{L}_{rec}(\theta_{G_d},\theta_E)$, $\mathcal{L}_{denoi}(\theta_{G_d},\theta_E)$
$\mathcal{L}_{rec}(\theta_{G_s},\theta_E)$, $\mathcal{L}_{denoi}(\theta_{G_s},\theta_E)$
y_1
y_2
y_3
y_m
Decoder G_d
Decoder G_s
A_{d1}
A_{d2}
A_{d3}
A_{dm}
A_{s1}
A_{s2}
A_{s3}
A_{sm}
Encoder E
Discriminator
$\mathcal{L}_{adv,D}(\theta_D)$
$\mathcal{L}_{adv,G_s}(\theta_E,\theta_{G_s})$
Classifier
$\mathcal{L}_{div,C}(\theta_C)$
$\mathcal{L}_{div,G_s}(\theta_E,\theta_{G_s})$
x_1
x_2
x_3
x_n

Figure 2.4 Model architecture for UNTS. Extracted from Surya et al. (2019).

The proposed model (UNTS) was trained using an English Wikipedia dump that was partitioned into Complex and Simple sets using a threshold based on Flesch Reading Ease scores. They also used 10,000 sentence pairs from EW-SEW (Hwang et al., 2015) and Web-Split (Narayan et al., 2017) datasets to train a model (UNTS+10K) with minimal supervision. Their models were compared against unsupervised systems from the MT literature (Artetxe et al., 2018a,b), as well as Sentence Simplification models like NTS (Nisioi et al., 2017) and SBSMT (Xu et al., 2016), and using TurkCorpus as test data. When evaluated using automatic metrics, SBSMT scored the highest on SARI, but both UNTS and UNTS+10K were not far from the supervised models. This same behaviour was observed with human evaluations. Even though the unsupervised model was trained using instances of sentence splitting from WebSplit, the authors do not report testing it on data for that specific simplification operation.

Table 2.6 summarises the performance of the models described in this Section. In this case, some of the values can be compared on the test set used. Of all the sequence-to-sequence models tested in TurkCorpus, ACCESS obtains the best SARI score, with NseLstm-B achieving the highest BLEU. The second best model in terms of SARI is DMASS-DCSS that incorporated paraphrasing knowledge from SPPDB. All these approaches used WikiLarge for training. Regarding the operations that they can perform, sequence-to-sequence models seem to be able to perform substitutions, deletions and reorderings, just like previous MT-based approaches. None of the papers reports if these models were tested in data specific for sentence splitting.

Table 2.6 Self-reported performance of neural sequence-to-sequence Sentence Simplification models.

Model	Train Corpus	Test Corpus	BLEU ↑	FKGL ↓	SARI ↑
NTS	EW-SEW	TurkCorpus	84.51		30.65
NTS+SARI	EW-SEW	TurkCorpus	80.69		37.25
NseLstm-B	WikiSmall	PWKP	53.42		17.47
	WikiLarge	TurkCorpus	92.02		33.43
	Newsela	Newsela	26.31		27.42
NseLstm-S	WikiSmall	PWKP	29.72		29.75
	WikiLarge	TurkCorpus	80.43		36.88
	Newsela	Newsela	22.62		29.58
Dress	WikiSmall	PWKP	34.53	7.48	27.48
	WikiLarge	TurkCorpus	77.18	6.58	37.08
	Newsela	Newsela	23.21	4.13	27.37
Dress-Ls	WikiSmall	PWKP	36.32	7.55	27.24
	WikiLarge	TurkCorpus	80.12	6.62	37.27
	Newsela	Newsela	24.30	4.21	26.63
PointerCopy+MTL	WikiSmall	PWKP	29.70	6.93	28.24
	WikiLarge	TurkCorpus	81.49	7.41	37.45
	Newsela	Newsela	11.86	1.38	32.98
targeTS	Newsela	Newsela	64.78		45.41
ACCESS	WikiLarge	TurkCorpus		7.22	41.87
DMASS+DCSS	WikiLarge	TurkCorpus		8.04	40.45
	WikiLarge	Newsela		5.17	27.28
UNTS	Wikipedia Dump	TurkCorpus	74.24		33.80
UNTS+10K	Wikipedia Dump	TurkCorpus	76.13		35.29

2.4.5 Discussion

In this Section we have surveyed different models for data-driven automatic Sentence Simplification. This review has helped to understand the benefits and shortcomings of each approach to the task. Traditionally, Sentence Simplification has been reduced to four simplification operations: replacing complex words or phrases, deleting or reordering sentence components, and splitting a complex sentence into several simpler ones (Narayan and Gardent, 2014; Zhu et al., 2010). Table 2.7 lists the reviewed Sentence Simplification models (grouped by approach), the techniques each of them explores, and the simplification operations that they can perform, considering the four traditional rewriting operations established.

Overall, SMT-based methods can perform replacements, short-distance reorderings and deletions, but fail to produce quality splits unless explicitly modelled using syntactic information or coupled with more expensive processes, such as semantic analysis. Grammar-based approaches can model splits (and syntactic changes in general) more naturally, but the critical process of selecting which rule(s) to apply, in which order, and how to determine the best simplification output is complex. In this respect, neural sequence-to-sequence approaches seem more straightforward, since they offer end-to-end architectures that can potentially learn to generate all operations simultaneously. However, properly training neural models requires access to large amounts of data, which is not a resource that exists in Sentence Simplification.

2.5 Summary and Final Remarks

In this Chapter, we have presented a survey of research in Sentence Simplification. We limited our review to models that learn to produce simplifications by learning from parallel corpora of original-simplified sentences. We described the main resources exploited by these approaches, detailed how models are trained using these datasets, and explained how they are evaluated. Based on this comprehensive review of the literature, we identify a few gaps in the area that motivate the research performed as part of this Thesis.

First, we observed that studies on how humans simplify identified that several rewriting operations are performed simultaneously. These included lexical changes (e.g. replacing words or phrases per simpler synonyms), sentence-level modifications (e.g. splitting a sentence or reordering its components), and even document-level alterations (e.g. joining sentences). Other tasks in NLP research (e.g. compression or split-and-rephrase) are strongly-related to these operations. As such, it could be possible to take advantage of their models and/or data to enhance the multi-operation capabilities of Sentence Simplification systems. We explore this idea in Chapter 6 through the implementation of a multi-task model.

We then described the main corpora used for training and evaluating Sentence Simplification models. They consist of automatic alignments extracted from (Simple) English Wikipedia and the Newsela corpus. Since they are prone to be noisy, we pointed out the issues that could arise during model implementation, such as training conservative systems. More importantly,

Table 2.7 Summary of Sentence Simplification approaches: SMT-based (first section), grammar-based (second section), semantics-assisted (third section), and neural sequence-to-sequence (fourth section). The operations listed are the ones the authors acknowledge that their models can perform. Operations with * are found in some of the outputs, but not explicitly modelled by the authors.

Model	Approach	Operations
Specia (2010)	PBSMT (Moses)	REP, REORD
Coster and Kauchak (2011b)	PBSMT (Moses)	REP, REORD
Coster and Kauchak (2011a)	PBSMT (Moses) + Deletion	REP, DEL, REORD
Wubben et al. (2012)	PBSMT (Moses) + Dissimilarity Ranking	REP
Zhu et al. (2010)	SBSMT	REP, DEL, REORD, SPLIT
Bach et al. (2011)	SBSMT	SPLIT
Xu et al. (2016)	SBSMT (Joshua) + SPPDB + SARI Optimisation	REP, REORD
Woodsend and Lapata (2011a)	QSTSGs + ILP	REP, DEL, REORD, SPLIT
Paetzold and Specia (2013)	STSGs + Perplexity Ranking	REP, DEL, REORD, SPLIT*
Feblowitz and Kauchak (2013)	STSGs Backoff + Log-linear Reranking	REP, DEL, REORD, SPLIT*
Narayan and Gardent (2014)	Deep Semantics (Boxer) + PBSMT	REP, DEL, REORD, SPLIT
Narayan and Gardent (2016)	Lexical Simp. + Deep Semantics (Boxer) + ILP	REP, DEL, SPLIT
Štajner and Glavaš (2017)	Event Detection for Splitting + Unsupervised Lexical Simplification	SUB, DEL, SPLIT
Narayan et al. (2017)	Semantics-aided Splitting + NTM	REP, REORD, SPLIT
Nisioi et al. (2017)	Seq2Seq (RNN) + PPDB + SARI	SUB, DEL, REORD
Zhang and Lapata (2017)	Seq2Seq (RNN) + RL	REP, DEL, REORD
Vu et al. (2018)	Seq2Seq (RNN) with NSE	REP, DEL, REORD
Guo et al. (2018)	Seq2Seq (RNN) + MTL	REP, DEL, REORD
Scarton and Specia (2018)	Seq2Seq (RNN) + Side Constraints	REP, DEL, REORD, SPLIT
Zhao et al. (2018)	Seq2Seq (Transformer) + SPPDB	REP, DEL, REORD
Martin et al. (2020)	Seq2Seq (Transformer) + Side Constraints	REP, DEL, REORD, SPLIT
Surya et al. (2019)	Seq2Seq + Unsupervised Training	REP, DEL, REORD

automatic alignments are less than ideal for evaluation purposes. Some work has involved collecting high-quality manual simplifications, such as for TurkCorpus and HSplit. However, they are limited in the simplification operations expressed in their instances (i.e. lexical paraphrasing for TurkCorpus and sentence splitting for HSplit). As such, in Chapter 4 we collect a new dataset that is bigger in size and with more diverse rewriting transformations.

We also presented the different methods used for evaluation of automatic simplifications. We explained the criteria used for collecting human judgements on grammaticality, meaning preservation and simplicity; and also the different automatic metrics that are computed for a practical assessment and comparison between models. We highlighted the limitations of these methods and proposed ways in which they could be improved. For automatic metrics, in particular, it is still undetermined which ones should be used, under what circumstances, and how to interpret their results. We propose an answer in Chapter 5 after a meta-evaluation study.

Finally, we elaborated a comprehensive survey on models for data-driven Sentence Simplification. We grouped the automatic systems into: relying on statistical machine translation architectures, induction of synchronous grammars, semantics-assisted, and neural sequence-to-sequence models. For each group, we explained how representative systems worked, and tried to compare them based on their self-reported automatic scores and ability to perform specific simplification operations. However, an objective comparison was not entirely possible since not every system was tested on the same dataset, nor used the same implementations of automatic metrics, nor reported on the operations that were actually correctly applied to input sentences. In Chapter 3, we develop algorithms and evaluation tools that allow a more informative and objective benchmarking of simplification models.

Chapter 3

Identification of Simplification Operations

The usual method for automatic evaluation of simplification systems is to use metrics that estimate the overall quality of the generated simplifications based on their similarity to human references. However, these scores do not provide insights about the strengths and weaknesses of each system, especially regarding the simplification operations that they are able to execute. In this Chapter, we attempt to answer the following research question: **Which simplification operations do automatic systems perform accurately?** We propose to automatically annotate the text transformations that generated a simplification output, and compare them to those present on manual references. We show that this per-operation error analysis contributes to a better understanding of the scores given by automatic evaluation metrics, and to an improved comparison between different simplification systems.

First, we conducted a manual exploration on automatically-aligned sentences from a simplification corpus produced by professional editors (Sec. 3.1). This study allowed to identify the most common simplification operations to focus on for the rest of our research. Then, we implemented novel algorithms to automatically annotate these frequent operations at word, phrase and sentence levels (Sec. 3.2). These automatic annotations were evaluated intrinsically (Sec. 3.2.2) and extrinsically as part of a simplification model based on sequence labelling (Sec. 3.2.3). Furthermore, we propose to exploit these operation labels for analysing the simplification errors of different models (Sec. 3.3). With this objective, we first introduce EASSE (Sec. 3.3.1), a software package for standardising automatic evaluation in Sentence Simplification, that also includes our operation labelling algorithms. Then, we carry out (to the best of our knowledge) the first benchmark of Sentence Simplification systems using automatic metrics for overall evaluation and per-operation error analysis (Sec. 3.3.2). Finally, we offer some remarks about the usefulness of our operation-based algorithms for better understanding the performance of automatic simplification systems (Sec. 3.4).

3.1 Manual Analysis of Human Simplifications

Our first goal is to understand how humans simplify texts. In particular, we would like to know which simplification operations are performed and at what granularity. As presented in Sec. 2.2, a few studies with this objective have already been carried out. However, most of them were focused on datasets extracted from EW and SEW (Amancio and Specia, 2014; Coster and Kauchak, 2011b; Xu et al., 2015), or only considered a limited set of simplification operations even with a higher-quality dataset (Xu et al., 2015).

We decided to conduct a manual analysis on the Newsela corpus (Xu et al., 2015), considering that it was produced by professional editors, and that it is supposed to contain more consistent and reliable simplifications. Articles in this corpus have up to 5 levels of simplicity, with level 0 being the original text and level 5 being the most simplified version (see details in Sec. 2.2.2). There are no publicly-available guidelines on how the simplification process was carried out by the editors. As such, in our analysis we assume that to produce a version of an article for a specific simplification level, the editor is based on the version of that article from the immediately previous level, i.e. 0→1, 1→2, 2→3, etc. This is because always simplifying "from scratch" (i.e. from level 0) could be inefficient, as deciding what content to keep/delete or writing elaborations for complex concepts are not trivial tasks. Having some of those operations already performed in a previous simplification of a text allows the editor to focus on incorporating changes that can further improve readability instead of executing them again. In addition, manual examination of the documents shows that it is rare to find cases where some information from level 0 appears in a higher level without also appearing in an intermediate one. Taking all those aspects into consideration, we believe our assumption is valid. We follow (Xu et al., 2015) and analyse the operations in automatically-aligned original-simplified sentence pairs, but with two main differences: (1) we use more sophisticated algorithms to extract the automatic alignments; and (2) we identify a richer set of simplification operations.

3.1.1 Automatic Sentence Alignments

In order to extract original-simplified sentence pairs between different simplification levels of articles in the corpus, we used the vicinity-driven alignment algorithms of (Paetzold and Specia, 2016c).[1] Automatic alignments produced by these algorithms have also been used in research involving training simplification models with the Newsela corpus (Alva-Manchego et al., 2017; Scarton et al., 2018b; Scarton and Specia, 2018). The extracted sentence alignments can be

[1] These algorithms are explained in Sec. 2.2.2. At the time this study was conducted, the more sophisticated alignment methods of (Štajner et al., 2018) and (Jiang et al., 2020) were not available.

classified as **identical** (1-to-1 alignments where the original and simplified sentences are the same), **1-to-1** (1-to-1 alignments where the original and simplified sentences are different), **1-to-N** (splits), and **N-to-1** (joins). Table 3.1 presents the number of alignment types extracted from each simplification level pair in the corpus.

Table 3.1 Number of sentence alignments types per level-pair in our aligned Newsela corpus.

Level Pair	Identical	1-to-1	1-to-N	N-to-1	Total
0→1	29,199 (42%)	27,549 (40%)	10,660 (15%)	2,035 (3%)	69,443
1→2	28,664 (38%)	34,124 (45%)	11,757 (15%)	1,408 (2%)	75,953
2→3	20,420 (29%)	36,192 (52%)	10,613 (15%)	2,191 (3%)	69,416
3→4	18,425 (30%)	32,336 (52%)	9,372 (15%)	2,280 (4%)	62,413
4→5	201 (20%)	590 (60%)	143 (15%)	48 (5%)	982

The number of identical sentence alignments in level pair 0→1 corresponds to around 42% of the total alignments in that level pair, and it is the highest quantity among the other types of alignments. However, this proportion decreases as we move up the level pairs. This could mean that the editor needs to perform more changes to the sentences as the level of simplicity increases. Across all level pairs, 1-to-1 alignments are more common than 1-to-N or N-to-1, indicating the relevance of paraphrasing operations over sentence splitting or joining. Furthermore, the lower number of N-to-1 among alignment types could indicate that, even though some summarisation is performed, it is not as significant as the other operations. The low number of sentence alignments in level pair 4→5 indicates that, for the version of the corpus used in this study, very few articles had a version in the highest level of simplification.

3.1.2 Manual Annotation

Our analysis of aligned sentences was conducted in different simplification levels of the Newsela corpus. From each level pair, we randomly selected 50 sentence pairs corresponding to 1-to-1, 1-to-N and N-to-1 alignments. For each sentence pair, we identified the simplification operations applied to the original sentence. If an operation was performed more than once in the sentence (e.g. more than one word was replaced), it was still registered as one.

The simplification operations considered are those identified by Caseli et al. (2009). This study considers a broader set of transformations than Xu et al. (2015), without being too specific as Amancio and Specia (2014). A summary of our findings can be seen in Table 3.2. For our analysis, we consider SIMPLE REWRITE as adding a connector (e.g. *by*), adding a punctuation mark, changing the tense of the main verb, or replacing a proper noun with a pronoun. On the other hand, STRONG REWRITE considers cases where some part (or all) of the sentence has been completely modified, even though the meaning remains almost similar.

Table 3.2 Number of sentences per level pair where the specific simplification operation was performed. More than one operation could have been performed in each sentence.

Operation / Level Pair	0→1	1→2	2→3	3→4	4→5	Total
DELETE: delete words or phrases	14	12	23	22	29	100
REPLACE: substitution of words	25	28	16	11	16	96
SIMPLE REWRITE: minor replacements	9	15	15	8	14	65
ADD: add information	12	3	9	16	10	50
SPLIT: splitting sentences	12	9	9	7	6	43
REORDER: reordering of words	7	4	8	5	4	28
STRONG REWRITE: complete rephrasing	5	4	5	7	7	28
DROP-C: dropping a clause	6	7	5	4	2	24
JOIN: joining sentences	2	1	1	1	1	6
REORDER-SVO: subject-verb-object reordering	0	1	0	1	0	2
PASS2ACT: passive to active voice	0	0	0	0	1	1
JOIN-C: joining clauses in same sentence	1	0	0	0	0	1

As was also shown in (Xu et al., 2015), the most common simplification operations are the ones related with compression (i.e. DELETE) and lexical paraphrasing (i.e. REPLACE, REORDER and SIMPLE REWRITE). The only "important" operation regarding syntactic alterations is SPLIT. There is also a medium presence of ADD, which corresponds to including explanations of concepts or external information. Because of these additions, deletions and splits, it is necessary to make adjustments to the text, so as to preserve grammaticality/fluency. Simple and (some) strong rewrites are, thus, a consequence of performing the previous operations. The almost absence of subject-verb-object reordering and passive to active voice transformations is surprising, since they are regarded as very important operations that make a text easier to read. With further inspection, we found that they were not performed because they were not needed, and not because the editors neglected them. This is probably due to a characteristic of the Newsela corpus (news articles) and not of the simplification process carried out.

3.2 Automatic Annotation of Human Simplifications

Manual identification of simplification operations is costly and time-consuming, despite providing us with useful information. In this Section, we propose novel annotation algorithms that can automatically identify common simplification operations in different granularities (e.g. word-level and sentence-level) and with relatively-high confidence.

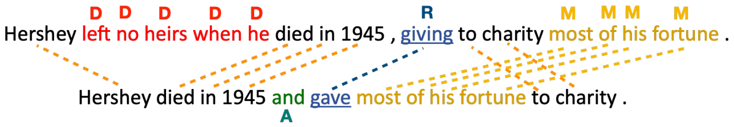

D D D D D
R
M M M M
Hershey left no heirs when he died in 1945 , giving to charity most of his fortune .
Hershey died in 1945 and gave most of his fortune to charity .
A

Figure 3.1 Example of automatic word-level annotations of simplification operations between an original (top) and a simplified (bottom) sentence. Labels 'D', 'R', 'A' and 'M' correspond to DELETE, REPLACE, ADD and REORDER, respectively. The dotted lines correspond to the word alignments.

3.2.1 Annotation Algorithms

We implement algorithms that attempt to identify the most common simplification operations according to our manual study: DELETE, REPLACE, ADD, REORDER, and SPLIT. Given an original sentence and its simplification, our algorithms use constituency-based parsing and word alignments to annotate the operations at word, phrase and sentence levels. For parsing we use Stanford CoreNLP (Manning et al., 2014), and test two different word aligners in Sec. 3.2.2.

Word-Level Annotations

Our algorithms first annotate word-level substitutions, deletions and additions with the following intuition: if two words are aligned and are not an exact match, the word in the original sentence receives a REPLACE tag; if a word in the original sentence is not aligned, it is annotated as a DELETE; and if a word in the simplified sentence is not aligned, it is annotated as an ADD. The implementation of this method is presented in Algorithm 1.

There may be some cases of REPLACE where two synonyms are not aligned, and they end up receiving incorrect labels (i.e. DELETE and ADD in the original and simplified sentences, respectively). The simplest case is when the aligned sentences are almost identical, except for the lexical substitution. In order to improve the REPLACE labeling in these situations, we employ a simple heuristic: for every word in the original sentence labeled as DELETE, we check if there is a word in the modified sentence that (1) is labeled as ADD, (2) has the same position in the sentence, and (3) has the same part-of-speech tag. If these criteria are met, then the word's label is changed to REPLACE. We then proceed to label REORDERings by determining if the relative index of a word (considering preceding or following DELETEs and ADDs) in the original sentence changes in its simplified counterpart. Algorithm 2 implements this method. Some words that are replaced may be subject to reorderings as well. As such, it is possible that a token requires more than one label (e.g. REPLACE and REORDER). Preliminary experiments showed that allowing multiple labels for a word was more prone to errors in the final annotation. As such, we decided to prioritise the most common operation, i.e. REPLACE. Figure 3.1 exemplifies an original-simplified sentence pair with word-level annotations.

Algorithm 1: Initial word-level annotation

```
Input: $O$: list with tokens of the original sentence, $S$: list with tokens of the simplified sentence, $A$: list
    with word alignments.
Output: $SLO$: simplification labels for each token in $O$, $SLS$: simplification labels for each token in $S$.
// labeling tokens in the original sentence
for $i \leftarrow 1$ to $len(O)$ do
    // find the tokens in S aligned to the i-th token in O
    $IS \leftarrow$ FindAlignments($A$, $i$, ‘$s$’)
    if $len(IS) > 0$ then                                          // it is aligned
        if $len(IS) = 1$ and $O_i = S_{IS_0}$ then
            $SLO_i \leftarrow$ ‘O’                                      // keep
        else                                                  // not an exact match
            $SLO_i \leftarrow$ ‘R’                                      // replace
        end
    else                                                            // not aligned
        $SLO_i \leftarrow$ ‘D’                                          // delete
    end
end
// labeling tokens in the simplified sentence
for $j \leftarrow 1$ to $len(S) + 1$ do
    // find the tokens in O aligned to the j-th token in S
    $IO \leftarrow$ FindAlignments($A$, $j$, ‘$o$’)
    if $len(IO) > 0$ then                                          // it is aligned
        $SLS_j \leftarrow$ ‘O’
        if $len(IO) > 1$ then
            // the current token in S replaces a phrase in O
            foreach $k \in IO$ do
                $SLO_k \leftarrow$ ‘R’
            end
        end
    else
        $SLS_j \leftarrow$ ‘A’                                              // add
    end
end
```

Phrase-Level Annotations

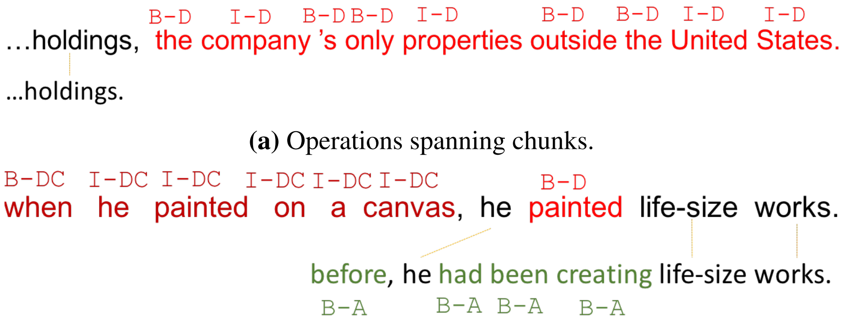
So far, the annotations have been performed on single words. However, we are also interested in capturing operations that span across syntactic units, such as phrases or clauses. To this end, we group repeated operation labels for entire syntactic units using IOB notation. The constituent parse trees of the aligned original and simplified sentences are used for this purpose. Our intuition is that if the majority of words within a syntactic unit have the same label, then the whole unit should receive a unit-level operation label (e.g. DELETE CLAUSE).[2] Algorithm 3 is used to label clauses and chunks, but in the latter case we do not use a particular unit label,

[2] We consider ‘majority’ as at least 75% to counteract the effect of incorrect labels due to word misalignments.

Algorithm 2: Annotation of reorderings.

```
Input: $SLO$: simplification labels for each token in original sentence, $SLS$: simplification labels for each
       token in simplified sentence, $A$: list with word alignments.
Output: $SLO$ modified.
$shift\_left \leftarrow 0$
for $i \leftarrow 0$ to $len(SLO)$ do
    if $SLO_i \in [\text{'D'}, \text{'R'}]$ then
        $shift\_left \leftarrow shift\_left + 1$
    else
        $IS \leftarrow$ FindAlignments($A$, $i$, 's')
        if $len(IS) > 0$ then
            $k \leftarrow IS_0$        // index of the aligned token in the simplified sentence
        else
            $k \leftarrow i$           // index of the token in the original sentence
        end
        $shift\_right \leftarrow 0$
        for $j \leftarrow 0$ to $k$ do
            if $SLS_j$ = 'A' then
                $shift\_right \leftarrow shift\_right + 1$
            end
        end
        if $i - shift\_left + shift\_right \neq k$ then
            if $SLS_i$ = 'O' then
                $SLS_i \leftarrow$ 'M'              // reorder or move
            end
        end
    end
end
```

since we are only interested in chunks being labeled as IOB.[3] Figure 3.2a presents an example of a DELETE labeling in chunks, while Figure 3.2b shows the unit label DELETE CLAUSE.


B-D I-D B-D B-D I-D B-D B-D I-D I-D
…holdings, the company 's only properties outside the United States.
...holdings.
B-DC I-DC I-DC I-DC I-DC I-DC B-D
when he painted on a canvas, he painted life-size works.
before, he had been creating life-size works.
B-A B-A B-A B-A

(a) Operations spanning chunks.

(b) Operations spanning a clause.

Figure 3.2 Examples of automatic phrase-level annotations of simplification operations, where an operation label spans across a syntactic unit, such as a chunk or a clause.

[3] We define a 'chunk' as in CoNLL 2000: http://www.cnts.ua.ac.be/conll2000/chunking.

Algorithm 3: Span-level annotation

```
Input: $SL$: list of labels for each token in the sentence, $P$: constituent parse tree of the sentence, $G$: list
    of group syntactic tags, $O$: list of operation labels to look for, $nl$: new span label to apply.
Output: $SL$ modified.
for $w \leftarrow 0$ to $len(SL)$ do
    if $SL_w \in O$ then
        $st \leftarrow$ GetSubTree($P, w$)
        $pt \leftarrow$ GetParent($P, st$)                                   // a tree
        while $pt \neq nil$ and GetTag($pt$) $\notin G$ do
            $pt \leftarrow$ GetParent($P, pt$)
        end
        if $pt \neq nil$ then                                          // the tree exists
            $begin \leftarrow$ index of the token where the leaves of $pt$ begin
            $same\_count \leftarrow$ number of leaves of $pt$ for which $SL_i \in O$
            if $same\_count$ is the majority of the number of leaves of $pt$ then
                $end \leftarrow begin + same\_count$
                $SL_{begin} \leftarrow$ 'B-' + $nl$
                for $i \leftarrow begin + 1$ to $end$ do
                    $SL_i \leftarrow$ 'I-' + $nl$
                end
            end
        end
    end
end
```

Sentence-Level Annotations

We exploit the word-level annotations to generate sentence-level operations labels. Following the same idea of our manual study (Sec. 3.1), if at least one word was labeled with a particular simplification operation, then we register that operation for the whole sentence. This logic is applied to identify DELETE, REPLACE, ADD, and REORDER. For SPLIT, we compute the number of sentences in the original and simplified sides using nltk,[4] and register a SPLIT if the number in the simplified side is higher than the one in the original side.

3.2.2 Intrinsic Evaluation

We first assess the quality of the automatic annotations generated by our algorithms by comparing them to manual annotations.

Test Datasets

We collected gold labels for the simplification operations of interest in different granularities:

[4] https://www.nltk.org/api/nltk.tokenize.html

- *Word and Phrase Level Annotations:* We randomly selected 100 automatically-aligned original-simplified sentence pairs from level pair 0-1 of the Newsela corpus. We asked four proficient English speakers to annotate the simplification operations previously defined using IOB notation. Of the 100 sentences, 70 were annotated only once and the remaining 30 were annotated by all four participants. For this last subset, we calculated the pairwise inter-annotator agreement, obtaining an average kappa value of 0.62. According to the scale in (Landis and Koch, 1977), this is a moderate level of agreement that indicates that the task is reproducible up to some degree.

- *Sentence-Level Annotations:* We use the annotations collected in our initial manual analysis (Sec. 3.1): 50 original-simplified sentence pairs per each of the 5 level pairs (0→1, 1→2, 2→3, 3→4, and 4→5) in the Newsela corpus, thus totalling 250 sentences.

Word Alignments

Our annotation algorithms depend heavily on the word alignments given as input. As such, we experimented with two tools for the task:

- *Monolingual Word Aligner* (MWA, Sultan et al., 2014): It is based on the idea that words with similar meanings are potentially aligned if they are located in similar contexts. MWA follows a simple pipeline: align identical words, align named entities, align content words, and align stop words. Similarity is measured in three levels: exact word or lemma match, similarity in the Paraphrase Database (PPDB, Ganitkevitch et al., 2013) if words are not an exact match, and no similarity if the pair does not exist in the PPDB. The contextual information for each word is extracted from its sentence's dependency tree. In our experiments, we modified MWA so that it is compatible with Python 3, and uses a more current version of Stanford CoreNLP (Manning et al., 2014)[5] for syntactic parsing.

- *SimAlign* (Sabet et al., 2020): Leverages contextualised word embeddings (i.e. BERT (Devlin et al., 2019)-like models) to align words. Given a pair of sentences, the tool first creates a similarity matrix between all words in each sentence using normalised cosine similarity. Then, word alignments are extracted using three methods: (1) **argmax**, where a word in a sentence is aligned to the most similar in the other one; (2) **itermax**, where an initial alignment based on argmax is iteratively refined to reduce the number of unaligned words; and (3) **match**, where word alignment is treated as a maximum weight matching

[5] `https://stanfordnlp.github.io/stanfordnlp/corenlp_client.html`

problem in order to find a global optimum alignment. Preliminary experiments showed that argmax and Multilingual BERT[6] yield the best results for our purposes.

Results

We ran the annotation algorithms with each word alignment output, and compared the corresponding automatic labels against the manual annotations in each test set. As such, we measure the performance of the algorithms in terms of Precision (P), Recall (R) and F1 score for each operation label in each annotation level (i.e. word, phrase and sentence).

For **word-level** operation labels (Table 3.3), SimAlign provides word alignments that allow the annotation algorithms to perform better than when using MWA in almost all cases. This could be a consequence of using BERT word embeddings, since more-informed word representations lead to word alignments of higher accuracy, even if using simple cosine similarity for matching. The main beneficiary is the REPLACE operation, with a significant improvement in recall of synonymous words, whilst the operation with the worst performance is REORDER. The confusion matrix in Figure 3.4a shows that the main source of errors is labelling a word as REORDER when no operation was performed according to the manual annotations. This is mainly caused by misalignments, such as the one shown in Figure 3.3, where the alignment of "is" with "was" causes the former to be considered as a replacement that changed position and, therefore, the algorithms interpret that all aligned words that come after the substitution were also reordered.

Table 3.3 Performance of the annotation algorithms for word-level operation labels.

			MWA			SimAlign		
Operation	Label	Frequency	P	R	F1	P	R	F1
DELETE	D	380	0.77	0.91	0.83	0.86	0.86	**0.86**
REPLACE	R	123	0.73	0.31	0.43	0.72	0.66	**0.69**
ADD	A	270	0.65	0.94	0.77	0.73	0.87	**0.79**
REORDER	M	25	0.24	0.92	**0.39**	0.16	0.84	0.26
	micro avg	798	0.67	0.83	0.74	0.70	0.83	**0.76**

For **phrase-level** operation labels (Table 3.4), MWA provides (slightly) more helpful word alignments than SimAlign for annotating the majority of operations. Taking all labels into consideration, the micro average F1 scores for both alignment tools are comparable. However, same as in the word-level scenario, SimAlign's alignments are more advantageous

[6] https://github.com/google-research/bert/blob/master/multilingual.md

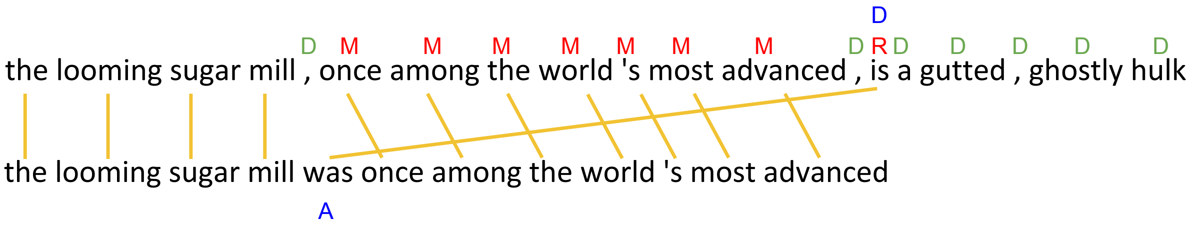
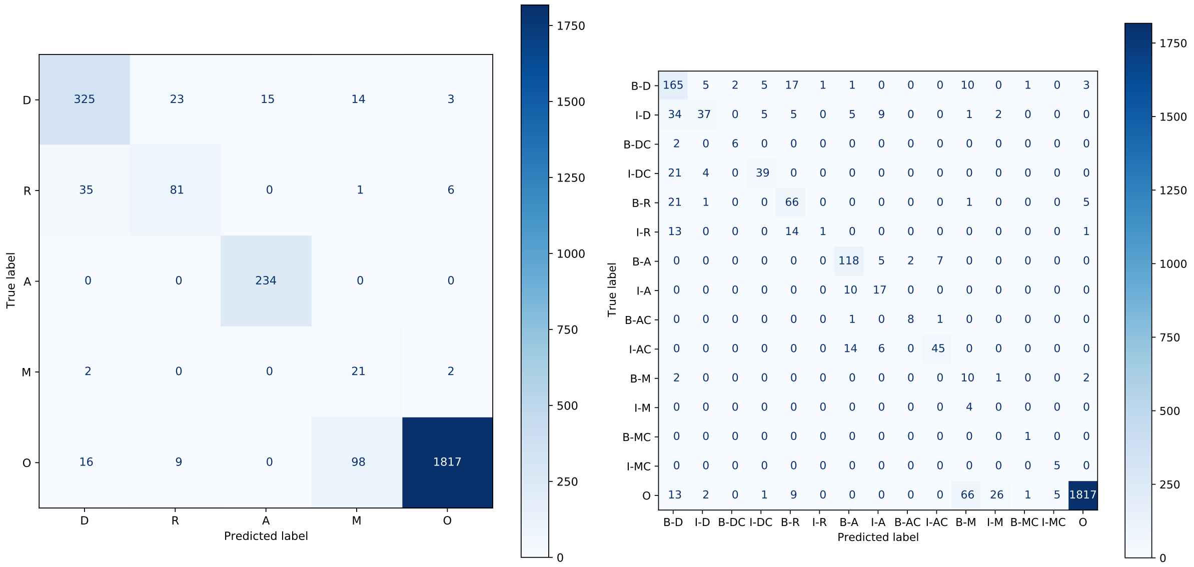

D
D M M M M M M M D R D D D D D
the looming sugar mill , once among the world 's most advanced , is a gutted , ghostly hulk
the looming sugar mill was once among the world 's most advanced
A

Figure 3.3 An example of misalignment causing an error in the annotation of the word-level operation label REORDER. Green: agreement between automatic and manual annotation; Red: incorrect automatic labels; Blue: labels missed by the automatic algorithms.



(a) Confusion matrix for word-level annotations.

(b) Confusion matrix for phrase-level annotations.

Figure 3.4 Confusion matrices for automatic annotations of word-level and phrase-level operation labels using word alignments from SimAlign. Label 'O' indicates that no operation was performed.

for identifying REPLACE operations. Since this is one of the most important operations to annotate, we conclude that SimAlign's alignments are the most suitable for our annotation algorithms. Both type of alignments allow for high scores in identification of clause-related labels (AC, DC, MC), despite them being less frequent than the other operation labels. After manual analysis, we observed that the annotation mistakes for these labels are caused by the parser not correctly identifying clauses. The confusion matrix in Figure 3.4b shows, once again, that the main source of errors is labelling REORDER when no operation was performed, also caused by misalignments as in the case of the word-level operation labels. Finally, even though the phrase-level labels allow for more fine-grained annotations, the overall performance of the

Table 3.4 Performance of the annotation algorithms for phrase-level operation labels.

			MWA			SimAlign		
Operation	Label	Frequency	P	R	F1	P	R	F1
DELETE	B-D	210	0.57	0.88	**0.69**	0.61	0.79	0.68
	I-D	98	0.72	0.47	**0.57**	0.76	0.38	0.50
	B-DC	8	0.67	0.75	0.71	0.75	0.75	**0.75**
	I-DC	64	0.75	0.64	**0.69**	0.78	0.61	0.68
REPLACE	B-R	94	0.59	0.32	0.41	0.59	0.70	**0.64**
	I-R	29	1.00	0.03	**0.07**	0.50	0.03	0.06
ADD	B-A	160	0.49	0.86	**0.63**	0.55	0.74	**0.63**
	I-A	34	0.43	0.68	**0.52**	0.40	0.50	0.44
	B-AC	10	0.78	0.70	0.74	0.80	0.80	**0.80**
	I-AC	66	0.82	0.64	0.72	0.85	0.68	**0.76**
REORDER	B-M	15	0.18	0.80	**0.30**	0.11	0.67	0.19
	I-M	4	0.00	0.00	0.00	0.00	0.00	0.00
	B-MC	1	1.00	1.00	**1.00**	0.33	1.00	0.50
	I-MC	5	1.00	1.00	**1.00**	0.50	1.00	0.67
	micro avg	798	0.54	0.67	**0.60**	0.55	0.65	0.59

algorithms (F1 = 0.60) is lower than that for the word-level annotations (F1 = 0.76). As such, we recommend the latter types of annotations for a more reliable analysis.

For **sentence-level** operation labels (Table 3.5), based on the previous findings, we only present results using word alignments from SimAlign in the corresponding test set (which is different from the one for word-level and phrase-level annotations).

Table 3.5 Performance of the annotation algorithms for sentence-level operation labels.

		All Sentences				Without REWRITEs			
Operation	Label	Frequency	P	R	F1	Frequency	P	R	F1
DELETE	D	119	0.57	0.94	0.71	112	0.63	0.95	**0.76**
REPLACE	R	93	0.50	0.96	0.66	114	0.69	0.90	**0.78**
ADD	A	48	0.30	1.00	0.46	72	0.51	0.96	**0.67**
REORDER	M	26	0.41	0.92	0.56	26	0.57	0.92	**0.71**
SPLIT	S	43	0.75	1.00	0.86	41	0.77	1.00	**0.87**
	micro avg	329	0.49	**0.96**	0.65	365	**0.63**	0.94	**0.75**

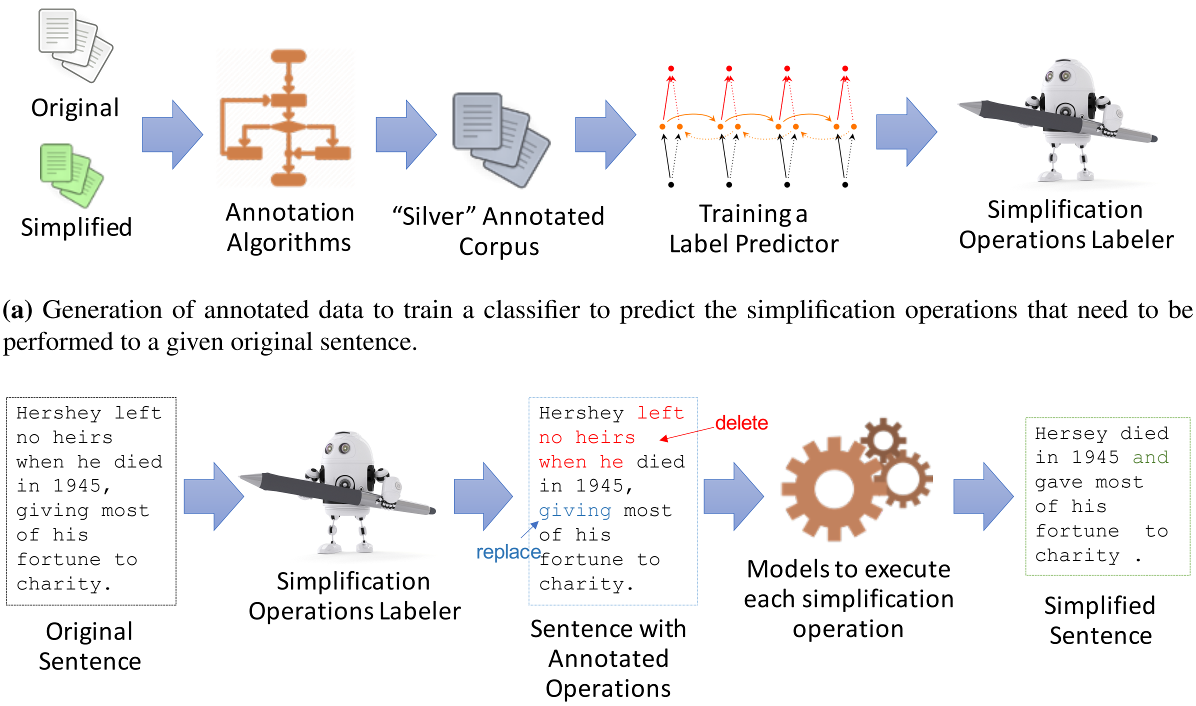
When using “All Sentences”, we notice high recall values for all operation labels. However, precision scores are low except for SPLIT, the easiest operation to identify. After manual

Table 3.6 Annotation errors due to manual labelling of STRONG REWRITE and SIMPLE REWRITE.

	Original Sentence	Simplified Sentence	Automatic Labels
STRONG REWRITE	Stockholm - Sweden's biggest submarine hunt **since the dying days of the Soviet Union has put countries around the Baltic Sea on edge**.	Stockholm - Sweden's biggest submarine hunt **is giving countries along the Baltic Sea an uneasy feeling of déjà vu**.	DELETE ADD REPLACE
	Printed products have continued a downward spiral **with the surge of digital technology**.	Printed products have continued their downward spiral **as improvements in technology allow for more materials to be digitized**.	DELETE ADD REPLACE
SIMPLE REWRITE	~~And what's more,~~ **the** dust seems to cover the whole planet.	**This** dust seems to cover the whole planet.	DELETE REPLACE
	But he hopes he**'ll** find a big one.	But he hopes he **will** find a big one.	REPLACE

examination, we observed that the main source of errors was sentences manually labelled as containing STRONG REWRITE and SIMPLE REWRITE operations. As shown in Table 3.6, since these operations are not explicitly identified by our algorithms, sentences that contain them are labelled as a mix of DELETE, ADD and REPLACE. In order to reduce this effect in the results, we removed all sentences that contained STRONG REWRITEs, and re-annotated those with SIMPLE REWRITEs with corresponding DELETE, ADD and/or REPLACE labels. The new performance scores are shown under "Without REWRITEs" in Table 3.5. We observe that precision scores improve for all operations of interest, with small reductions in recall.

To summarise, our intrinsic evaluation experiments have shown that our annotation algorithms are capable of identifying the simplification operations performed in given pairs of original-simplified sentences. By exploiting word alignments (especially those extracted with SimAlign), the algorithms are best at annotating DELETE, REPLACE, ADD and SPLIT operations at word and sentence levels, accordingly. However, our algorithms demonstrated sensitivity to word misalignments, particularly affecting the identification of REORDER operations. Furthermore, sentences with SIMPLE REWRITEs and STRONG REWRITEs proved challenging, unless their annotations can be reduced to more fine-grained operations, such as DELETE, ADD and REPLACE. Finally, the annotation algorithms that identify operations at word-level and phrase-level using word alignments from MWA were included in MASSAlign (Paetzold et al., 2017), a software package that also provides the vicinity-driven sentence alignment methods of (Paetzold and Specia, 2016c). All the annotation algorithms using either MWA or SimAlign have also been included in EASSE (Alva-Manchego et al., 2019a), detailed in Sec. 3.3.1.


Original
Simplified
Annotation Algorithms
"Silver" Annotated Corpus
Training a Label Predictor
Simplification Operations Labeler
Hershey left no heirs when he died in 1945, giving most of his fortune to charity.
Original Sentence
Simplification Operations Labeler
Hershey left no heirs when he died in 1945, giving most of his fortune to charity.
delete
replace
Sentence with Annotated Operations
Models to execute each simplification operation
Hersey died in 1945 and gave most of his fortune to charity .
Simplified Sentence

(a) Generation of annotated data to train a classifier to predict the simplification operations that need to be performed to a given original sentence.

(b) At test time, the trained classifier annotates the given original sentences with word-level simplification operation labels, and dedicated models execute their particular operations.

Figure 3.5 Pipeline architecture that models Sentence Simplification as a Sequence Labelling problem.

3.2.3 Extrinsic Evaluation

In (Alva-Manchego et al., 2017), we explored using automatic annotations of simplification operations to assist in generating automatic simplifications. We modelled Sentence Simplification as a Sequence Labelling problem, proposing a pipeline architecture for the task (Figure 3.5). In general terms, we used distant supervision to train a classifier to predict word-level simplification operations that are later executed individually by dedicated models.[7]

Generation of Annotated Data

We used the word-level annotation algorithms presented previously to generate automatically-annotated "silver" data (in contrast to human-annotated "gold" data) that can be used to train a simplification operation predictor (Figure 3.5a). In particular, we used the algorithms that identify DELETE and REPLACE using word alignments from MWA.[8] We focused on these two

[7] The Thesis author contributed to this paper by generating the annotated data, computing the automatic metrics, analysing and discussing results from the experiments, and as a judge for manual evaluation. The other activities were performed by collaborators, and are included here for completeness.

[8] At the time of these experiments, SimAlign was unavailable.

operations since we lacked models capable of applying the others. For example, ADD would require access to external resources that allow obtaining (or inferring) the necessary information to be added to the sentences (e.g. explanations of unknown concepts). We annotated all 1-to-1 automatic sentences alignments from the Newsela corpus, extracted as described in Sec. 3.1.1. We focused on this subset of the dataset since we were not interested in sentence pairs where no changes were made (thus discarding "identical" alignments), nor in simplifications involving structural modifications (thus discarding 1-to-N and N-to-1 alignments) since we did not implement a module for sentence splitting or merging. The automatically-annotated data was randomly split into training (80%), development (10%) and test (10%) sets, and normalised for entities (including names, locations, numbers).

Training a Simplification Operations Predictor

A Bidirectional Recurrent Neural Network with Long Short Term Memory units (Hochreiter and Schmidhuber, 1997) was trained to predict the simplification operation that needed to be performed in each word in a given input sentence.[9] Overall, the classifier tends to over-predict the majority class (i.e. to copy a word), turning its proposed simplification operations rather conservative. DELETE is the simplification operation with the best performance: 0.76 of precision and 0.90 of recall. REPLACE also has a relatively high precision (0.70), but low recall (0.39). As such, executing these operations correctly would result in simplifications that favour compression over lexical paraphrasing. However, as will be shown later, when coupled with a state-of-the-art lexical simplifier, in particular, the generated simplifications are "simpler" than those produced by strong baselines, both in terms of automatic and human evaluation.

Execution of Simplification Operations

At test time, a given original sentence goes through the pipeline in Figure 3.5b in order to simplify it. First, the simplification operation labeller predicts, for each token in the sentence, whether it needs to be deleted, replaced or copied (i.e. left as-is). Then, dedicated models are in charge of applying each word-level operation. In our implementation, executing DELETE simply omits the respective word in the output sentence. For REPLACE, we use the supervised Lexical Simplification approach of (Paetzold and Specia, 2017a). First, their simplifier generates candidate substitutions for each word labelled with REPLACE using data from Newsela and retrofitted context-aware word embedding models, and selects the ones that fit the context of the target word through an unsupervised boundary ranking approach (Paetzold and Specia, 2016b).

[9] Architecture and training details can be found in (Alva-Manchego et al., 2017).

Then, it ranks candidates using a supervised neural ranking model trained over manually annotated simplifications, and selects the best following a final confidence check that ensures that the highest-ranked candidate is suitable within context.

Automatic Evaluation

Our proposed approach (SeqLab) is compared against three MT-based models trained using our same splits of the Newsela dataset: Moses (Koehn et al., 2007), Nematus (Sennrich et al., 2017), and NTS (Nisioi et al., 2017). In addition, we used the "silver" labels in a semi-oracle trial where we apply the actual simplification operations as given in the annotated corpus. In other words, we simply take the automatic labels as true and use the alignments between original and simplified words to apply the actual operations. For our automatic evaluation, we computed two commonly-used metrics in MT: BLEU (Papineni et al., 2002) and TER (Snover et al., 2006) between the hypotheses (i.e. the generated automatic simplifications) and the references, and between the hypotheses and the original sentences. This allows us to analyse not just the quality of the output when compared to simplification references, but also how different the automatic simplifications are from the original input. In addition, we also computed the simplification-specific metric SARI (Xu et al., 2016).[10] Results are shown in Table 3.7.

Table 3.7 Performance of MT-based and operation-based simplification models. Metrics are BLEU and TER between simplified version (Hyp) and reference (Ref) or original version (Orig), the percentage of sentences copied from the input (%Same), and SARI for the simplifications.

	Hyp vs. Ref			Hyp vs. Orig		
System	BLEU↑	TER↓	SARI↑	BLEU↓	TER↑	%Same↓
Semi-Oracle	67.33	22.66	61.71	61.63	26.01	10.83
Moses	**57.79**	**40.19**	24.58	98.30	0.86	89.50
Nematus	46.90	52.84	29.89	76.29	20.10	30.45
NTS	53.79	45.24	30.44	77.63	16.70	42.76
SeqLab	41.37	48.72	**31.29**	**59.71**	**25.24**	**14.06**

The semi-oracle model produces more accurate and less conservative simplifications than all MT-based approaches. When comparing Hyp and Ref, it achieved the highest SARI and BLEU scores with the lowest TER, indicating that the generated simplifications are comparable to the human references. When comparing Hyp and Orig, it has the lowest rate of copied input sentences among all systems, indicating that its simplifications are not conservative.

[10] We computed SARI at sentence-level with the original python script from https://github.com/cocoxu/simplification, and averaged the scores from all sentences for a corpus-level score.

These results indicate that if we were able to predict the word-level operation labels with high accuracy, and also implement simplification modules capable of applying them properly, we could produce high-quality simplifications. Indeed, even though SeqLab only applies two operations (REPLACE and DELETE), and the operation predictor's performance is imperfect, our model achieved the highest SARI score in the test set when compared to all MT-based models, as well as producing less conservative outputs (best scores in Hyp vs. Orig).

Human Evaluation

A manual assessment of the generated simplifications was also conducted to validate the findings of the automatic metrics. We sampled 100 original sentences of the test set and asked four proficient English speakers to rate the simplifications produced by the MT-based models and SeqLab, as well as the simplified human reference. Each judge was presented with the original sentence and all its simplifications in random order, and was asked to use 5-points Likert scales to qualify the grammaticality, meaning preservation and simplicity of the simplifications. For simplicity, in particular, judges were asked to assess the "improvements in simplicity" over the original sentence. Therefore, a simplification where no changes were made received a score of 1. Results are shown in Table 3.8.

Table 3.8 Average scores and standard deviation for human judgments on grammaticality, meaning preservation and simplicity for the simplifications evaluated.

	Grammaticality	Meaning	Simplicity
Reference	5.00 ± 0.0	4.45 ± 0.9	2.70 ± 1.3
Moses	**4.98 ± 0.2**	**4.99 ± 0.1**	1.14 ± 0.4
Nematus	4.49 ± 0.9	3.99 ± 1.2	1.46 ± 0.9
NTS	4.75 ± 0.6	4.08 ± 1.3	1.53 ± 1.0
SeqLab	4.16 ± 1.0	3.91 ± 1.1	**1.66 ± 0.9**

SeqLab achieved the highest Simplicity score of all automatic systems. However, its less-conservative simplifications affect its Meaning Preservation judgements. In order to improve the grammatically of the output, it could be possible to incorporate a post-processing model that ensures the fluency of the generated simplification. However, this was out of the scope of our study and was left for future work. Overall, all automatic systems obtained Simplicity scores significantly lower than the Reference, signalling that standard sequence-to-sequence MT-based models are insufficient to generate high-quality simplifications. Finally, the fact that Reference sentences did not achieve very high Simplicity scores could be caused by the type of simplifications selected for the study: 1-to-1 alignments with no structural simplifications.

To summarise, our extrinsic evaluation experiments have shown that our algorithms for annotating simplification operations could be used in the implementation of automatic simplification systems. In (Alva-Manchego et al., 2017), we proposed a sequence labelling approach for Sentence Simplification based on distant supervision. In particular, the semi-oracle experiment showed that executing the operations suggested by the automatically-produced labels can in fact simplify the original input sentence. In addition, the operation labels helped the proposed pipeline model to generate simplifications that are "simpler" and less conservative than those produced by standard MT-based models. These results evidence the quality of the simplification operations identified by our algorithms, and their usefulness beyond just the analysis of original-simplified sentence pairs.

3.3 Error Analysis based on Correctness of Operations

As explained in Chapter 2, to automatically evaluate the output of a Sentence Simplification model, researchers normally use metrics from MT (e.g. BLEU (Papineni et al., 2002)), readability metrics (e.g. Flesch Kincaid (Kincaid et al., 1975)) and simplicity metrics (e.g. SARI (Xu et al., 2016)). Whilst automatic metrics are easy to compute, they only provide overall performance scores without giving insights about specific strengths and weaknesses of a simplification approach. Therefore, we propose to exploit the word-level annotation algorithms to further analyse the performance of simplification models based on how effective they are at executing specific simplification operations. We show how this per-operation error analysis contributes to a better understanding of the scores provided by automatic evaluation metrics, and to an improved comparison between different simplification models.

3.3.1 EASSE

In order to compare text generation models objectively, it is important to use the same test corpora and (implementations of) metrics that evaluate their performance. As evidenced in Chapter 2, this is not always the case for Sentence Simplification systems. Most evaluation 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., 2018b)) suffer from insufficient documentation, or require executing multiple scripts with hard-coded paths, which prevents researchers from using them.

In order to tackle this issue, in (Alva-Manchego et al., 2019a) we introduced EASSE (**E**asier **A**utomatic **S**entence **S**implification **E**valuation), 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.[11] With this tool, we made the following contributions: (1) we provide popular automatic metrics in a single software package, (2) we supplement these metrics with word-level operation analysis (Sec. 3.2) and reference-less Quality Estimation (QE) features, (3) we provide straightforward access to commonly used evaluation datasets, and (4) we generate a comprehensive HTML report for quantitative and qualitative evaluation of a Sentence Simplification system.[12]

Automatic Evaluation Metrics

Although human judgements on grammaticality, meaning preservation and simplicity are considered the most reliable method for evaluating a SS 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 wraps or re-implements the most common evaluation metrics in Sentence Simplification:

BLEU. EASSE wraps up the implemenation available in SacreBleu (Post, 2018)[13]. 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 output and reference sentences.

SARI. EASSE re-implements SARI's corpus-level version in Python (it was originally available in Java). In this version, for each operation ($ope \in \{add, del, keep\}$) and n-gram order, precision $p_{ope}(n)$, recall $r_{ope}(n)$ and F1 $f_{ope}(n)$ scores are calculated. These are then averaged over the n-gram order to get the overall operation F1 score F_{ope}:

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

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

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

[12] The thesis author contributed with most of the functionalities in EASSE, except for the quality estimation features and the HTML report, which are included here for completeness.

[13] https://github.com/mjpost/sacreBLEU

Xu et al. (2016) indicate that only precision should be considered for the deletion operation, but we followed the Java implementation that uses F1 score for all operations in corpus-level SARI. We also provide a sentence-level implementation equivalent to the original Python one[14].

SAMSA. EASSE re-factors the original SAMSA implementation[15] 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)[16] through their official Python interface; and (3) a single function call to get sentence-level SAMSA scores instead of running a series of scripts.

FKGL. EASSE re-implements FKGL by porting publicly available scripts[17] to Python 3 and fixing some edge case inconsistencies (e.g. newlines incorrectly counted as words or bugs with memoization).

Word-level Operation Analysis

EASSE includes algorithms to determine which simplification operations an automatic system performs more effectively. This is done based on the word-level annotation algorithms introduced in Sec. 3.2 for DELETE, REPLACE and REORDER. We also introduced the COPY label for all words in the original sentence that are aligned to one in the simplification, but were not affected by the aforementioned operations. We generate two sets of automatic operation labels: (1) between the original sentences and their reference simplifications, and (2) between the original sentences and their automatic simplifications produced by the system being evaluated. Considering (1) as reference labels, we calculate the F1 score of each operation in (2) to estimate their correctness. When more than one reference simplification exists, we calculate the per-operation F1 scores of the output against each reference, and then keep the highest one as the sentence-level score. The corpus-level score is the average of sentence-level scores.

Quality Estimation Features

EASSE uses Quality Estimation (QE) features from Martin et al. (2018)'s open-source repository[18] to provide additional information on specific behaviours of simplification systems that

[14] https://github.com/cocoxu/simplification/blob/master/SARI.py

[15] https://github.com/eliorsulem/SAMSA

[16] https://stanfordnlp.github.io/stanfordnlp/corenlp_client.html

[17] https://github.com/mmautner/readability

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

are not reflected in metrics. The 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[19]. Some of these features are further described in Sec. 4.3.2.

Access to Test Datasets

EASSE provides access to three publicly available datasets for automatic Sentence Simplification evaluation (Table 3.9): PWKP (Zhu et al., 2010), TurkCorpus (Xu et al., 2016), and HSplit (Sulem et al., 2018a). All of them consist of original sentences extracted from English Wikipedia articles, and either simplifications from Simple English Wikipedia or produced through crowdsourcing. EASSE can also evaluate systems' outputs in user-provided datasets.

Table 3.9 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.

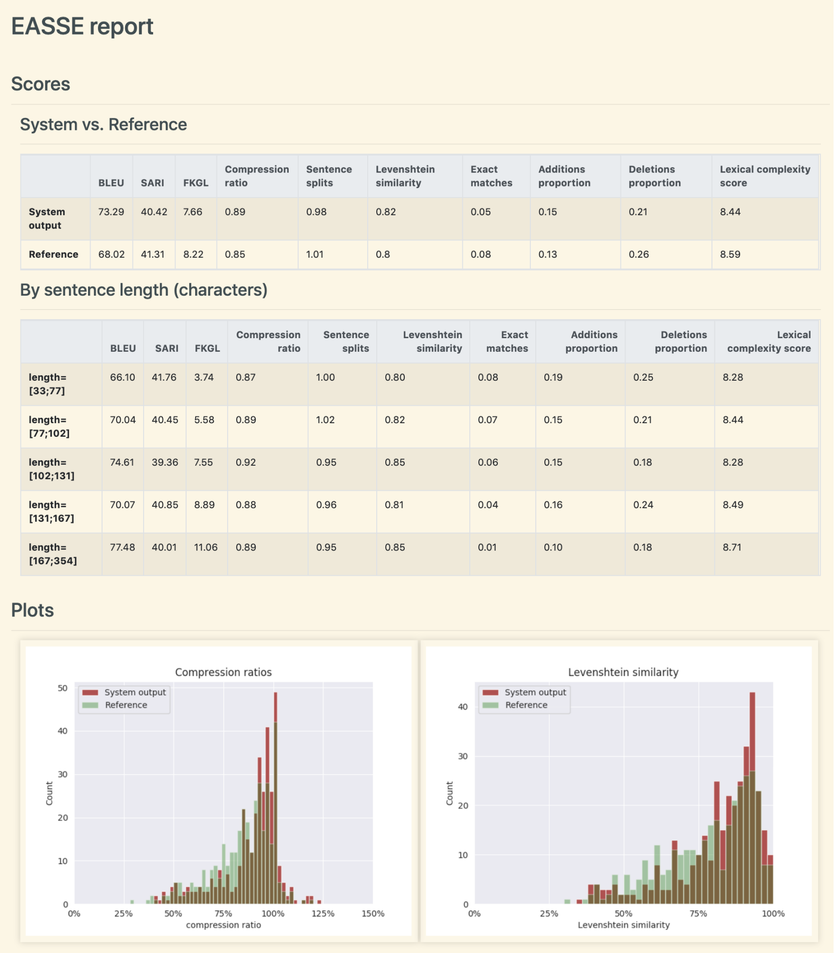
Test Dataset	Instances	Alignment Type	References
PWKP	93	1-to-1	1
	7	1-to-N	1
TurkCorpus	359	1-to-1	8
HSplit	359	1-to-N	4

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 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 original sentence length, to get insights on how the model handles short original sentences versus long original sentences, e.g. *does the model keep short sentences unmodified more often than long sentences?* This report further facilitates qualitative analysis of system outputs by displaying original 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

[19] 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.

behaviours of the systems. Instances that are sampled include simplifications with sentence splitting, simplifications that significantly modify the original sentence, output sentences with a high compression rate, those that display lexical simplifications, among others. Each of these aspects is illustrated with 10 instances. Figure 3.6 shows the first part of an example report.[20]



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 3.6 An example of a performance report generated by EASSE.

3.3.2 Benchmarking Sentence Simplification Systems

As part of our survey in (Alva-Manchego et al., 2020b), we used the functionalities and resources available in EASSE to compare representative Sentence Simplification systems that had publicly-available outputs for PWKP or TurkCorpus at the time of our study. The 1-to-N alignments in PWKP mean that some instances of sentence splitting are present in the dataset. This is a limitation of TurkCorpus that only contains 1-to-1 alignments, mostly providing

[20] A full report can be seen in`https://github.com/feralvam/easse/blob/master/demo/report.html`

instances of lexical paraphrasing and deletion operations. On the other hand, each original sentence in TurkCorpus has eight simplified references produced through crowdsourcing. This allows us to more confidently use metrics that rely on multiple references. We first compared the models' outputs using automatic metrics to obtain an overall measure of simplification quality, in particular: BLEU, SARI, SAMSA and Flesch-Kincaid Grade Level (FKGL). We were also interested in an in-depth study of the simplification capabilities of each model. As such, we used the word-level operation analysis to determine which simplification operations each model performs more effectively.

For a fair comparison, we detokenised and recased all original sentences and system outputs using the pre-processing scripts from Moses.[21] Then, we used EASSE (Sec. 3.3.1) to compute all metrics with the same configuration: tokenisation using SacreMoses[22] and case-sensitive calculation. For the word-level analysis, we obtained word alignments using SimAlign with the argmax algorithm (the best configuration according to Sec. 3.2.2).

Comparing Models in the PWKP Test Set

For the PWKP test set, models that have publicly-available outputs are: Moses (released by Zhu et al.), PBSMT-R, QG+ILP (released by Narayan and Gardent), Hybrid, TSM, UNSUP, EncDecA, DRESS, DRESS-LS and EditNTS (Dong et al., 2019).[23] Overall scores using standard metrics are shown in Table 3.10 sorted by SARI.

Table 3.10 Performance measured with automatic metrics in the PWKP test set.

Model	SARI↑	BLEU↑	SAMSA↑	FKGL↓
Reference	100.00	100.00	27.46	9.74
Hybrid	**49.76**	43.56	36.08	10.30
Moses	45.92	**46.06**	36.16	11.61
DRESS-LS	39.81	35.84	30.59	8.54
DRESS	39.35	34.06	30.25	8.40
UNSUP	37.90	36.34	37.62	7.77
TSM	37.39	30.07	38.56	**6.41**
EditNTS	37.70	18.73	24.76	6.57
PBSMT-R	34.78	44.68	37.03	12.26
QG+ILP	34.77	39.90	**40.84**	7.11
EncDecA	31.88	46.52	37.46	12.13

[21] https://github.com/moses-smt/mosesdecoder/tree/master/scripts
[22] https://github.com/alvations/sacremoses
[23] These systems were described in detail in Chapter 2.

According to the values of the automatic metrics, Hybrid is the model that produces the simplest output as measured by SARI, followed by Moses. If we consider BLEU as indicative of grammaticality, Moses produces the most fluent output, followed closely by Hybrid. This is not surprising since MT-based models, in general, tend to produce well-formed sentences. Also, TSM achieves the lowest FKGL, which seems to be indicative of shorter output, rather than simpler output (its SARI score is in the middle of the pack). We also note that grammaticality has no impact on FKGL values; that is, a text with low grammaticality can still have a good FKGL score. Therefore, since EditNTS obtains the lowest BLEU score, its FKGL value is not reliable. In terms of SAMSA, the best model is QG+ILP, followed by TSM. Since this metric focuses on assessing sentence splitting, it is expected that the models that explicitly perform this operation scored best. From these results, we could conclude that Hybrid is the overall best model, since it achieves the highest SARI score, BLEU and SAMSA scores close to the highest, and a FKGL score not too far from the reference.

It could be surprising to see that the SAMSA score for the Reference is one of the lowest. This is explained by the way the metric is computed. Since it relies on word alignments, if the simplification significantly changes the original sentence, then these alignments are not possible, resulting in a low score. For example:

(5) **Original:** Genetic engineering has expanded the genes available to breeders to utilize in creating desired germlines for new crops .

(6) **Reference:** New plants were created with genetic engineering .

This is the reason why SAMSA should only be used for evaluating instances of sentence splitting that do not perform significant rephrasings of the sentence.

Table 3.11 presents the results of our operation-based performance measures for models' outputs on PWKP. From Table 3.10, we saw that Hybrid got the highest SARI score. With the results on Table 3.11 we can understand better why that happened. Since SARI is a metric aimed mostly at lexical simplification, it rewards replacements and making small changes to the original sentence. In this test set, Moses and Hybrid's REPLACE and COPY operations are amongst the most accurate. EncDecA, which got the worst SARI score, has a very low REPLACE performance, and has the lowest DELETE accuracy, which points out to them mostly repeating the original sentence without much modifications. Finally, EditNTS is the best one in deleting content, which explains its low (and "good") FKGL score. Overall, most systems are good at preserving content but, even though REPLACE is an important simplification operation, all of them struggle with performing it correctly.

Table 3.11 Performance measured with operation-specific F1 score in the PWKP test set.

Model	DELETE	REORDER	REPLACE	COPY
Hybrid	26.99	17.95	**32.95**	26.99
Moses	27.95	17.96	24.62	54.43
DRESS-LS	29.65	12.88	4.27	54.23
DRESS	29.87	12.68	4.43	53.46
UNSUP	19.44	10.89	14.60	45.34
TSM	20.40	22.67	16.57	47.18
EditNTS	**37.42**	**23.25**	5.59	34.62
PBSMT-R	6.64	5.96	15.95	**54.84**
QG+ILP	13.22	6.27	18.30	38.96
EncDecA	7.42	3.85	9.69	53.12

Comparing Models in the TurkCorpus Test Set

The models evaluated on this test set are almost the same ones as before, except for Moses, QG+ILP, TSM and UNSUP for which we could not find available outputs on this test set. However, we now also include SBSMT (PPDB+SARI), NTS+SARI and UNTS+10K. Since TurkCorpus is multi-reference, we use all 8 of them for measuring BLEU and SARI. For calculating Reference values for these two metrics and FKGL, we randomly sampled one of the 8 human references for each instance and computed the scores using the other 7 as references. We also calculated SAMSA for each of the four manual simplifications in HSplit, and chose the highest as an upper-bound. Results are presented in Table 3.12 sorted by SARI score.

Table 3.12 Performance measured with automatic metrics in the TurkCorpus test set.

Model	SARI↑	BLEU↑	SAMSA↑	FKGL↓
Reference	40.21	70.79	54.88	9.13
ACCESS	**41.05**	75.56	42.36	7.62
DMASS-DCSS	39.67	72.26	36.74	8.09
SBSMT(PPDB+SARI)	39.13	71.98	41.60	8.36
PBSMT-R	37.83	81.91	46.08	9.27
EditNTS	37.35	**85.84**	46.68	8.56
DRESS-LS	36.81	80.59	44.69	7.78
DRESS	36.66	77.53	43.24	7.65
UNTS+10K	36.03	66.88	**47.28**	8.35
NTS+SARI	35.96	83.10	45.49	8.32
Hybrid	31.54	49.53	45.31	**5.28**

In this test set, ACCESS scored the highest SARI, followed by the models that used external knowledge from (Simple) PPDB: DMASS-DCSS and SBSMT (PPDB+SARI). The SARI scores of those models are also very close to that of the random Reference. The most fluent model according to BLEU is EditNTS, closely followed by NTS+SARI. This could be due to (standard) neural sequence-to-sequence models being capable of generating grammatical output of high quality. Contrary to what was observed in the PWKP test set, Hybrid achieved the worst score in SARI. This could be explained by the low FKGL score. It appears that Hybrid tends to modify the original sentence much more than other models, potentially by deleting content and producing shorter outputs. In this test set, this behaviour is penalised since most references were only rewritten considering paraphrases of words and phrases. In terms of SAMSA, UNTS+10K scored the highest, which could be explained by the model being trained also using instances from split-and-rephrase datasets, specifically (Narayan et al., 2017).

Table 3.13 presents the results of our operation-based performance measures for models' outputs on the TurkCorpus test set. Once again, they complement the scores of the automatic metrics. ACCESS, DMASS-DCSS and SBSMT (PPDB+SARI) are the best at performing REPLACE, explaining their high SARI scores. However, they differ in their effectiveness at performing DELETE, with ACCESS being much superior. Since SARI also evaluates precision of deletions, this explains why ACCESS was the winner in this metric. EditNTS got the best BLEU score, which is explained by its high score in COPY. Even though Hybrid achieved the lowest FKGL value, it is the best in DELETE, has the lowest COPY, and produces extremely inaccurate replacements. This, again, suggests that a low FKGL score does not necessarily indicate good simplifications. Finally, the origin of the TurkCorpus set itself could explain some of these results. According to Xu et al. (2016), the participants performing the simplifications were instructed to mostly produce replacements with virtually no deletions. As such, copying is a significant operation and, therefore, models that are good at performing it reflect better the characteristics of the human simplifications in this dataset.

3.4 Summary and Final Remarks

In this Chapter, we attempted to answer the following research question: **Which simplification operations do automatic systems perform accurately?**

We started by conducting a **manual study** on automatically-aligned original-simplified sentence pairs from the Newsela corpus, a dataset that contains simplifications produced by professional editors. In that analysis, we determined that the most common sentence-level simplification operations are related to lexical paraphrasing (REPLACE and SIMPLE REWRITE),

Table 3.13 Performance measured with operation-specific F1 score in the TurkCorpus test set.

Model	DELETE	REORDER	REPLACE	COPY
ACCESS	34.91	23.09	52.49	87.54
DMASS-DCSS	25.58	18.85	52.47	81.12
SBSMT(PPDB+SARI)	11.34	7.60	**52.82**	**91.36**
PBSMT-R	22.67	17.61	42.38	91.15
EditNTS	35.10	**25.09**	21.80	90.27
DRESS-LS	34.22	22.93	18.66	86.69
DRESS	35.26	23.84	21.19	85.84
UNTS+10K	22.65	13.18	29.60	90.66
NTS+SARI	19.49	20.92	32.03	82.88
Hybrid	**44.75**	21.26	2.37	69.18

compression (DELETE), addition of information (ADD), and sentence splitting (SPLIT). The high frequency of DELETEs, in particular, is related to the target audience of the Newsela corpus: children in different grade levels. The editors considered acceptable to lose some information from the original texts in order to simplify them for this specific group of readers. This contrasts with other datasets, such as SimPA (Scarton et al., 2018a), that only contain lexical and syntactic simplifications, since their target audience (users of websites from Public Administration) require access to all the information available.

Based on our previous findings, we implemented **novel annotation algorithms** that identify the most common simplification operations, namely: DELETE, REPLACE, ADD, REORDER, and SPLIT, using labels at word, phrase and sentence levels, accordingly. We first validated that these automatic annotations have relatively-high accuracy through an intrinsic evaluation that allowed us to: (1) identify which word aligner (the core component of the algorithms) benefits the annotations the most (i.e. SimAlign); and (2) which type of simplification instances are more problematic for the algorithms to annotate (e.g. those with strong rewritings). Furthermore, the automatic word-level labels also demonstrated their usefulness in developing automatic Sentence Simplification systems. In particular, they were a paramount component in the implementation of SeqLab, a **system that models simplification as a sequence labelling problem**. We showed that by explicitly indicating whether a token in a given original sentence needed to be deleted, replaced or copied, we could generate better simplifications than standard MT-based architectures. This idea later motivated the development of EditNTS (Dong et al., 2019), where the simplification model internally predicts the word-level operation labels and is trained in end-to-end, instead of in a pipeline architecture as we originally designed.

Finally, we used the word-level annotations generated by our algorithms to carry out an operation-based error analysis of the performance of several Sentence Simplification systems.

Identification of Simplification Operations

First, we implemented **EASSE, a software package for standardising automatic evaluation** in simplification. EASSE provides easy access to public test datastes, standard automatic metrics, detailed word-level operation error analysis, quality estimation features, and a comprehensive performance report. Using these functionalities, we elaborated a **benchmark of different approaches to Sentence Simplification**. We compared several models using standard overall-performance metrics, and proposed a new operation-specific method for qualifying the simplification operations that each model performs. The former analysis provided us with insights about the simplification capabilities of each system, which help better explain the initial automatic scores. Most importantly, it allowed us to suggest an answer to our research question: Current Sentence Simplification models are good at preserving content from the original input sentences. However, this is due to their mostly-conservative nature. We still need to provide them with the ability to accurately determine what to copy and what to delete. In addition, although progress has been made in recent years, replacing complex words with simpler synonyms is still challenging.

Chapter 4

ASSET: A New Evaluation Dataset

In the previous Chapter, we explained how several rewriting operations can be performed to simplify sentences, such as lexical paraphrasing (i.e. REPLACE), compression (i.e. DELETE) and sentence splitting (i.e. SPLIT). Despite this variety, models for automatic Sentence Simplification are evaluated using two types of datasets: (1) multi-reference and single-operation, such as TurkCorpus and HSplit; or (2) single-reference and multi-operation, such as PWKP and Newsela. Also, while reference simplifications in (1) were produced manually, those in (2) are generated through (imperfect) automatic alignments. This makes it difficult to reliably assess the ability of simplification models in more realistic *abstractive* scenarios, i.e. cases with substantial modifications to the original sentences. In this Chapter, we attempt to answer the following research question: **Do we need multi-reference multi-operation evaluation datasets to support better assessment of Sentence Simplification models?** In (Alva-Manchego et al., 2020a), we introduced ASSET (**A**bstractive **S**entence **S**implification **E**valuation and **T**uning), a new multi-reference dataset for tuning and evaluation of Sentence Simplification models, where each manual reference was produced through several simplification operations.

First, we motivate the development of ASSET by highlighting the limitations of current evaluation data, and then justify collecting simplification references via crowdsourcing (Sec. 4.1). Then, we detail our methodologies for data collection and quality control (Sec. 4.2). After that, through quantitative (Sec. 4.3) and qualitative (Sec. 4.4) experiments, we show that simplifications in ASSET are better at capturing characteristics of simplicity when compared to other standard evaluation datasets for the task. Furthermore, we motivate the development of better methods for automatic evaluation using ASSET, since we show that current popular metrics may not be suitable when multiple simplification operations are performed (Sec. 4.5). Finally, we offer some remarks about the usefulness and potential of ASSET for better evaluation of Sentence Simplification systems in multi-operation scenarios (Sec. 4.6).

4.1 Motivation

As presented in Sec. 2.1.1, studies on how humans simplify have shown that the scope of simplification operations 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. However, even though most models are trained on datasets displaying several simplification operations (e.g. WikiLarge (Zhang and Lapata, 2017)), we currently do not measure a system's performance in scenarios where substantial modifications to the original sentences have been performed. In this Section, we motivate the development of an evaluation dataset with multi-operation reference simplifications by pointing out the limitations of current evaluation data. In addition, we give grounds for a data collection methodology based on crowdsourcing.

4.1.1 Evaluation Data for Sentence Simplification

As detailed in Sec. 2.2, most datasets for Sentence Simplification (Coster and Kauchak, 2011b; Hwang et al., 2015; Zhu et al., 2010) consist of automatic sentence alignments between related articles in English Wikipedia (EW) and Simple English Wikipedia (SEW). Unfortunately, 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 in PWKP (Zhu et al., 2010) 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. Whilst the Newsela corpus (Xu et al., 2015) contains simplifications produced by professionals applying multiple simplification operations, original-simplified sentence alignments are automatically computed and thus imperfect. In addition, its data can only be accessed after signing a restrictive public-sharing licence and cannot be redistributed, hampering reproducibility.

Evaluating models on automatically-aligned sentences is problematic since references can be noisy. For example, a 1-to-1 alignment can be identified instead of a 1-to-N, and thus not signalling that a splitting operation was performed. This is specially a disadvantage of the PWKP and Newsela datasets that only provide one reference simplification per original sentence. 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. On the other hand, while also providing multiple (4) human references, HSplit (Sulem et al., 2018a) contains simplifications involving only splitting for the 359 sentences in the test set of TurkCorpus.

The presented scenario prevents evaluating a model's ability to perform a more diverse set of rewriting transformations when simplifying sentences. For ASSET, we built on TurkCorpus and HSplit by collecting a dataset that provides multiple manually-produced simplification references involving multiple types of simplification operations simultaneously.

4.1.2 Crowdsourcing Manual Simplifications

Crowdsourcing has mostly been used for creating datasets for Lexical Simplification (Horn et al., 2014; Shardlow et al., 2020; Yimam et al., 2017a,b), or to elicit human judgements on the quality of automatic simplifications (Jiang et al., 2020; Kriz et al., 2019; Zhang and Lapata, 2017), manual simplifications (Lasecki et al., 2015) and human perception of complexity (Brunato et al., 2018). However, a few projects have crowdsourced manual simplification references.

Pellow and Eskenazi (2014a) built a corpus of everyday documents (e.g. driving test preparation materials), and analysed the feasibly of crowdsourcing their manual simplifications. In order to control the quality of the data, workers went through a training session that had two purposes: (1) to block spammers that submitted noisy work; and (2) to filter our workers without the necessary 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, 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 a manual check of their first several submissions. More recently, Scarton et al. (2018a) used volunteers to collect simplifications for SimPA, a dataset with sentences from the Public Administration domain (e.g. websites that describe services, citizen rights and duties). One particular characteristic of their methodology is that lexical and syntactic simplifications were performed independently.

Due to the reported success stories related to crowdsourcing simplifications of sentences, we used this methodology for creating manual references in ASSET. In particular, we tried to incorporate the majority of their best practices for quality control, so as to ensure that the collected simplifications provide a reliable resource for evaluation of automatic systems.

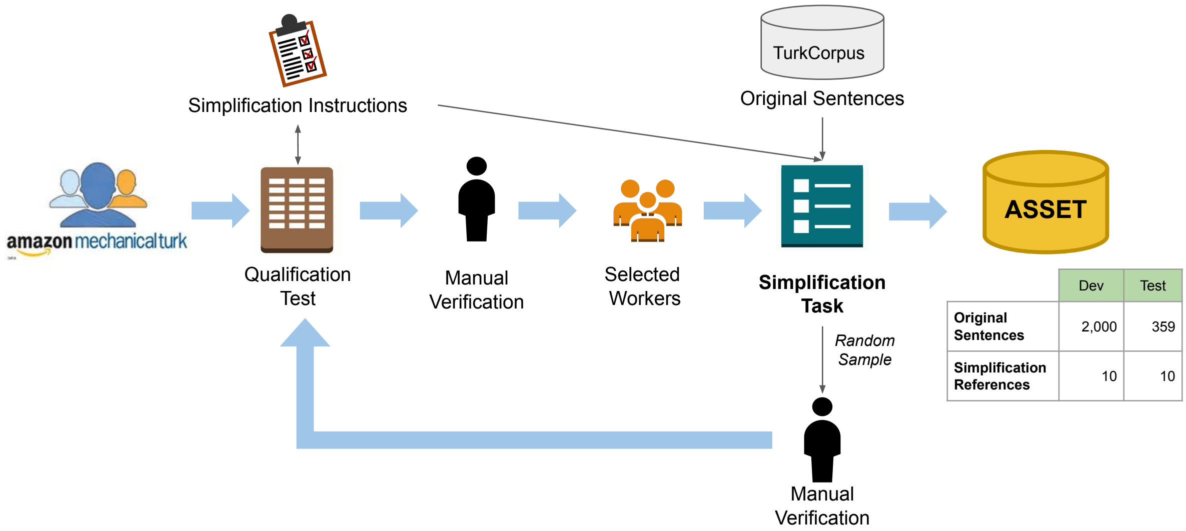
4.1.3 Direct Assessment

Direct Assessment (DA, Graham et al., 2013, 2017) is a methodology for crowdsourcing reliable human evaluations on the quality of machine translations. Since having experts judge MT quality can be expensive, it is convenient to exploit the benefits of crowdsourcing but maintaining confidence on the judgements collected. DA accounts for this by implementing quality control mechanisms, such as: including human reference translations that should receive high scores, including bad translations that should receive low scores, and repeating evaluation instances to measure annotator consistency. In addition, DA moved away from the established practice of using ordinal rankings via relative preference judgements (Bojar et al., 2017a, 2016a, 2015), and proposed to collect direct estimates of quality using a continuous scale (1-100). In this scenario, a judge is shown only one translation at a time and asked to submit their level of agreement with a prompt, normally regarding the adequacy of the translation w.r.t a reference. Using continuous values for evaluation provides the additional benefit of allowing statistical analyses and aggregation of the scores, such as normalising the assessments of each judge by their own mean and standard deviation to reduce personal biases. With all these benefits, DA has turned in the de-facto methodology for human evaluation in the WMT Shared Tasks for Machine Translation in recent years (Barrault et al., 2020, 2019; Bojar et al., 2018).

In this Thesis, we are the first to use DA for collecting human judgements of Sentence Simplification quality. We believe it is a proven methodology for evaluating generation tasks, and it is particularly suitable for our multi-operation simplification scenario where we aim to assess quality of automatic outputs in a general way and not particular to any specific operation.

4.2 Creating ASSET

We extended TurkCorpus by using the same original sentences, but crowdsourced manual references that encompass a richer set of simplification operations. Since TurkCorpus was adopted as the standard dataset for evaluating Sentence Simplification models, several system outputs on this dataset are already publicly available (Martin et al., 2020; Zhang and Lapata, 2017; Zhao et al., 2018). Therefore, we can now assess the capabilities of these and other systems



	Dev	Test
Original Sentences	2,000	359
Simplification References	10	10

Figure 4.1 Overview of the crowdsourcing methodology followed to collect manual reference simplifications for ASSET (development and test sets) and ensuring their quality.

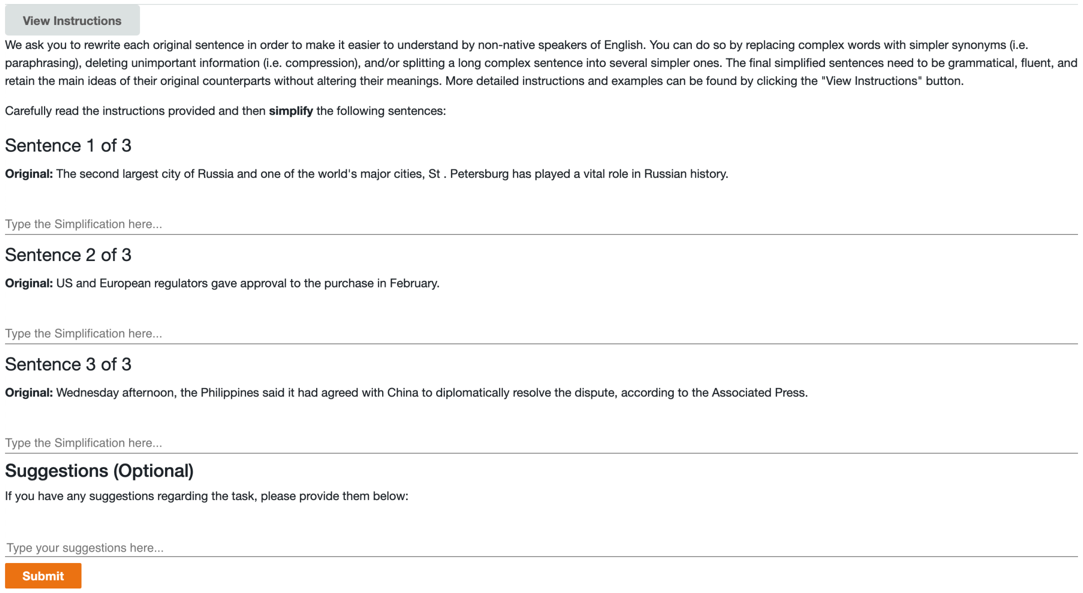
in scenarios with varying simplification expectations: lexical paraphrasing with TurkCorpus, sentence splitting with HSplit, and multiple operations with ASSET.

4.2.1 Data Collection Protocol

Manual simplifications were collected using Amazon Mechanical Turk (AMT). AMT allows to publish HITs (Human Intelligence Tasks) that workers can choose to work on, submit an answer, and collect a reward if the work is approved. This platform was also used for creating TurkCorpus. Figure 4.1 depicts the general methodology followed for collecting the simplification references in ASSET. We detail each step below.

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. Following university guidelines, workers were presented with an Information Sheet and had to “sign” a Consent Form before participating in the study. These documents can be found in Appendix A.



We ask you to rewrite each original sentence in order to make it easier to understand by non-native speakers of English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentences need to be grammatical, fluent, and retain the main ideas of their original counterparts without altering their meanings. More detailed instructions and examples can be found by clicking the "View Instructions" button.

Carefully read the instructions provided and then **simplify** the following sentences:

Sentence 1 of 3

Original: The second largest city of Russia and one of the world's major cities, St . Petersburg has played a vital role in Russian history.

Type the Simplification here...

Sentence 2 of 3

Original: US and European regulators gave approval to the purchase in February.

Type the Simplification here...

Sentence 3 of 3

Original: Wednesday afternoon, the Philippines said it had agreed with China to diplomatically resolve the dispute, according to the Associated Press.

Type the Simplification here...

Suggestions (Optional)

If you have any suggestions regarding the task, please provide them below:

Type your suggestions here...

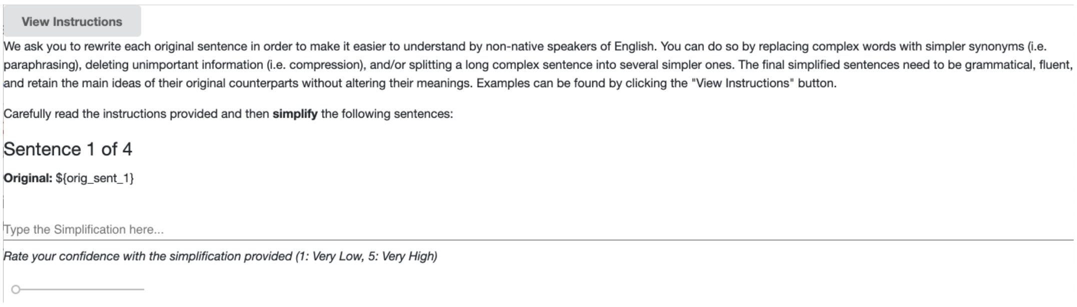
Figure 4.2 Qualification Test that workers had to pass before participating in the Simplification Task.

Simplification Instructions

Following Pellow and Eskenazi (2014a), we showed workers explanations and examples of multiple simplification operations, namely: lexical paraphrasing (lexical simplification and reordering), sentence splitting, and compression (deleting "unimportant" information). We also included an example where all simplification operations were performed. However, we clarified that it was at their discretion to decide which types of rewriting to execute in any given original sentence. Full instructions are available in Appendix B. The same set of guidelines was used for the Qualification Test and the Simplification Task, both detailed below.

Qualification Test

We provided a training session to workers in the form of a Qualification Test. Each HIT consisted of three sentences to simplify (Figure 4.2), 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 performed the Qualification Test in 5 batches, requesting increasing numbers of participants (20, 30, 40, 40, 50). This allowed us to: (1) gather suggestions from workers and adjust the instructions accordingly; and (2) modify the time allocated for the task and the payment that workers received for it. In the end, 180 workers took the Qualification Test, out of which 97 passed (54%) and were allowed to work on the Simplification Task.



We ask you to rewrite each original sentence in order to make it easier to understand by non-native speakers of English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentences need to be grammatical, fluent, and retain the main ideas of their original counterparts without altering their meanings. Examples can be found by clicking the "View Instructions" button.

Carefully read the instructions provided and then **simplify** the following sentences:

Sentence 1 of 4

Original: ${orig_sent_1}

Type the Simplification here...

Rate your confidence with the simplification provided (1: Very Low, 5: Very High)

Figure 4.3 Simplification Task that selected workers had to perform to collect manual simplifications.

Simplification Task

Only workers who passed the Qualification Test had access to this task. Similar to Pellow and Eskenazi (2014a), each HIT now consisted of four original sentences that needed to be simplified (Figure 4.3). 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). Furthermore, we randomly sampled submissions to verify their quality, and iteratively adjusted the instructions and/or selected new workers, accordingly. In the end, we collected 10 manual simplifications for each of the 2,359 original sentences in TurkCorpus. Table 4.1 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 participants in ASSET had more freedom to change the structure of the original sentences.

Table 4.1 Examples of simplifications collected for ASSET together with their corresponding version from TurkCorpus and HSplit for the same original sentences.

Original	Their eyes are quite small, and their visual acuity is poor.
ASSET	They have small eyes and poor eyesight.
TurkCorpus	Their eyes are very little, and their sight is inferior.
HSplit	Their eyes are quite small. Their visual acuity is poor as well.
Original	His next work, Saturday, 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.
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.
Original	He settled in London, devoting himself chiefly to practical teaching.
ASSET	He lived in London. He was a teacher.
TurkCorpus	He rooted in London, devoting himself mainly to practical teaching.
HSplit	He settled in London. He devoted himself chiefly to practical teaching.

4.2.2 Dataset Statistics

Table 4.2 presents some general statistics from simplifications in ASSET.[1] We show the same statistics for TurkCorpus and HSplit for comparison.

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. Furthermore, simplifications are shorter on average in ASSET, given that we allowed annotators to delete information that they considered unnecessary. Finally, while HSplit only provides simplifications for the test set of TurkCorpus, in ASSET we collected references for both the development and test sets (2,000 and 359 original sentences, respectively). In the next section, we further compare these datasets with more detailed textual features.

Table 4.2 General surface statistics for ASSET compared with TurkCorpus and HSplit. A simplification instance is an original-simplified sentence pair.

	ASSET	TurkCorpus	HSplit
Original Sentences	2,359	2,359	359
Number of References	10	8	4
Tokens per Reference	19.04	21.29	25.49
Simplification Instances			
1-to-1	17,245 (73%)	18,499 (98%)	408 (28%)
1-to-N	6,345 (27%)	373 (2%)	1,028 (72%)
Total	23,590 (100%)	18,872 (100%)	1,436 (100%)

4.3 Characterising Simplifications in ASSET

We study the data collected for ASSET by identifying the simplification operations executed, and by computing several textual features to measure the *abstractiveness* of the text transformations performed. 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.

4.3.1 Simplification Operations

We use the annotation algorithms introduced in Chapter 3 to automatically identify the simplification operations present in the manual references of each dataset. In particular, we compute the sentence-level labels (obtained with SimAlign word alignments) for the three

[1] We identified 1-to-N instances using regular expressions that were manually verified in a random sample.

operations we instructed the workers to perform: DELETE (for compression), REPLACE (for lexical paraphrasing) and SPLIT (for sentence splitting).

Single-Label Analysis

Figure 4.4 shows the percentages of simplification instances in each dataset where a particular simplification operation was executed. In the plot, instances where none of the three main operations under study were automatically-annotated are either considered as: (1) 'none' if the original sentence and its simplification are exactly the same, or (2) 'unidentified' in any other case. Through manual examination, we noticed that the latter corresponds to instances where very minimal changes were made, such as capitalisation.

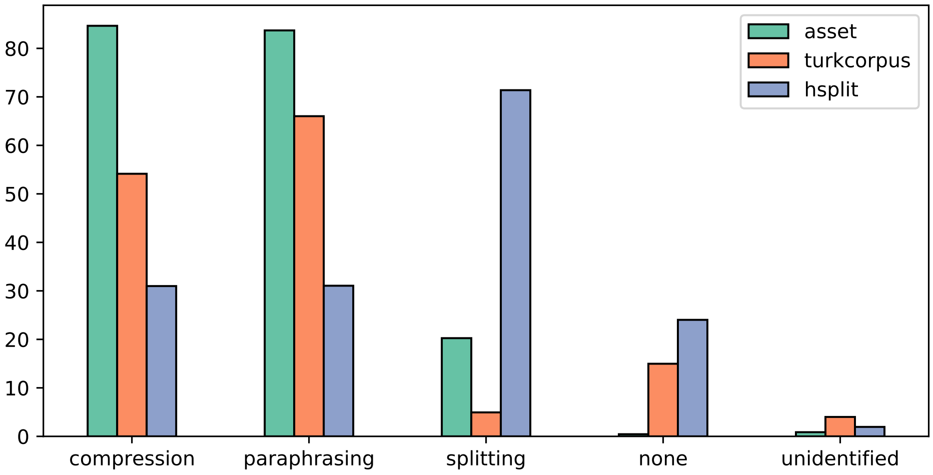


Figure 4.4 Proportion of simplification instances that contain a particular simplification operation in ASSET, TurkCorpus and HSplit.

Some form of **compression** is present in all datasets, but with the highest percentage in ASSET. This is expected since we explicitly informed the annotators that this was a valid operation to perform. However, the high value for TurkCorpus (and HSplit to some extend) is unanticipated. After further analysis, we discovered that this is caused by how the annotation algorithms identify DELETE at the sentence level: if at least one word has been labelled as DELETE, then the whole instance is labelled as undergoing compression. This causes an over-estimation of the operation, since some cases of deletion are due to minimal changes to preserve grammaticality after applying other operations (e.g. removing a relative pronoun after splitting a sentence). Nonetheless, the differences between datasets are significant enough to argue that compression is executed in more instances of ASSET than TurkCorpus and HSplit. A more nuanced analysis of the frequency of this operation is presented in the next section.

Lexical paraphrasing is identified in almost all instances of ASSET and in most of TurkCorpus, evidencing the relevance of this operation when simplifying a sentence in all situations. Similar to the identification of compression in HSplit, a manual examination showed that the lexical paraphrases in this dataset were performed to preserve grammaticality rather than for simplification purposes, for example: *one* → *a*, *'s* → *is*, *the* → *a*. According to the typology of simplification operations presented in Chapter 3, these replacements should be considered as cases of SIMPLE REWRITE. However, the current version of our annotation algorithms are incapable of identifying them as such. Whilst this limitation over-estimates the presence of lexical paraphrasing, we argue that the relative difference between datasets still holds the argument that simplifications in ASSET do contain more lexical simplifications.

Sentence splitting has its highest percentage in HSplit, but not all its instances contain the operation. This is because HSplit is composed of two sets of manual references: one where annotators were asked to split sentences as much as they could (similar to the split-and-rephrase task), and one where they were asked to split the original sentence only if it made the simplification easier to read and understand. In our analyses, we have considered HSplit as a whole because differences between datasets far outweigh differences between these two sets. The presence of splitting in TurkCorpus is also unusual, since the annotators were instructed not to perform it. However, a manual examination showed that at least one of the workers did not completely follow these guidelines.[2] This evidences the importance of sentence splitting when humans simplify sentences, but that it is not mandatory to perform every possible split to improve a sentence's readability. For that reason, we argue that sentence splitting in ASSET is adequately represented, albeit increasing its frequency could improve the manual references.

Multi-Label Analysis

It is possible for a simplification instance to have been produced by executing more than one operation. In fact, that is a desirable characteristic of the manual references in ASSET. Figure 4.5 shows the percentages of simplification instances in each dataset where a particular **combination** of simplification operation was executed.

Simplifications in TurkCorpus are focused on two operations: compression and lexical paraphrasing, either by applying them in isolation or together. On the other hand, instances in HSplit focus on sentence splitting, or on combining it with compression or paraphrasing. As it was mentioned in the previous analysis, in these cases of multi-operation instances, it is likely that the deletions (for TurkCorpus) and paraphrases (for HSplit) were performed to

[2] https://github.com/cocoxu/simplification/blob/master/data/turkcorpus/test.8turkers.tok.turk.1

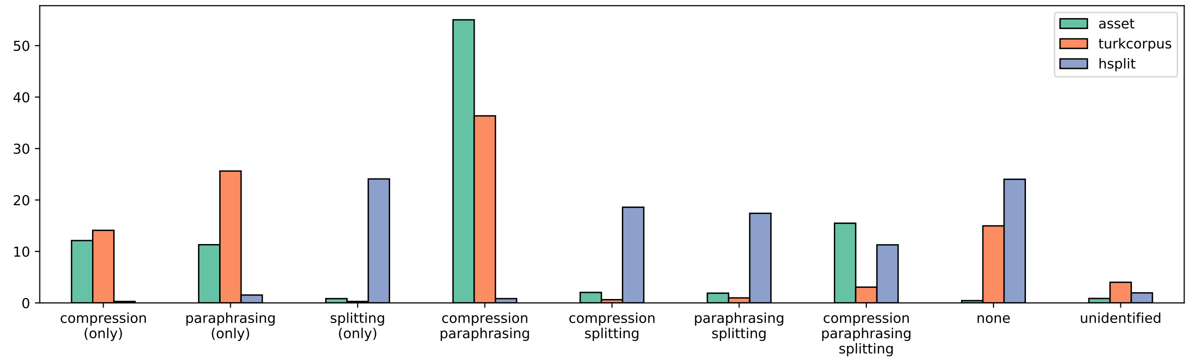


Figure 4.5 Proportion of simplification instances that contain a particular combination of simplification operations in ASSET, TurkCorpus and HSplit.

preserved the fluency and grammatical structure of the simplified sentences after applying the corresponding main operations. In the case of ASSET, most of the collected simplifications correspond to a combination of compression and lexical paraphrasing. However, there are also instances with only compression or paraphrasing, and even instances with all three operations. Overall, this is evidence of the variability of the collected manual references in ASSET.

Finally, ASSET presents the lowest percentage of instances where no modifications were performed to the original sentences. This contrasts with the relatively-high percentages in TurkCorpus and HSplit. We can hypothesise two reasons for this: (1) the human editors were not able to come up with suitable simplifications based on only applying a single operation (either lexical paraphrasing or splitting); or (2) the quality controls did not identify these instances on time to better instruct the annotators not to simply copy the original sentences. The behaviour in HSplit could be explained by (1), since they used expert editors, whilst TurkCorpus could be explained by (2) since, as mentioned before, a few of their editors did not completely follow their simplification guidelines.

4.3.2 Textual Features

We also computed several low-level features for all simplification instances using the `tseval` package (Martin et al., 2018), as an alternative way of describing the simplification operations:[3]

- **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 (Loper and Bird, 2002).

[3] For this particular section in the ASSET's paper, a collaborator was in charge of describing and computing the textual features (included here for completeness), whilst the thesis author contributed to the analysis of the scores.

- **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 normalised 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 in 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., 2016). 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

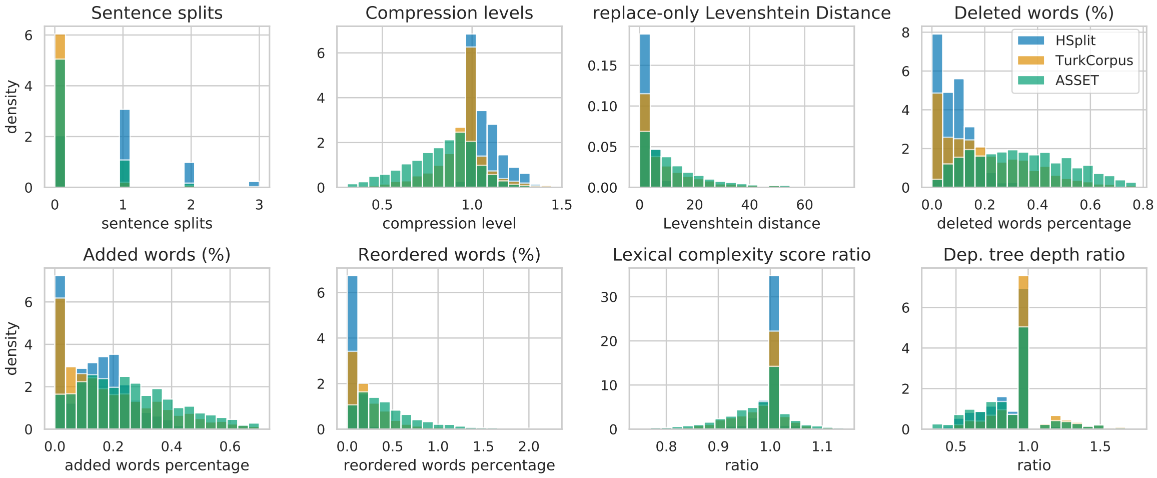


Figure 4.6 Density of textual features in simplifications from HSplit, TurkCorpus, and ASSET.

Table 4.3 Proportion 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.

	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%

dependency trees. Parsing is performed using spaCy.[4] This feature serves as a proxy to measure improvements in structural simplicity.

For each dataset, each textual feature was computed for all its simplification instances and then aggregated as a histogram (Figure 4.6) and as a percentage (Table 4.3). For the latter, in particular, we report the percentage of sentences that: have at least one sentence split, have a compression 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).

Similar to what we observed in the previous section, sentence splitting is practically non-existent in TurkCorpus (only 4.6% of instances have one split or more), and are more present and distributed in HSplit (density values above 1 for various numbers of splits). In ASSET, annotators tended not to split sentences (high density of 0 splits), and those who did mostly divided the original sentence into just two sentences (1 split).

[4] https://spacy.io/

Compression is a differentiating feature of ASSET (highest value in Table 4.3). 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 signalling that annotators consider deleting information as an important simplification operation.

By analysing 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 comfirmed 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, indicating that significant structural changes were not made. This a strange result, especially considering the high percentage of sentence splitting in HSplit. We hypothesise that this textual feature may not have been suitable for assessing structural simplicity.

4.4 Rating Simplifications in ASSET

In this Section, we measure the quality of the collected simplifications using human judges. In particular, we study if the multi-operation simplifications in ASSET are preferred over lexical-paraphrase-only or splitting-only simplifications in TurkCorpus and HSplit, respectively.

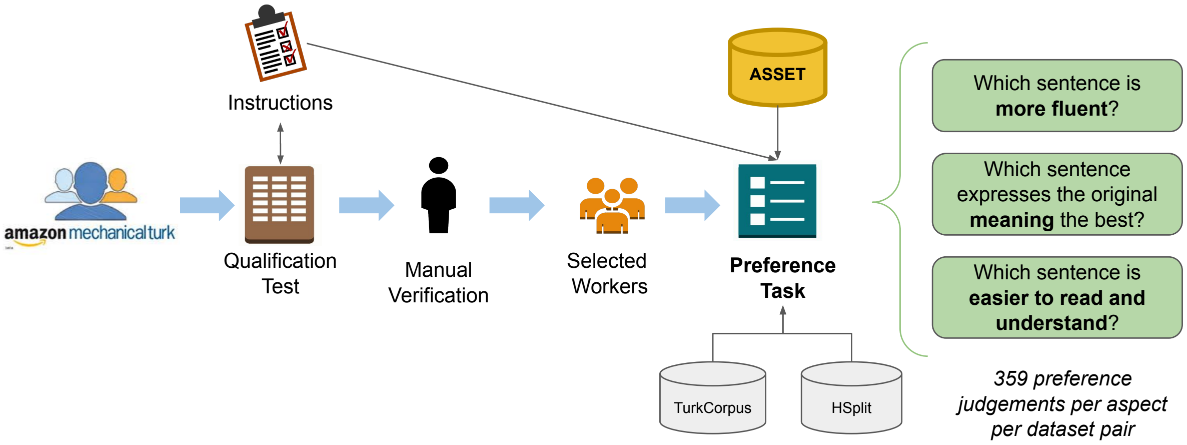

Instructions
amazon mechanical turk
Qualification Test
Manual Verification
Selected Workers
ASSET
Preference Task
TurkCorpus
HSplit
Which sentence is more fluent?
Which sentence expresses the original meaning the best?
Which sentence is easier to read and understand?
359 preference judgements per aspect per dataset pair

Figure 4.7 Crowdsourcing methodology followed to collect human preference judgements comparing simplifications from ASSET vs TurkCorpus and ASSET vs HSplit.

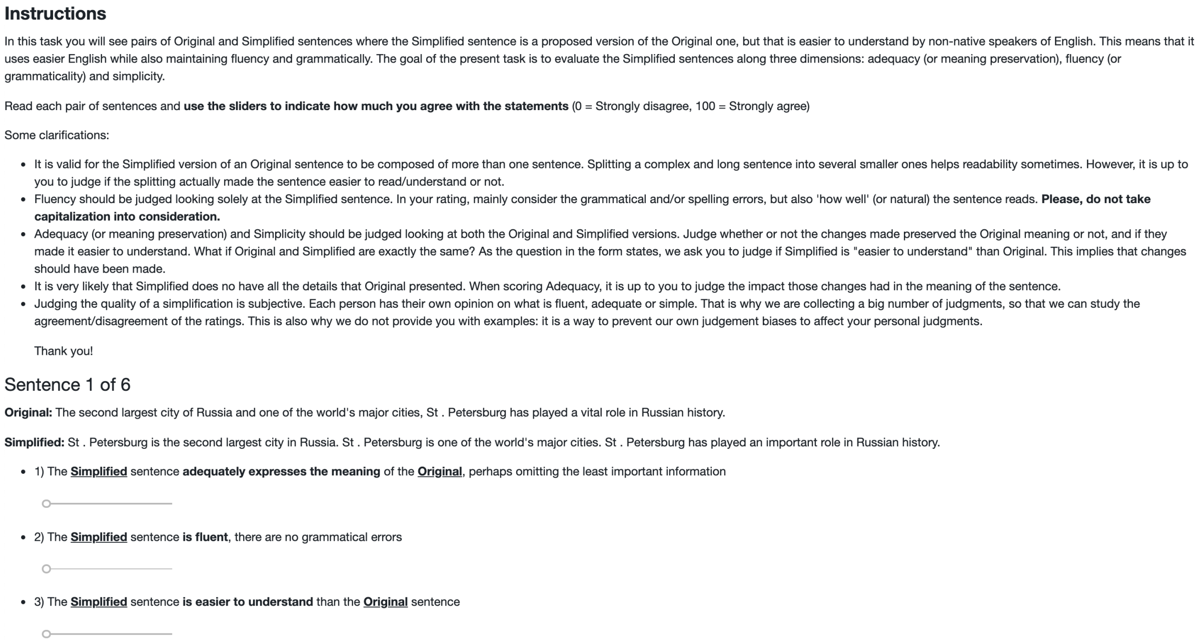
4.4.1 Collecting Human Preferences

Preference judgements were crowdsourced with a protocol similar to that of the manual simplifications (Sec. 4.2.1). An overview of the methodology followed can be seen in Figure 4.7.

Selecting Human Judges

Workers needed to comply with the same basic requirements as described in Sec. 4.2.1. For this task, the Qualification Test consisted of 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 statements questioning the quality of each aspect under review (see Fig. 4.8 for specific instructions and questions asked). Using continuous scales when crowdsourcing human evaluations is common practice in Machine Translation (Barrault et al., 2019; Bojar et al., 2018), since it results in higher levels of inter-annotator agreement (Graham et al., 2013). The six sentence pairs for the Rating Qualification Test consisted of:

- Three submissions to the Simplification Qualification Test, 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.



Instructions

In this task you will see pairs of Original and Simplified sentences where the Simplified sentence is a proposed version of the Original one, but that is easier to understand by non-native speakers of English. This means that it uses easier English while also maintaining fluency and grammatically. The goal of the present task is to evaluate the Simplified sentences along three dimensions: adequacy (or meaning preservation), fluency (or grammaticality) and simplicity.

Read each pair of sentences and **use the sliders to indicate how much you agree with the statements** (0 = Strongly disagree, 100 = Strongly agree)

Some clarifications:

- It is valid for the Simplified version of an Original sentence to be composed of more than one sentence. Splitting a complex and long sentence into several smaller ones helps readability sometimes. However, it is up to you to judge if the splitting actually made the sentence easier to read/understand or not.
- Fluency should be judged looking solely at the Simplified sentence. In your rating, mainly consider the grammatical and/or spelling errors, but also 'how well' (or natural) the sentence reads. **Please, do not take capitalization into consideration.**
- Adequacy (or meaning preservation) and Simplicity should be judged looking at both the Original and Simplified versions. Judge whether or not the changes made preserved the Original meaning or not, and if they made it easier to understand. What if Original and Simplified are exactly the same? As the question in the form states, we ask you to judge if Simplified is "easier to understand" than Original. This implies that changes should have been made.
- It is very likely that Simplified does no have all the details that Original presented. When scoring Adequacy, it is up to you to judge the impact those changes had in the meaning of the sentence.
- Judging the quality of a simplification is subjective. Each person has their own opinion on what is fluent, adequate or simple. That is why we are collecting a big number of judgments, so that we can study the agreement/disagreement of the ratings. This is also why we do not provide you with examples: it is a way to prevent our own judgement biases to affect your personal judgments.

  Thank you!

Sentence 1 of 6

Original: The second largest city of Russia and one of the world's major cities, St . Petersburg has played a vital role in Russian history.

Simplified: St . Petersburg is the second largest city in Russia. St . Petersburg is one of the world's major cities. St . Petersburg has played an important role in Russian history.

- 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

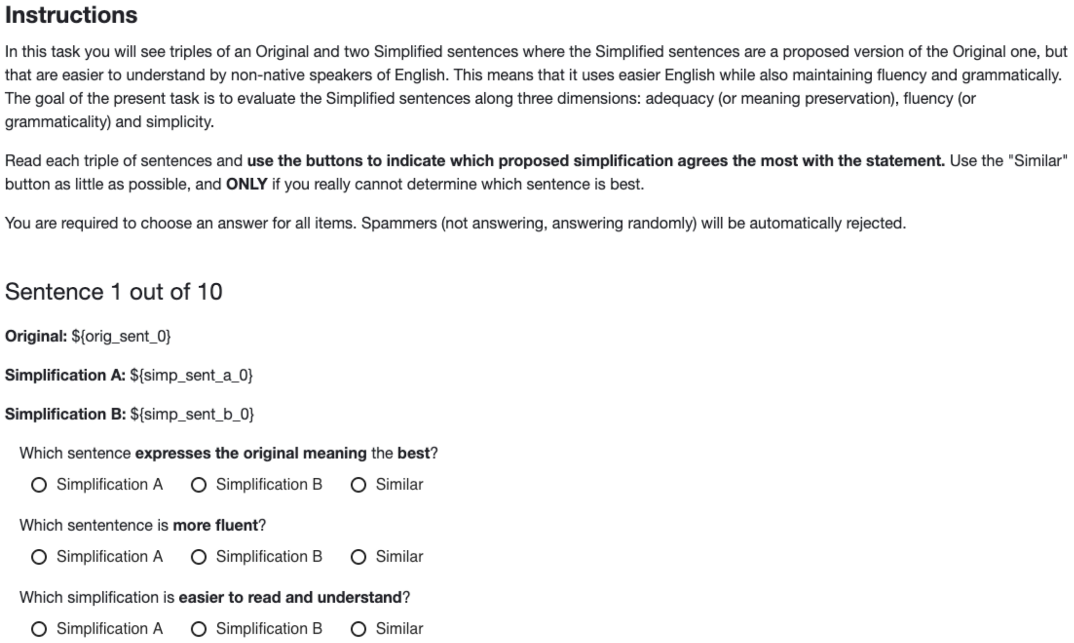
Figure 4.8 Qualification Test that workers had to pass to participate in the Preference and Rating Tasks.

- 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.

As before, all submissions were manually reviewed to validate the quality control established, and to select the qualified workers for the task. In the end, 37% (146/390) of workers who took the Rating Qualification Test were allowed to participate in the Preference 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 HSplit. Then, qualified workers were asked to choose which simplification (ASSET vs TurkCorpus and ASSET vs HSplit) best answers three questions related to fluency, meaning preservation and simplicity (see Fig. 4.9 for specific instructions and questions). Workers were also allowed to judge simplifications as



Instructions

In this task you will see triples of an Original and two Simplified sentences where the Simplified sentences are a proposed version of the Original one, but that are easier to understand by non-native speakers of English. This means that it uses easier English while also maintaining fluency and grammatically. The goal of the present task is to evaluate the Simplified sentences along three dimensions: adequacy (or meaning preservation), fluency (or grammaticality) and simplicity.

Read each triple of sentences and **use the buttons to indicate which proposed simplification agrees the most with the statement.** Use the "Similar" button as little as possible, and **ONLY** if you really cannot determine which sentence is best.

You are required to choose an answer for all items. Spammers (not answering, answering randomly) will be automatically rejected.

Sentence 1 out of 10

Original: ${orig_sent_0}

Simplification A: ${simp_sent_a_0}

Simplification B: ${simp_sent_b_0}

Which sentence **expresses the original meaning** the **best**?

○ Simplification A ○ Simplification B ○ Similar

Which sententence is **more fluent**?

○ Simplification A ○ Simplification B ○ Similar

Which simplification is **easier to read and understand**?

○ Simplification A ○ Simplification B ○ Similar

Figure 4.9 Preference Task that selected workers had to perform to compare simplifications from ASSET against those in TurkCorpus and HSplit.

"similar" when they could not determine which one was better. In the end, we collected 1,077 judgements per dataset pair (359 sentence pairs * 3 aspects).[5]

Table 4.4 Proportion 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).

	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 4.5 Examples of simplifications from ASSET and TurkCorpus or HSplit used in the Preference Task. Sentence pairs were compared for Fluency (F), Meaning Preservation (M) and Simplicity (S). Winners in each aspect are marked with ✓.

	Sentence Pairs Selected for Comparison	F	M	S
Original	He settled in London, devoting himself chiefly to practical teaching.			
ASSET	He became a teacher in London.	✓		✓
TurkCorpus	He chose London as his residing place and dedicated himself mainly to practical teaching.		✓	
Original	His next work, Saturday, 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.	✓		✓
HSplit	His next work is Saturday. It follows an especially eventful day. A day in the life of a successful neurosurgeon.		✓	

4.4.2 Results and Analysis

Table 4.4 (top section) presents, for each quality aspect, 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 4.4). In this case, however, the differences in preferences are greater than with TurkCorpus. This could indicate that for our particular set of raters (i.e. native or proficient English speakers), changes in syntactic structure are not enough for a sentence to be considered simpler. Results could be different with raters from a different population group. Table 4.5 presents sentence pairs that exemplify these findings.

We hypothesise that our raters favouring TurkCorpus and HSplit over ASSET in terms of meaning preservation could be caused by an inherent bias that equates preserving meaning with being conservative (i.e. performing minimal changes). This could be tackled by providing more detailed guidelines. For instance, by explaining that compression is a valid (and relevant) simplification operation, with examples that show what should (and should not) be deleted so as to preserve most of the original meaning of the sentence. In the same line, we could refine

[5] For this particular section in the ASSET’s paper, a collaborator was in charge of collecting the human judgements and describing the results, and it is included here (with a few modifications) for completeness.

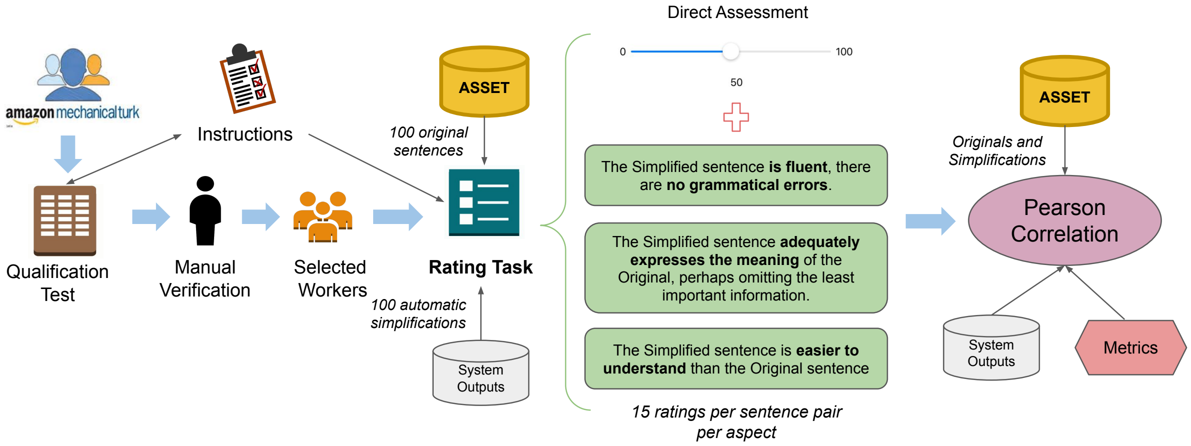

amazon mechanical turk
Instructions
Qualification Test
Manual Verification
Selected Workers
ASSET
100 original sentences
Rating Task
100 automatic simplifications
System Outputs
Direct Assessment
0
100
50
The Simplified sentence is fluent, there are no grammatical errors.
The Simplified sentence adequately expresses the meaning of the Original, perhaps omitting the least important information.
The Simplified sentence is easier to understand than the Original sentence
15 ratings per sentence pair per aspect
ASSET
Originals and Simplifications
Pearson Correlation
System Outputs
Metrics

Figure 4.10 Methodology followed to collect direct assessment ratings on the quality of automatic simplifications, to then measure their correlation with evaluation metrics computed using manual references from ASSET.

the question used to elicit the ratings/preferences to establish a difference between "meaning" and "main idea", and then collect more fine-grained judgements.

4.5 Evaluating Evaluation Metrics

In this Section we study the behaviour of automatic evaluation metrics for Sentence Simplification when using ASSET's manual references. In particular, we measure the correlation of standard metrics with human judgements of fluency, meaning preservation and simplicity, on automatic simplifications produced by state-of-the-art systems. An overview of the methodology followed can be seen in Figure 4.10.

4.5.1 Experimental Setup

Evaluation Metrics

We analysed the behaviour of two standard metrics in automatic evaluation of Sentence Simplification outputs: BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016).[6] BLEU has shown positive correlation with human judgements of grammaticality and meaning preservation (Wubben et al., 2012; Xu et al., 2016; Štajner et al., 2014), while SARI has high correlation with judgements of simplicity gain (Xu et al., 2016). In our experiments, we used the implementations of these metrics available in EASSE (Alva-Manchego et al., 2019a). We

[6] These metrics are detailed in Sec. 2.3.2.

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 (Martin et al., 2020), an encoder-decoder architecture conditioned on explicit attributes of Sentence Simplification.[7] All these system outputs (and many more) are included in EASSE.

Collection of Human Ratings

We randomly selected 100 original sentences from ASSET and, for each of them, we sampled one system simplification. The automatic simplifications were chosen so that the distribution of simplification operations (e.g. sentence splitting, compression, lexical paraphrasing) would match that from human simplifications in ASSET.[8] 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 Sec. 4.4.1. In the end, we collected 4,500 human ratings (100 segments * 15 ratings * 3 aspects).

4.5.2 Inter-Annotator Agreement

We followed the process suggested in (Graham et al., 2013). First, we normalised the scores of each rater by their individual mean and standard deviation, which helps eliminate individual judge preferences. Then, the normalised 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

[7] These simplification systems are detailed in Sec. 2.4.

[8] The identification of operations was based on textual features (Sec. 4.3.2) instead of the annotation algorithms.

Table 4.6 Pearson correlation of human ratings with **automatic metrics** on system simplifications. * indicates a significance level of p-value < 0.05.

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

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.[9] According to the scale of (Landis and Koch, 1977), all values point to higher than moderate levels of agreement. This is evidence of the reliability of the scores collected, while also showing the subjectivity of the simplification task.

4.5.3 Correlation with Evaluation Metrics

We computed the Pearson correlation between the normalised ratings and the automatic evaluation metrics of our interest (BLEU and SARI) using ASSET or TurkCorpus as the set of manual 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., 2018a). Results are reported in Table 4.6.[10]

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 (Sulem et al., 2018b; Wubben et al., 2012; Xu et al., 2016). 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"*.[11] The judgements they requested differ from

[9] These scores were computed by a collaborator in the ASSET's paper.

[10] All the scores in this Section were computed by a collaborator in the ASSET's paper. The thesis author was in charge of analysing and discussing them.

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

Table 4.7 Pearson correlation of human ratings with **textual features** on system simplifications. * indicates a significance level of p-value < 0.01.

Text 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

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 rewriting operations.

In Table 4.7, we further analyse the human ratings by computing their correlations with textual features (as in Sec. 4.3). The results reinforce our previous observations that ratings on Meaning Preservation 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. Simplicity did not show strong correlations (positive or negative) with any of the features. We hypothesise that, for our raters, the precision of the changes made was more valuable to assess the quality of automatic simplifications in this aspect.

4.6 Summary and Final Remarks

In this Chapter, we attempted to answer the following research question: **Do we need multi-reference multi-operation evaluation datasets to support better assessment of Sentence Simplification models?**

First, we argued that current datasets for automatic evaluation prevent studying models' capabilities to perform more *abstractive* simplifications, i.e. where several changes were applied to the original sentences. For instance, whilst TurkCorpus and HSplit provide multiple manual simplification references (a desirable feature in evaluation of text-to-text generation tasks), they are focused on single operations: lexical paraphrasing and sentence splitting, respectively. Based on previous research on how human editors simplify (Sec. 2.1.1) and our own manual study (Sec. 3.1), we **motivated the creation of a new evaluation dataset** that, on top of being muti-reference, contains manual references where multiple simplification operations were applied. With this goal, we **proposed to use crowdsourcing** to collect such simplifications, based on the success of previous work (Pellow and Eskenazi, 2014a; Scarton et al., 2018a; Xu et al., 2016), whose best practices for quality control were adopted in our data collection.

We **introduced ASSET, a new dataset for tuning and evaluation** of Sentence Simplification models.[12] Using the Amazon Mechanical Turk platform, we: (1) designed HITs with adequate guidelines to collect manual simplifications with multiple operations (compression, lexical paraphrasing and splitting); (2) performed qualification tests to select suitable workers; and (3) conducted a crowdsourcing task that iteratively refined the instructions and selection of workers. As a result, we collected 10 high-quality multi-operation simplification references for each of the 2,359 original sentences in TurkCorpus (2,000 for the development set and 359 for the test set). Since ASSET is aligned to TurkCorpus and HSplit, we compared the manual simplifications in each dataset through quantitative and qualitative experiments.

Manual references in ASSET, TurkCorpus and HSplit were first automatically-compared and showed that **simplifications in ASSET offer more variability in the operations performed by human editors**. The analysis consisted of identification of sentence-level operations via the annotation algorithms introduced in Chapter 3, and computation of a series of surface-level textual features. In addition, we crowdsourced human preference judgements that compare simplifications in ASSET against TurkCorpus and ASSET against HSplit for all 359 sentences in the test set along three quality dimensions: fluency, meaning preservation and simplicity. Results showed that **humans judged simplifications in ASSET as simpler than those in other evaluation datasets**. These findings support ASSET as a new high-quality benchmark for evaluation of Sentence Simplification systems.

Finally, we **motivated the development of new metrics for evaluating multi-operation sentence simplifications**. Based on the Direct Assessment methodology (Graham et al., 2017) used in Machine Translation, we crowdsourced human ratings on fluency, meaning preservation and simplicity for automatic outputs of five state-of-the-art sentence simplification systems. Then, we analysed whether these human ratings correlated with BLEU or SARI multi-reference scores computed using ASSET's manual references. We found that, whilst BLEU correlates with meaning preservation, none of the metrics shows a strong correlation with simplicity judgements. In particular, we argue that SARI, a widely-used simplification-specific metric, is unsuitable for assessing automatic simplifications where multiple operations have been applied. Moving forward, we should either develop new automatic metrics, or implement more suitable evaluation protocols that take advantage of different evaluation metrics that already exist (even

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

for other text-to-text generation tasks), and consider the specific characteristics of the manual references being used. We explore the latter in Chapter 5.

All the previously-described experiments and findings allow us to suggest an answer to our research question: When allowed to do so, human editors tend to apply several rewriting operations when simplifying sentences, and generate various references that are simpler than those produced by executing a single simplification operation. However, current evaluation metrics are not capable of exploiting this improvement in reference quality. Therefore, these less-restricted multi-operation simplifications could serve to study and develop more suitable evaluation metrics and/or protocols that improve automatic evaluation.

Chapter 5

The (Un-)Suitability of Automatic Metrics

At the end of the previous Chapter, we highlighted some issues with commonly-used evaluation metrics to assess the quality of automatic simplifications using multi-operation manual references. Preliminary findings showed that BLEU and SARI scores, in particular, have poor correlation with human judgements of simplicity in this evaluation setting. In this Chapter, we extend that initial study to better answer the following research question: **Can current evaluation metrics measure the ability of automatic systems to perform multi-operation simplifications?** We conduct, to the best of our knowledge, the first meta-evaluation of automatic metrics for Sentence Simplification. We focus on their correlation with human judgements on simplicity, based on our findings from Chapter 4. As a result, we propose a set of guidelines for automatic evaluation of sentence simplifications that improves the computation and interpretation of automatic scores, especially for multi-operation simplifications.

First, we motivate a meta-evaluation of automatic metrics in Sentence Simplification based on the need to better understand how metrics used in the area behave across different ways of manually assessing simplicity, and inspired by similar work that has produced important insights in Machine Translation (MT), mainly (Sec. 5.1). Then, we describe the characteristics and limitations of two existing datasets with human judgements on Simplicity Gain and Structural Simplicity of system outputs, which motives the collection of a new dataset with Simplicity scores crowdsourced through Direct Assessment (Sec. 5.2). After that, we conduct our meta-evaluation to analyse how the correlations between popular sentence-level automatic metrics and human judgements vary under different test conditions: the level of perceived simplicity, the simplification approach implemented by the systems, and the set of manual simplification references used (Sec. 5.3). Based on our findings, we elaborate a set of recommendations for better evaluation of automatic sentence simplifications; and suggest ways to improve the current state of automatic evaluation (Sec. 5.4). Finally, we summarise our mains results and highlight the applicability of our findings in current research in the area (Sec. 5.5).

5.1 Motivation

The preferred method for evaluating the quality of automatic simplifications is eliciting human judgements on different criteria, such as grammaticality, meaning preservation and simplicity. However, these can be costly to obtain at large scale, or unreasonable to collect while tuning simplification models. This creates scenarios where automatic metrics are used as proxies for human judgements. Therefore, it is important to understand how these metrics behave under different circumstances, to better interpret their scores. We first review common practices for collecting human judgements on the simplicity of system outputs against which metrics are evaluated, and motivate our choice of Direct Assessment as our data labelling methodology. Then, we briefly explain the benefits of conducting meta-evaluations of automatic metrics.

5.1.1 Human Evaluation of Simplicity

When obtaining human judgements on the simplicity of a system output, there are three important components to consider: the question to elicit the judgement, what the judges are shown, and how they submit their judgement. It is generally agreed to show both the original and simplified sentences so that the raters can determine if the latter is simpler than the former in a particular way. However, several variations have been tested for the other two components.

The most common approach is to ask how much simpler the system output is compared to the original sentence, using Likert scales of 0-5, 1-4 or 1-5 (the higher the better) to submit discrete scores (Alva-Manchego et al., 2017; Dong et al., 2019; Feblowitz and Kauchak, 2013; Guo et al., 2018; Jiang et al., 2020; Kriz et al., 2019; Kumar et al., 2020; Narayan and Gardent, 2014, 2016; Vu et al., 2018; Woodsend and Lapata, 2011a; Wubben et al., 2012; Zhang and Lapata, 2017). A variation in the question is presented in (Štajner et al., 2014) asking raters about how much irrelevant information was eliminated with a 1-3 Likert scale. A variation in the scale is presented in (Nisioi et al., 2017) with -2 to +2 scores instead, allowing to distinguish instances with no changes in their simplicity (0), and instances where the automatic system actually hurt the readability of the original sentence (-1 or -2).

Most work does not specify what "being simpler" entails, and trusts human judges to use their own understanding of the concept. In contrast, Xu et al. (2016) experimented with Simplicity Gain, asking judges to count *"how many successful lexical or syntactic paraphrases* *occurred in the simplification"*. The authors argue that this framing of the task allows for easier judgements and more informative interpretation of the scores, while reducing the bias towards models that perform minimal modifications. In a similar fashion, Nisioi et al. (2017) and Cooper and Shardlow (2020) asked judges to count the number of changes made by automatic systems, and then to identify how many of them were "correct" (i.e. preserved meaning and grammaticality, while making the sentence easier to understand). On a different line of work, Sulem et al. (2018b,c) focused on Structural Simplicity, requesting judges to use the -2 to +2 scale to answer *"is the output simpler than the input, ignoring the complexity of the words?"* This is intended to focus the evaluation in a specific operation: sentence splitting.

For the dataset collected as part of our study (Sec. 5.2.3), we follow common practice and present human judges with both original sentences and their automatic simplifications. Furthermore, since the focus of this Thesis is on multi-operation simplifications, we rely on a general definition of simplicity instead of one for an specific (set of) operation(s). Finally, as in Sec. 4.5, we experiment with collecting continuous scores following the Direct Assessment methodology (Graham et al., 2017), since they can be standardised to remove individual rater's biases, resulting in higher inter-annotator agreement (Graham et al., 2013).

5.1.2 Meta-Evaluation of Automatic Metrics

Studies on the correlation of human judgements on simplicity and automatic scores have mostly been performed when introducing new metrics or datasets.[1] Xu et al. (2016) argued that SARI correlates with crowdsourced judgements of Simplicity Gain when the simplification references had been produced by lexical paraphrasing, whilst SAMSA (Sulem et al., 2018b) was shown to correlate with expert judgements of Structural Simplicity. When introducing HSplit (Sulem et al., 2018a), a dataset of manual references for sentence splitting, the authors argued that BLEU (Papineni et al., 2002) was not a good estimate for (Structural) Simplicity. However, these studies did not analyse if the absolute correlations varied in different subgroups of the data. In contrast, we show that correlations do vary depending on the quality of the simplifications, the system types and the set of references used.

Our investigation is mainly inspired by previous work in MT evaluation. There is a long tradition of meta-evaluations in this area, mostly due to the WMT Metrics Shared Tasks that, since 2007, promote the development of new metrics for MT, and compares them in a standard setting using human judgements collected through Direct Assessment, primarily in the latest years (Bojar et al., 2017b, 2016b; Ma et al., 2018, 2019) . This data has been used to further

[1] Štajner et al. (2014) analysed several MT metrics without introducing a new resource, but focused on human judgements of grammaticality and meaning preservation.

study the behaviour of metrics at the sentence-level across different dimensions (Fomicheva and Specia, 2019), to analyse the protocols for metric evaluation at the system-level (Mathur et al., 2020), to study the effect of the quality of references used to compute metrics (Freitag et al., 2020), among others. Similar studies have been performed for Summarisation (Fabbri et al., 2020; Peyrard, 2019), Grammatical Error Correction (Chollampatt and Ng, 2018), Image Captioning (Kilickaya et al., 2017), and Natural Language Generation, in general (Novikova et al., 2017; Reiter and Belz, 2009).

5.2 Datasets with Human Judgements on Simplicity

In this Section, we describe the three datasets that will be used in our meta-evaluation study. Each dataset is composed of a set of original sentences and their automatic simplifications produced by several simplification systems. Most importantly, each simplification instance in the datasets was evaluated by humans along several quality aspects, such as grammaticality, meaning preservation and (some form of) simplicity. These datasets were chosen (or created) since each provides a different kind of simplicity judgement: Simplicity Gain (Xu et al., 2016), Structural Simplicity (Sulem et al., 2018c), and Simplicity based on Direct Assessment (new). This allows the study of the behaviour of the metrics along varied ways of measuring simplicity.

5.2.1 Simplicity Gain Dataset

This dataset was collected in (Xu et al., 2016) to study the suitability of FKBLEU and SARI as automatic metrics that evaluate the Simplicity Gain of automatic simplifications.[2] The authors used 93 original sentences from the test set of TurkCorpus and outputs from four automatic Sentence Simplification systems:

- PBMT-R (Wubben et al., 2012), which exploits the phrase-based statistical MT model Moses (Koehn et al., 2007) and chooses the candidate automatic simplification that is most dissimilar to the original sentence.
- SBMT-{BLEU | FKBLEU | SARI} (Xu et al., 2016), variations of a syntax-based statistical MT model trained on paraphrases from the Paraphrase Database (Ganitkevitch et al., 2013) and tuned using BLEU, FKBLEU or SARI, respectively.

For their Simplicity Gain judgements, workers on Amazon Mechanical Turk (AMT) were asked to count the number of *"successful lexical or syntactic paraphrases occurred in the*

[2] Details on these metrics were presented in Sec. 2.3.2.

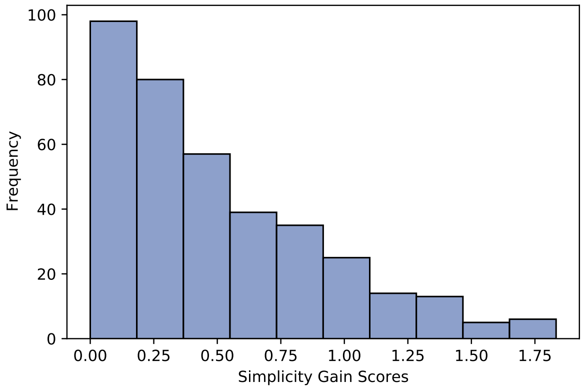


Figure 5.1 Distribution of Simplicity Gain scores in the dataset of (Xu et al., 2016).

simplification" (Xu et al., 2016). The judgements from five different Workers are averaged to get the final score for each instance. The authors do not report inter-annotator agreement.

This dataset has some notable limitations that could prevent the generalisation of findings based on its data. For instance, the number of evaluated instances (372) is small, and they were produced by only four automatic systems, three of which have very similar characteristics. In addition, as shown in Figure 5.1, the evaluated systems did not perform significant simplification changes (as judged by humans), since most instances were rated with Simplicity Gain scores below 1, with a high frequency of values between 0 and 0.25.

5.2.2 Structural Simplicity Dataset

This dataset was collected in (Sulem et al., 2018c) to evaluate the performance of Sentence Simplification models that mix hand-crafted rules (based on a semantic parsing) for sentence splitting, with standard MT-based architectures for lexical paraphrasing. Later, Sulem et al. (2018a) further exploited this data to examine the suitability of BLEU for assessing Structural Simplicity.[3] Sulem et al. (2018c) used the first 70 sentences from the test set of TurkCorpus, and outputs from 25 automatic systems (all introduced in their paper, unless explicitly stated):[4]

- Hybrid (Narayan and Gardent, 2014), which relies on phrase-based statistical MT coupled with semantic analysis to learn to split sentences.

[3] Sulem et al. (2018b) also collected a dataset with Structural Simplicity judgements. However, they used simplification instances from PWKP (Zhu et al., 2010), which is unsuitable for the purposes of our analysis since: (1) it does not contain manual simplification references, but automatic alignments to sentences from Simple Wikipedia; and (2) it is single-reference, which is unfair for reference-based metrics.

[4] The authors include two other outputs: "Identity" (repeating the original sentence) and "Simple Wikipedia" (the reference from PWKP). However, we exclude them from our study due to our focus on automatic systems.

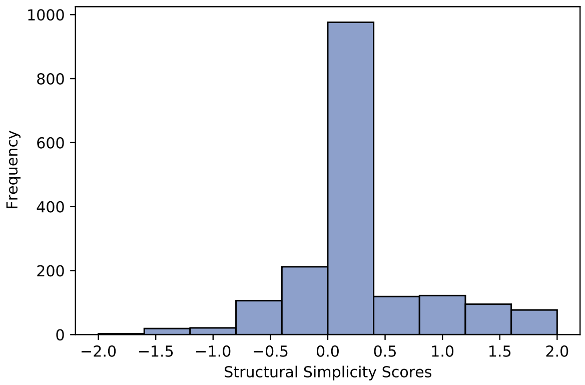


Figure 5.2 Distribution of Structural Simplicity scores in the dataset of (Sulem et al., 2018c).

- SBMT-SARI, explained before, relies on a syntax-based statistical MT approach.
- DSS (Direct Semantic Splitting) consists of two hand-crafted rules for sentence splitting based on UCCA (Abend and Rappoport, 2013) semantic annotations provided by the TUPA parser (Hershcovich et al., 2017). There is also a version of this rule-based system where the semantic annotations are manual (DSS$^{\mathrm{m}}$).
- NTS (Nisioi et al., 2017) is a neural sequence-to-sequence model that uses a standard encoder-decoder architecture with attention (more details in Sec. 2.4.1). The authors include four versions of this model, mixing initialisation with default or word2vec embeddings, and selecting the highest or fourth-best hypothesis according to SARI.
- SENTS first uses DSS or DSS$^{\mathrm{m}}$ for sentence splitting, and then the resulting output goes through a version of NTS. Different combinations amount for 8 systems.
- SEMoses is a version of SENTS where Moses replaces NTS. Variations of SEMoses include: incorporating semantic information to the training data (SETrain1-Moses and SETrain2-Moses); adding a Language Model trained on split sentences (Moses-LM, SEMoses-LM, SETrain1-Moses-LM, SETrain2-Moses-LM); and/or exchanging the automatic semantic parsing with manual parsing (SEMoses$^{\mathrm{m}}$, SEMoses$^{\mathrm{m}}$-LM).

Native English speakers were asked to use a 5-point Likert scale (-2 to +2 scores) to measure Structural Simplicity: *"is the output simpler than the input, ignoring the complexity of the words?"*. The judgements from three different annotators are averaged to get the final score for each instance. The authors report a moderate inter-annotator agreement (using Cohen's quadratic weighted κ) of 0.48, measured as the average agreement of the three annotator pairs.

Compared to the Simplicity Gain dataset, this one is bigger (1,750 instances) and offers more variability in the system outputs collected. In addition, Figure 5.2 shows that the distribution of scores spans across all possible values, indicating that some systems even hurt the Structural Simplicity of the original sentence. Despite the over-representation of simplifications with scores between 0 and 0.5, around 32% of instances do improve Structural Simplicity, indicating that an analysis based on perceived quality across different levels is possible.

5.2.3 The New Simplicity-DA Dataset

This dataset is an extension of the one collected as part of ASSET in Sec. 4.5. We used the original sentences from the test set of ASSET, which are the same ones in HSplit and TurkCorpus. This allowed us to collect publicly-available outputs from six automatic Sentence Simplification systems:[5]

- PBMT-R, explained before, a model based on phrase-based statistical MT.
- Hybrid, explained before, combines semantic analysis with phrase-based statistical MT.
- SBMT-SARI (Xu et al., 2016), a syntax-based statistical MT model trained on paraphrases from the Paraphrase Database and tuned using SARI.
- DRESS-LS (Zhang and Lapata, 2017), a neural model that uses an RNN-based encoder-decoder architecture with attention combined with reinforcement learning.
- DMASS-DCSS (Zhao et al., 2018), a neural model that uses the Transformer architecture (Vaswani et al., 2017) and memory-augmentation with paraphrasing rules from the Simple Paraphrase Database (Pavlick and Callison-Burch, 2016).
- ACCESS (Martin et al., 2020), a neural model with a Transformed-based encoder-decoder architecture that conditions the generation of simplifications on explicit desired text attributes (e.g. length and/or dissimilarity with original input).

For each system, we randomly sampled 100 automatic simplifications, not necessarily all from the same set of original sentences, but ensuring that the system output was not exactly the same as the original sentence. Then, we collected human ratings using Amazon Mechanical Turk, following the same methodology described in Sec. 4.5. Workers were asked to assess the quality of the automatic simplifications in three aspects: fluency, meaning

[5] These systems are described in more detail in Sec. 2.4.

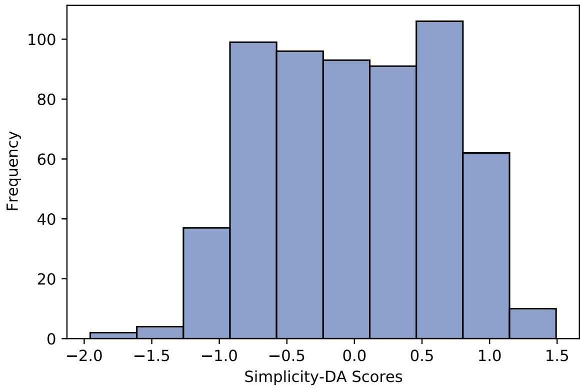


Figure 5.3 Distribution of Simplicity scores in the newly-collected Simplicity-DA dataset.

preservation and simplicity. For each aspect, raters needed to submit a score between 0 and 100, depending on how much they agreed with a specific question.[6] For simplicity, in particular, workers were asked: *Rate your level of agreement to the statement: "The Simplified sentence is easier to understand than the Original sentence"*. This follows the Direct Assessment (DA) methodology (Graham et al., 2017), commonly used in Machine Translation Shared Tasks (Barrault et al., 2019; Bojar et al., 2018). As such, henceforth, we refer to this kind of simplicity judgements as Simplicity-DA.

For each simplification instance, we collected 15 ratings in each aspect. These ratings are then standardised by the mean and standard deviation of each worker to reduce particular biases. The average of all 15 standardised ratings is the final score for the instance in each aspect. We computed inter-annotator agreement (IAA) for each aspect using a similar method to Pavlick and Tetreault (2016): (1) we estimated 5 equally-distributed bins based on all standardised ratings for the particular aspect; (2) for each instance, we simulated two raters by randomly choosing one score as rater A and the average of the others as rater B; (3) we determined the bins corresponding to the scores to discretise them; and (4) we computed Cohen's weighted κ (Cohen, 1968) between raters A and B. This process was repeated 1,000 times, obtaining a mean and variance of 0.533 ± 0.023 for Simplicity-DA, in particular.

The IAA for the collected ratings in our dataset is higher than that for the Structural Simplicity dataset, even though both are considered moderate according to the scale of Landis and Koch (1977). In addition, our dataset is bigger in size and offers more variability of system outputs than the Simplicity Gain dataset. In particular, we include state-of-the-art neural sequence-to-sequence models, the current trend in automatic simplification systems. Furthermore, Figure 5.3 shows that the Simplicity-DA ratings are more diversely-distributed

[6] Details on how workers were selected and the specific questions that were asked are provided in Sec. 4.5.

across all scores values than the other datasets. This benefits our meta-evaluation since one of our intended dimensions of study is the perceived low or high quality (in terms of simplicity) of the system outputs. Overall, we argue that the newly-collected Simplicity-DA dataset is a valid alternative view at human judgements of simplicity, more reliable for analysing automatic metrics in a multi-operation simplification scenario.

5.3 Meta-Evaluation of Automatic Evaluation Metrics

In this Section, we study the behaviour of automatic evaluation metrics at the sentence-level. The datasets described previously are suited for an analysis at this granularity since they contain human judgements for each individual simplification instance. Furthermore, metrics explicitly-developed to measure (some form of) simplicity, such as SARI and SAMSA, operate by-definition at the sentence-level.[7] Our meta-evaluation analyses how the correlations of automatic metrics with human judgements vary under several conditions: different degrees of simplicity of the system outputs, the approaches used by the simplification systems, and the set of manual simplification references used to compute the metrics' scores.

5.3.1 Experimental Setting

We focus our study on metrics that were specifically-developed to estimate the simplicity of system outputs, or that have been traditionally-used for this task, namely:[8] BLEU (Papineni et al., 2002), SARI (Xu et al., 2016), SAMSA (Sulem et al., 2018b), FKGL (Kincaid et al., 1975), FKBLEU (Xu et al., 2016) and iBLEU (Sun and Zhou, 2012).[9] We also experiment with the arithmetic mean (AM) and geometric mean (GM) of BLEU-SARI and SARI-SAMSA.[10]

We also include BERTScore (Zhang et al., 2020), a reference-based metric that computes the similarity between tokens in a system output and tokens in a manual reference, using cosine similarity between their contextual embeddings, namely BERT (Devlin et al., 2019). This allows to capture information from the context without relying on n-grams. BERTScore has shown good results for assessing automatic outputs in Machine Translation and Image Captioning. This metric provides three types of scores: BERTScore$_{\text{recall}}$ matches each token in the reference to its most similar one in the system output, BERTScore$_{\text{precision}}$ does the opposite,

[7] SARI has a corpus-level version (STAR) that is commonly-reported to compare the performances of automatic systems. However, the original authors only validated the correlation of SARI with human judgements at sentence-level, not system-level (`https://github.com/cocoxu/simplification/issues/9`).

[8] Detailed descriptions for all these metrics are provided in Sec. 2.3.2

[9] Even though FKGL is a document-level metric, we include it in our study following Xu et al. (2016).

[10] For completeness, Appendix C includes results with the AM and GM of other combinations of metrics.

and BERTScore$_{\mathrm{F1}}$ is the combination of the two. When multiple references are available, BERTScore compares the system output against all references and returns the highest value.

We used the implementations of these metrics provided by EASSE (Alva-Manchego et al., 2019a). Most of the metrics are sentence-level by definition, with the exception of BLEU (and derivations). In this case, we used a smoothed version with method `floor` and default value 0.0 in SacreBLEU (Post, 2018).[11] For a fair comparison, we detokenised and recased all original sentences and system outputs in the three datasets. Then, we set EASSE to compute all metrics with the same configuration: tokenisation using SacreMoses[12] and case-sensitive calculation.

In order to compare automatic evaluation metrics, we followed the methodology of recent editions of the WMT Metrics Shared Task (Ma et al., 2018, 2019). First, we computed the correlations between automatic scores and human judgements via Pearson's r for each metric. Since the simplicity ratings in our human evaluation datasets are absolute instead of relative rankings between instances, this method is better suited (and easier to apply) than Kendall's Tau. Furthermore, we performed Williams significance tests (Williams, 1959) to determine if the increase in correlation between two metrics is statistically significant or not.

5.3.2 Metrics across Simplicity Quality Levels

Our first dimension of analysis is the perceived quality of the automatic simplifications. We investigate whether it is easier or harder for metrics to evaluate low-quality or high-quality simplifications, as determined by their human judgements on simplicity. In order to do this, we split each dataset into groups of "low" and "high" simplicity, and compute Pearson's r in each subset. Following (Fomicheva and Specia, 2019), we use quantiles to perform this split, which allows us to have the same number of data points in each quality group. For our analysis, we split the instances in each dataset into two groups, and compute the correlations of metrics and human judgements for the top 50% ("High"), the bottom 50% ("Low") and "All" available instances. We refrained from using four groups as in (Fomicheva and Specia, 2019) considering the lower number of data points in each of our datasets compared to those used in their study.

Simplicity-DA

Table 5.1 presents the correlations for each quality split of this dataset, divided into reference-based and non-reference-based metrics. The former were computed using reference simplifica-

[11] https://github.com/mjpost/sacrebleu

[12] https://github.com/alvations/sacremoses

Table 5.1 Pearson correlations between **Simplicity-DA** judgements and metrics scores computed using references from **ASSET**, for **low/high/all quality splits** (N is the number of instances in the split). Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	Low ($N = 300$)	High ($N = 300$)	All ($N = 600$)
Reference-based	BERTScore$_{\text{Precision}}$	0.512	0.287	**0.617**
	BERTScore$_{\text{F1}}$	0.518	0.224	0.573
	iBLEU	0.398	0.253	0.504
	BLEU-SARI (AM)	0.417	0.239	0.503
	BERTScore$_{\text{Recall}}$	0.471	0.172	0.500
	BLEU	0.405	0.235	0.496
	BLEU-SARI (GM)	0.408	0.215	0.476
	SARI	0.336	0.139	0.359
	SARI-SAMSA (AM)	0.203	0.050	0.166
	SARI-SAMSA (GM)	0.222	0.024	0.156
	FKBLEU	0.131	0.006	0.098
Non-Reference-based	FKGL	0.272	0.093	0.117
	SAMSA	0.103	0.010	0.058

tions from ASSET, since the Simplicity-DA judgement is not limited to a particular operation being performed, and simplifications in ASSET were created by applying several of them.

When "All" instances are considered, BERTScore$_{\text{precision}}$ shows a strong correlation and no metric is better than it. Flesch-based metrics (FKGL and FKBLEU) have the lowest correlations, providing further evidence that these type of metrics are unsuitable for sentence-level evaluation. Surprisingly, simplification-specific metrics, SARI and SAMSA, also fair poorly. One possible explanation is that these metrics were developed to only assess the execution of particular simplification operations: lexical paraphrasing and sentence splitting, respectively. Therefore, they do not seem to be able to use simplification references where multiple operations were applied, further supporting the preliminary results from Sec. 4.5. Combining them naively through their arithmetic or geometric means does not yield good correlations in this dataset either. BLEU shows a moderate correlation, and combining it with SARI does not significantly improve the correlation with Simplicity-DA judgements in this dataset.[13]

When comparing the correlations between the "Low" and "High" splits, we can notice that the ones in the latter are much lower. This could be interpreted as: **low scores of some metrics do indicate "bad" quality of a simplification (in terms of Simplicity-DA), but high scores do not necessarily imply "good" quality.** This could be explained by how (most of) the

[13] See Appendix C for plots showing if increases in correlation between metrics are statistically significant.

Table 5.2 Examples of original sentences with some of their simplification references in ASSET, and system outputs with corresponding human and automatic scores from the Simplicity-DA dataset. The reference selected by the automatic metric as most-similar to the system output is underlined.

Original Sentence	In 1998, Culver ran for Iowa Secretary of State and was victorious.
System Output	In 1998, Culver ran for Iowa Secretary of State.
Sample References	Culver ran and won Iowa's secretary of State in 1998.
	In 1998, Culver ran for Iowa Secretary of State. He won the election.
	In 1998, Culver successfully ran for Iowa Secretary of State.
Simplicity-DA	0.551 **BERTScore**$_{\textbf{Precision}}$ 0.984
Original Sentence	Below are some useful links to facilitate your involvement.
System Output	Below is some useful links to help with your involvement
Sample References	Here are good links to get you to do it.
	Here are some useful links to help you.
	Below are some useful links to help with your involvement.
Simplicity-DA	0.327 **BERTScore**$_{\textbf{Precision}}$ 0.934
Original Sentence	He was appointed Companion of Honour (CH) in 1988.
System Output	He was appointed Companion of Honour in 1988.
Sample References	He was made the Companion of Honour (CH) in 1988.
	In 1988 he was chosen as a Companion of Honour.
	He was appointed Companion of Honour in 1988.
Simplicity-DA	0.436 **BERTScore**$_{\textbf{Precision}}$ 1.000

metrics assess the system outputs (i.e. by computing their similarity to the manual references), and by the question used to elicit Simplicity-DA judgements (*Rate your level of agreement to the statement: "The Simplified sentence is easier to understand than the Original sentence"*).

One possible reason is that simplifying a sentence may be limited to a few important changes that improve its readability (e.g. replacing some words or splitting a long sentence into two), whilst keeping the rest of the original sentence as-is. Not performing these key modifications (or unnecessary ones) would be penalised by the human judges, resulting in low Simplicity-DA scores. However, similarity-based metrics could still provide high scores that, in fact, are indicative of the overlap between the system output and the references due to meaning preservation, but not of the changes that improve simplicity. The first example in Table 5.2 illustrates this scenario. The reference selected by BERTScore$_{\text{precision}}$ as the most similar to the system output is a clever simplification that uses the adverb "succesfully" to replace the clause "and was victorious" from in the original sentence. Since the rest of the sentence is unchanged, it has a high overlap with the system output that merely deleted the "and was victorious" clause. This overlap is rewarded by the automatic metric with a high score, but the human judges

Table 5.3 Pearson correlations between **Simplicity Gain** judgements and metrics scores computed using references from **TurkCorpus**, for **low/high/all quality splits** (N is the number of instances in the split). Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	Low ($N = 186$)	High ($N = 186$)	All ($N = 372$)
Reference-based	SARI	0.292	0.240	**0.331**
	BERTScore$_{\text{F1}}$	0.215	0.236	0.247
	BERTScore$_{\text{Recall}}$	0.221	0.217	0.241
	BERTScore$_{\text{Precision}}$	0.209	0.231	0.241
	BLEU-SARI (GM)	0.246	0.177	0.214
	BLEU-SARI (AM)	0.223	0.172	0.187
	iBLEU	0.181	0.136	0.128
	BLEU	0.178	0.132	0.123
	FKBLEU	0.041	0.007	0.092
Non-Reference-based	FKGL	0.045	0.101	0.147
	SAMSA	0.120	0.042	0.013

did not consider this compression sufficient for improved simplicity, perhaps because it also removed important information needed to understand the main idea of the original sentence.

Another possible reason is a disagreement between the changes the human judges deemed necessary to give a good Simplicity-DA score, and what the editors that created ASSET considered as valid simplifications. The second and third examples in Table 5.2 illustrate this scenario. The selected references are almost exactly the same as the corresponding system outputs, and therefore BERTScore$_{\text{precision}}$ gave them very high scores. However, the human judges did not consider that the changes performed were sufficient to grant a high value of Simplicity-DA for improved simplicity. This may not be indicative that references in ASSET are incorrect, but rather that not all of them have the same degree of simplicity. Therefore, as a way to improve automatic evaluation, it could be useful to enrich the manual references with human judgements on their simplicity level. In this way, an automatic score would not be only based on the similarity to a reference, but also on the potential level of simplicity that the system output could achieve if it were an exact match with that particular reference.

Simplicity Gain

Table 5.3 presents the correlations in each quality split of this dataset, for reference-based and non-reference-based metrics. The former were computed using reference simplifications from TurkCorpus, since the Simplicity Gain judgement is limited to counting lexical paraphrases, and simplifications in TurkCorpus were created by only applying that same operation.

In this dataset, SARI has a moderate correlation, and the highest among all metrics when "All" evaluation instances are considered, similar to the results in (Xu et al., 2016). Just like in the Simplicity-DA dataset, Flesch-based metrics and SAMSA show low correlations, whilst BLEU (and its derivations) are in the middle of group. The different versions of BERTScore are second-best, and have similar performances, i.e. there is no statistically-significant difference between them.[14] Also, combining SARI with BLEU does not improve its individual correlation. When comparing the correlations between the "Low" and "High" quality splits, most metrics have lower Person's *r* in "High". However, this is not a consistent behaviour, and the differences are not as considerable as observed in the Simplicity-DA dataset.

A possible reason for the overall moderate-to-low correlations is that what (most of) the metrics are trying to measure does not correspond with the definition of Simplicity Gain. Almost all metrics compute the similarity between the system output and the references. However, measuring Simplicity Gain implies identifying the changes made by the system, and then verifying that they are correct. In order to do this, it is necessary to take the original sentence into consideration, and not just the system output and the references. SARI is the only metric that attempts to follow this logic, by computing the correctness of the n-grams kept, deleted and added. However, lexical paraphrasing is strongly related to performing replacements, an operation that SARI does not directly identify and measure. The examples in Table 5.4 show how this limitation hurts the metric: whilst in the second instance there are less correct replacements than in the first one, the SARI score is higher. By not directly counting correct replacements, the metric is affected by the conservative nature of the outputs and references that copied most of the original sentences. Then, it is the correctness of kept and (not) deleted n-grams what contributes to get a high score. Consequently, SARI is not measuring Simplicity Gain, explaining why the correlation with human judgements is barely moderate.[15]

The concept of Simplicity Gain is easy to understand: it is the number of correct changes. If metrics were able to measure it accurately, automatic scores would be more straightforward to interpret, facilitating the comparison of simplifications generated by different systems. However, future analyses on the correlation between metrics and human judgements of this kind require collecting more and better data. The Simplicity Gain dataset from Xu et al. (2016) that we use in this study is quite small, with only 372 evaluated instances. In addition, automatic simplifications from only four systems were included, three of which are of similar

[14] See Appendix C for more details.

[15] It could be argued that the examples in Table 5.4 are cherry-picked and do not signal a trend. We believe that the low correlation already reflects that SARI and Simplicity Gain are not measuring the same phenomenon. The examples are just an attempt to explain why this happens. A more in-depth analysis is left for future work.

Table 5.4 Examples of original sentences and system outputs with corresponding human and automatic scores from the Simplicity Gain dataset. Changes related to lexical paraphrasing are boldfaced.

Original Sentence	Jeddah is the **principal** gateway to Mecca, Islam's holiest city, which able-bodied Muslims **are required to** visit at least once in their **lifetime**.
System Output	Jeddah is the **main** gateway to Mecca, Islam's holiest city, which sound Muslims **must** visit at least once in **life**.
Simplicity Gain	1.83 **SARI** 0.462
Original Sentence	The Great Dark Spot is thought to **represent** a hole in the methane cloud deck of Neptune.
System Output	The Great Dark Spot is thought to **be** a hole in the methane cloud deck of Neptune.
Simplicity Gain	1.25 **SARI** 0.587

characteristics (SBMT-based), and without state-of-the-art neural models. All of this impedes generalisations that could be relevant in current Sentence Simplification research.

Structural Simplicity

Table 5.5 presents the correlations in each quality split of this dataset, for reference-based and non-reference-based metrics. The former were computed using reference simplifications from HSplit, since the Structural Simplicity judgement is limited to qualifying sentence splitting, and simplifications in HSplit were created by only applying that operation.

In this dataset, most metrics have moderate correlations with human judgements when "All" evaluated instances are used. BLEU obtains the highest correlation, but its not the best overall since its differences with BLEU-SARI (GM) and BERTScore$_{\text{Recall}}$ are not statistically significant (see details in Appendix C). This would seem to contradict the findings of Sulem et al. (2018a), who argue that BLEU does not correlate well with Structural Simplicity. However, as will be shown in the next Section, the magnitude of the correlation depends on the approach of the systems included in the study. Whilst Sulem et al. (2018a) only used models tailored for sentence splitting to reach that conclusion, in this first analysis we are using all available system outputs in the dataset. Furthermore, the low correlation of SAMSA is surprising, since this metric was specifically-designed to evaluate sentence splitting, and it showed better performance in the dataset of (Sulem et al., 2018b). However, they measured the correlation at the system-level, whilst we are analysing it at the sentence-level. Finally, BERTScore$_{\text{Precision}}$, the best metric in the Simplicity-DA dataset, has the poorest correlation in the "All" data split. From previous results, we know that BERTScore$_{\text{Precision}}$ is very good at measuring the

Table 5.5 Pearson correlations between **Structural Simplicity** human judgements and automatic metrics scores computed using manual references from **HSplit**, for **low/high/all quality splits** (N is the number of instances in the split). Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	Low ($N = 875$)	High ($N = 875$)	All ($N = 1,750$)
Reference-based	BLEU	0.421	**0.643**	0.443
	BLEU-SARI (GM)	0.329	0.589	0.438
	iBLEU	0.408	0.635	0.436
	BLEU-SARI (AM)	0.346	0.599	0.431
	BERTScore$_{\text{Recall}}$	0.411	0.601	0.430
	BLEU-SAMSA (AM)	0.289	0.608	0.420
	BLEU-SAMSA (GM)	0.293	0.569	0.370
	FKBLEU	0.395	0.608	0.364
	BERTScore$_{\text{F1}}$	0.483	0.529	0.325
	SARI	0.137	0.418	0.313
	BERTScore$_{\text{Precision}}$	**0.552**	0.310	0.090
Non-Reference-based	SAMSA	0.103	0.431	0.284
	FKGL	0.070	0.165	0.228

similarity between a system output and a reference. As such, its low correlation would indicate that simple similarity matching is not enough to measure Structural Simplicity.

When comparing the correlations between the “Low” and “High” splits, we can notice that the ones in the former are much lower for all metrics but BERTScore$_{\text{Precision}}$. In fact, this metric has the highest correlation in the “Low” split, with a substantial increase over its own correlation in the “All” data split. This could also be explained by our previous argument. A low score in Structural Simplicity implies that the system output does not contain any sentence splitting, or that the changes made (if any) are not structural. In these situations, BERTScore$_{\text{Precision}}$ would not be able to match a reference in HSplit, since they most likely contain only sentence splitting. In turn, the metric returns a low score that correlates well with a low human judgement.

We further analyse the behaviour of SAMSA, a metric specifically-designed to evaluate Structural Simplicity. Recall that SAMSA first uses a semantic parser to identify the Scenes in the original sentence, and a syntactic parser to identify the sentence splits in the system output. Then, it counts how many of the words corresponding to the Participants of each Scene align with words in each sentence split. Ideally, all Participants of a single Scene should appear in a single sentence split. The first example in Table 5.6 illustrates a case where this logic may be problematic. SAMSA identifies that there is only one Scene in the original sentence and only one sentence split in the system output. Since both sentences are exactly the same,

Table 5.6 Examples of original sentences and system outputs with corresponding human and automatic SAMSA scores from the Structural Simplicity dataset.

Original Sentence	Orton and his wife welcomed Alanna Marie Orton on July 12 2008.
System Output	Orton and his wife welcomed Alanna Marie Orton on July 12 2008.
Structural Simplicity	0.00 **SAMSA** 1.00
Original Sentence	Graham attended Wheaton College from 1939 to 1943, when he graduated with a BA in anthropology.
System Output	Graham attended Wheaton College from 1939 to 1943. He graduated with a BA in anthropology.
Structural Simplicity	0.33 **SAMSA** 1.00
Original Sentence	Jeddah is the principal gateway to Mecca, Islam's holiest city, which able-bodied Muslims are required to visit at least once in their lifetime.
System Output	Jeddah is the principal gateway to Mecca.
Structural Simplicity	2.00 **SAMSA** 0.14

the word alignment is perfect and SAMSA gives the simplification the highest score possible. However, the human judges noticed that no changes were performed and, therefore, they gave the instance a score of 0. On the one hand, this could suggest that SAMSA should only be used when sentence splitting was actually performed in the simplification instance. On the other hand, it could be argued that the original sentence was already structurally simple, and that no splitting was necessary, making the human score of 0 unfair. This points out to possible issues in the data collection, and that perhaps using a -2 to +2 scale is unsuitable for these scenarios.

We further explore our last argument of potential incompatibilities between what Structural Simplicity should measure, and what the human judges qualified as such. The second and third examples in Table 5.6 suggest that there are indeed problems. The second example shows that a perfectly-reasonable and correct splitting (with a SAMSA score of 1.0) received a low score from the judges. More worryingly, the third example presents a sentence where no splitting was performed (and with substantial compression) that received the highest score for Structural Simplicity. This could indicate that the raters did not consider sentence splitting as the only mechanism for improving the simplicity of the structure of a sentence.

In an attempt to quantify this phenomenon, Figure 5.4 presents the distribution of Structural Simplicity scores for instances where sentence splitting was performed and where it was not. Instances with attempted splitting only amount to 17% (306/1,750) of the total of instances in the dataset. Whilst this is a low quantity, their human scores span along all possible values for Structural Simplicity. It is encouraging that most instances where no splitting was attempted received a human score close to 0. However, as pointed out previously, there are many that were judged by humans with high values of Structural Simplicity. We could hypothesise that this is

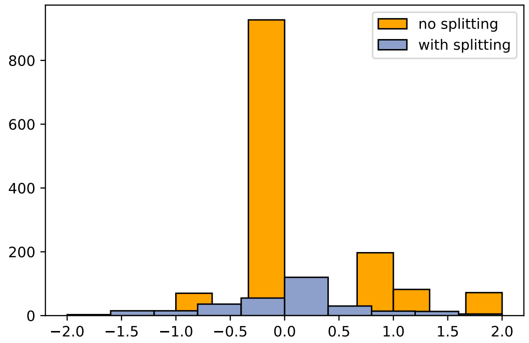


Figure 5.4 Distribution of Structural Simplicity scores in the dataset of (Sulem et al., 2018c) for instances with and without sentence splitting in the system output.

caused by a misunderstanding of the rating instructions, or by that fact that there are operations beyond sentence splitting that can also improve the structural simplicity of a sentence.

(Improvement in) Structural Simplicity is a relevant feature to evaluate in automatic simplifications. Isolating its assessment both manually and through metrics can contribute to a more fine-grained analysis of the performance of automatic systems. However, it is important to establish adequate quality control mechanisms that ensure the trustworthiness of the collected data, so that we can develop metrics that accurately resemble the intended human judgements.

5.3.3 Metrics across Types of Systems

We now investigate whether the correlations of the automatic scores are affected by the type of system that generated the simplifications. In order to do this, we classified the systems in the datasets into: phrase-based MT (PBMT), syntax-based MT (SBMT), neural sequence-to-sequence (S2S), and semantics-informed rules (Sem) by themselves or coupled with one of the previous types (i.e. Sem+PBMT, Sem+S2S). For this study, we do not use the Simplicity Gain dataset since it only contains simplifications produced by PBMT and SBMT systems, thus not providing enough variability to study the effect of various system types.

Simplicity-DA

Table 5.7 presents the correlations of each metric for the different system types in this dataset, with reference-based metrics computed using simplifications from ASSET. BERTScore$_{\text{precision}}$ achieves the highest correlations in all groups, and for S2S and Sem+PBMT models, in particular, no other metric is statistically-equal. Most metrics show higher correlations in the S2S group than in others. However, because the number of data points is smaller in the latter,

Table 5.7 Pearson correlations between **Simplicity-DA** human judgements and automatic metrics scores computed using references from **ASSET**, for **splits based on system type** (N is the number of instances in the split). Correlations of metrics not significantly outperformed by any other in the system type split are boldfaced. Metrics are grouped in Reference-based (top) and Non-Reference-based (bottom).

Metric	SBMT ($N = 100$)	PBMT ($N = 100$)	S2S ($N = 300$)	Sem+PBMT ($N = 100$)
BERTScore$_{\text{Precision}}$	0.537	0.459	**0.650**	**0.624**
BERTScore$_{\text{F1}}$	0.528	0.400	0.588	0.568
iBLEU	0.315	0.336	0.536	0.335
BERTScore$_{\text{Recall}}$	0.527	0.375	0.484	0.470
BLEU	0.295	0.347	0.546	0.333
BLEU-SARI (GM)	0.298	0.320	0.508	0.308
SARI	0.228	0.173	0.310	0.240
SARI-SAMSA (AM)	0.243	0.121	0.209	0.291
SARI-SAMSA (GM)	0.250	0.080	0.190	0.333
FKBLEU	0.006	0.058	0.092	0.138
FKGL	0.055	0.063	0.104	0.062
SAMSA	0.184	0.067	0.126	0.248

stronger conclusions cannot be formulated. Overall, since the current trend is to develop S2S models, it is encouraging that modern metrics are capable of evaluating them, but keeping in mind the nuances we signalled in the previous Section regarding quality levels.

Structural Simplicity

Table 5.8 presents the correlations of each metric in the different system type groups in this dataset. Reference-based metrics were computed using manual references from HSplit. All metrics achieve their highest correlations in the S2S group, except for BERTScore$_{\text{precision}}$. As presented in the previous Section, this metric is particularly good at judging instances with low Structural Simplicity, which seem to be those from the PBMT and SBMT groups, mainly.

In the previous Section, we observed that BLEU had high correlation with high-scoring quality judgements (in terms of Structural Simplicity). Here, we notice that that behaviour is limited to simplifications produced by systems in the S2S and Sem+S2S groups. This appears to contradict the observations of Sulem et al. (2018a), who used this same dataset to conclude that BLEU is a bad estimator of Structural Simplicity. The reason behind this disagreement is that for their sentence-level study "HSplit as Reference Setting", the systems they chose were: DSS, DSS$^{\text{m}}$, SEMoses, SEMoses$^{\text{m}}$, SEMoses-LM, and SEMoses$^{\text{m}}$-LM. That is, systems within the Sem and Sem+PBMT groups, for which BLEU, indeed, shows poor correlations. Perhaps a

Table 5.8 Pearson correlations between **Structural Simplicity** judgements and automatic metrics scores computed using references from **HSplit**, for **splits based on system type** (N is the number of instances in the split). Correlations of metrics not significantly outperformed by any other in the system type split are boldfaced. Metrics are grouped in Reference-based (top) and Non-Reference-based (bottom).

Metric	PBMT ($N = 70$)	SBMT ($N = 70$)	S2S ($N = 280$)	Sem ($N = 140$)	Sem+PBMT ($N = 630$)	Sem+S2S ($N = 560$)
BLEU	0.284	0.380	0.661	0.130	0.147	0.540
BLEU-SARI (GM)	0.157	0.341	0.589	0.097	0.185	0.515
iBLEU	0.252	0.380	0.642	0.130	0.145	0.536
BLEU-SARI (AM)	0.184	0.364	0.603	0.100	0.175	0.507
BERTScore$_{\text{Recall}}$	0.339	0.418	0.635	0.066	0.134	0.480
BLEU-SAMSA (AM)	0.240	0.334	0.603	0.095	0.072	0.573
BLEU-SAMSA (GM)	0.216	0.279	0.563	0.109	0.075	0.561
FKBLEU	0.215	0.344	0.617	0.009	0.119	0.539
BERTScore$_{\text{F1}}$	0.405	0.497	0.553	0.180	0.049	0.362
SARI	0.015	0.286	0.330	0.028	0.166	0.355
BERTScore$_{\text{Precision}}$	**0.501**	**0.571**	0.292	**0.330**	0.096	0.111
SAMSA	0.141	0.177	0.368	0.052	0.009	0.497
FKGL	0.205	0.016	0.251	0.083	0.155	0.242

reason they chose this type of setup is explained by Figure 5.5. Whilst S2S and Sem+S2S have more instances that were scored with good Structural Simplicity, these groups contain very few system outputs where sentence splitting was attempted. Therefore, we believe that Sulem et al. (2018a)'s conclusion should be more nuanced: *BLEU is a bad metric to estimate Structural Simplicity in system outputs where sentence splitting was attempted.*

Nevertheless, not considering system outputs in the S2S and Sem+S2S groups reduces the future impact of the previous statement, since the current trend in Sentence Simplification research is developing that type of models. For their system-level study "Standard Reference Setting", Sulem et al. (2018a) did include systems from the S2S group, but computed BLEU using references from Simple Wikipedia and TurkCorpus, which are not focused on sentence splitting. We believe that this experimental setting is unfair to BLEU, and that more cautious analysis should be performed to determine if a metric should (or should not) be used to assess Structural Simplicity in current S2S models.

5.3.4 Effect of Simplification References

The third dimension of analysis for our meta-evaluation is the set of simplification references used to compute automatic evaluation scores. Since there can be multiple correct simplifications

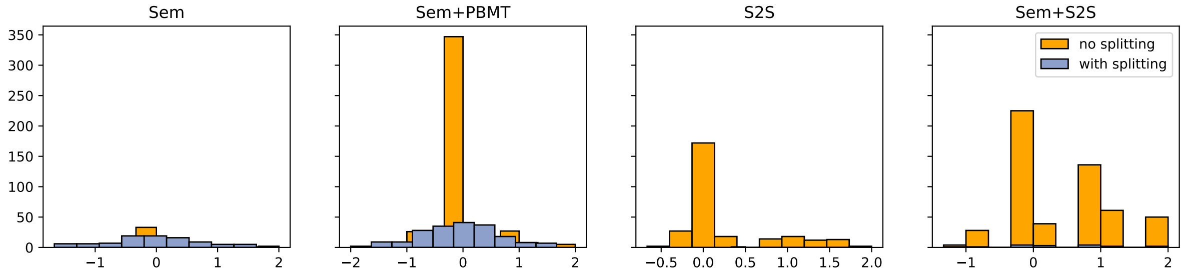


Figure 5.5 Distribution of Structural Simplicity scores in the dataset of (Sulem et al., 2018c) for instances with and without sentence splitting in the system output and for each system type.

for the same original sentence, it is possible that a reference-based metric becomes more reliable if it has access to more manual references for comparison. It is worth remembering that whilst BLEU and SARI take all references for each original sentences into account when computing their scores, BERTScore takes one at a time and returns the maximum score. In this Section, we investigate whether the correlations of reference-based metrics vary depending on using all available simplification references or particular subsets of them. We only experiment with the Simplicity-DA dataset because its simplicity judgements are not tied to performing a specific type of simplification operation, as is the case for the other datasets. Thus, having a more varied set of references could be advantageous for reference-based metrics in this scenario. In addition, we take advantage of the fact that the original sentences in the Simplicitiy-DA dataset have corresponding manual simplifications in three multi-reference datasets: ASSET (10 references), TurkCorpus (8 references) and HSplit (4 references). Recall that the manual simplifications in each dataset were produced via different operations: lexical paraphrasing in TurkCorpus; sentence splitting in HSplit; and lexical paraphrasing, compression, and sentence splitting in ASSET.

ASSET vs All References

Table 5.9 presents the correlations of each reference-based metric computed using either (1) only the 10 manual reference simplification from ASSET; or (2) the simplifications from ASSET (10 references) merged with those from TurkCorpus (8 references) and HSplit (4 references), that is “All References” (22). We further divide this analysis into “Low”, “High” and “All” quality splits as in a previous Section.[16] When using “All” instances, most metrics have a slight increase in their Pearson’s r when All References are used, with BERTScore$_{\text{Precision}}$ achieving the highest correlations, and being statistically superior to every other metric. This

[16] We do not add the system type dimension since the number of instances in each subgroup would be too small to allow drawing strong conclusions.

Table 5.9 Pearson correlations between **Simplicity-DA** judgements and reference-based metrics scores **grouped by the set of manual references used**. Within each group, we divide the data into Low/High-/All quality splits. Correlations of metrics not significantly outperformed by any other in their group and quality split are boldfaced.

Metric	ASSET			All References			Selected References		
	Low	High	All	Low	High	All	Low	High	All
BERTScore$_{\text{Precision}}$	0.512	0.287	**0.617**	0.541	0.280	**0.629**	0.543	0.276	**0.635**
BERTScore$_{\text{F1}}$	0.518	0.224	0.573	0.530	0.202	0.576	0.534	0.202	0.584
iBLEU	0.398	0.253	0.504	0.398	0.250	0.537	0.396	0.244	0.536
BLEU	0.405	0.235	0.496	0.404	0.230	0.526	0.402	0.223	0.525
BLEU-SARI (AM)	0.417	0.239	0.503	0.418	0.218	0.519	0.418	0.221	0.523
BERTScore$_{\text{Recall}}$	0.471	0.172	0.500	0.476	0.165	0.506	0.479	0.165	0.511
BLEU-SARI (GM)	0.408	0.215	0.476	0.410	0.195	0.490	0.410	0.205	0.496
SARI	0.336	0.139	0.359	0.366	0.097	0.353	0.352	0.115	0.350

improvement seems to be caused by better detection of “Low” quality simplifications. If fact, using All References slightly affects BERTScore$_{\text{Precision}}$ (and most metrics) in its ability to detect system outputs of “High” Simplicity-DA. As in a previous Section, we hypothesise that this is caused by the different degrees of simplicity that each manual reference has in each dataset. By having more references available, BERTScore$_{\text{Precision}}$ is more-likely to match one with a system output, and then return a high score. However, high-similarity with a reference does not necessarily mean high improvements in simplicity, since the manual reference could correspond to a valid simplification but with a relatively-low degree of simplicity.

ASSET vs Selected References

In the previous analysis, we changed the set of references for all sentences that are being assessed at the same time. We now analyse the effect of changing the set of references for each individual sentence. More concretely, we devise an experiment where, for each automatically-simplified sentence, reference-based metrics compare it to a subset of all available references based on the simplification operations that were attempted. We identify three possible cases: (1) the system attempted only sentence splitting; (2) the system attempted only lexical paraphrasing and/or deletion; (3) the system attempted another possible combination of operations.[17] Depending on the case, a different set of references would be used: HSplit for (1), TurkCorpus and ASSET for (2), and ASSET for (3). We use the sentence-level annotation

[17] There is one more possible case: (4) the system did not attempt any operation (i.e. the original sentence and the system output are the same). However, by design, no such instances are present in the Simplicity-DA dataset.

algorithms from Chapter 3 to identify the attempted operations. Column "Selected References" in Table 5.9 presents the correlations of reference-based metrics computed following the previous logic. All metrics but SARI improve their correlations when instances of "All" qualities are used. As before, this is caused by better detection of "Low" quality simplifications.

5.4 Recommendations for Automatic Evaluation

Our meta-evaluation has allowed us to better understand the behaviour of traditional and more modern metrics for assessing automatic simplifications. Based on those findings, in this Section we a set a list of recommendations related to the present and future of automatic evaluation of Sentence Simplification systems.

5.4.1 Evaluation of Current Simplification Systems

Which automatic metric(s) should be used?

It is difficult to determine an overall "best" metric across all types of simplicity judgements. For Simplicity-DA, BERTScore$_{\text{precision}}$ achieved the highest correlations in all dimensions of analysis. For Simplicity Gain, SARI is better than all BERTScore variants, but that difference is not statistically significant when assessing low and high quality simplifications separately. In addition, there is not enough data to determine if that behaviour translates to modern sequence-to-sequence models. The comparison is even less clear for Structural Simplicity, since the correlations are heavily dependent on the system type or, rather, on evaluating simplifications where sentence splitting was actually attempted, instances of which are insufficient in the dataset used. SAMSA was specifically-developed for this type of simplicity evaluation, and manual inspection suggests that it is doing what it was designed for. As such, even though our analysis does not seem to support its use, we argue that this is caused by the lack of adequate data with judgements on Structural Simplicity.

Overall, we suggest to use multiple metrics, and mainly BERTScore$_{\text{precision}}$ for reference-based evaluation. SARI could be used when the simplification system only executed lexical paraphrasing, whilst SAMSA may be useful when it is guaranteed that splitting was performed.

Which simplification references should the metric(s) compare against?

Simplifications in ASSET are well suited for reference-based evaluation. Incorporating references from TurkCorpus and HSplit seems to only slightly improve the correlations. In addition, it appears that selecting which references to use for each sentence individually benefits the computation of metrics. However, for both cases, the improvements are limited to evaluation of low-quality simplifications.

How should the automatic scores be interpreted?

For Simplicity-DA, low automatic scores of most metrics appear to be good estimators of low quality, whilst high scores do not necessarilly suggest high quality. This indicates that metrics could be more useful for development stages of simplification models. Following the recommendation of using multiple metrics, we suggest to use BERTScore$_{\text{precision}}$ to get a first evaluation. If the score is low, then it signals that the quality of the output is also low. However, when the score is high, it is important to look at other metrics, such as SARI, to verify if the overall quality is good. However, for final arguments on the superiority of one system over another, human evaluation should be preferred.

For Simplicity Gain, metrics' correlations are low to moderate in general, so it is unclear if they are actually measuring this type of human judgement. In the case of Structural Simplicity, inconsistencies in the human judgements (i.e. high scores for instances where no splitting was attempted) hinders the interpretation of results.

5.4.2 Development of New Metrics

Considering the advantages and disadvantages of current metrics, as well as the problems identified in the data used for evaluating them, we provide some suggestions for the development of new resources for automatic evaluation.

What type of human judgements should be collected?

We experimented with crowdsourcing simplicity judgements following the methodology of Direct Assessment that has been successful in Machine Translation research. We believe that submitting continuous scores on how much simpler a system output is over the original sentence, gives raters more flexibility on their judgements, and facilitates subsequent analyses. However, whilst the type of score collected (continuous or discrete) influences the ratings, it is even more important to ensure that raters submit judgements that follow the kind of simplicity that is intended to be measured. As such, it is paramount to train raters before they perform the actual task, and establish quality control mechanisms through out the data collection process. As of the kind of simplicity judgement to elicit, both Simplicity Gain and Structural Simplicity have advantages over requesting absolute simplicity scores. Therefore, we recommend to collect more human judgements based on them, using modern simplification models and simplification instances with adequate characteristics for what we are trying to evaluate.

What characteristics should new metrics have?

For Simplicity-DA, Simplicity Gain and Structural Simplicity, raters were asked compare the automatic simplification to the original sentence, and then submit a particular kind of judgement. Therefore, if humans submit evaluations taking both the original sentence and the simplification into consideration, then we should expect that automatic metrics do so too. Both SARI and SAMSA follow this logic, and we would expect that new metrics take that idea even further. For example, by replacing n-gram matching and syntax-based word alignments, respectively, by similarity of contextual word embeddings, as is done in BERTScore.

Furthermore, we have explained that not every manual simplification in multi-reference datasets (i.e. ASSET, TurkCorpus and HSplit) have the same simplicity level. If references were enriched with this type of information, similarity-based metrics would be able to provide more accurate scores, perhaps combining how similar the system output is to a reference with the simplicity level that could be achieved.

How to ensure that new metrics work?

Our meta-evaluation has shown that different factors influence the correlation of human judgements with automatic scores: perceived quality level, system type, and set of references used for computation. As such, new automatic metrics should not only be evaluated on their absolute overall correlation. It is important to also analyse the reasons behind that value considering the different factors that could be affecting it. In this way, we can determine in which situations the new metrics prove more advantageous than others.

5.5 Summary and Final Remarks

In this Chapter, we attempted to answer the research question: **Can current evaluation metrics measure the ability of automatic systems to perform multi-operation simplifications?**

We collected a **new dataset for evaluation of automatic metrics**. Following the same methodology as in Sec. 4.5, we used Direct Assessment (Graham et al., 2017) to crowdsource human ratings on fluency, meaning preservation and simplicity. The dataset consists of **600 automatic simplifications generated by six different systems**, three of which are based on modern neural sequence-to-sequence architectures. This makes it bigger and more varied than the Simplicity Gain dataset (Xu et al., 2016). In addition, our data collection process established adequate quality control mechanisms to ensure higher reliability and consistency of the human ratings. We collected **15 ratings per simplification instance** to increase inter-annotator agreement, which contrasts with the Simplicity Gain dataset that has five raters, and the Structural Simplicity dataset (Sulem et al., 2018c) that only has 3. Our data collection process can be fine-tuned, and more system outputs should be included. However, our dataset's features are sufficient to offer an alternative view at simplicity judgements over system outputs.

We used our newly-collected dataset (Simplicity-DA), together with the Simplicity Gain and Structural Simplicity datasets to conduct, to the best of our knowledge, **the first meta-evaluation study of automatic metrics in Sentence Simplification**. We analysed the variations of the correlations of sentence-level metrics with human judgements along three dimensions: the perceived simplicity level, the system type, and the set of references used to compute the automatic scores. For the first dimension, we found that **metrics can more reliably score low-quality simplifications** in terms of Simplicity-DA, whilst this effect is not apparent in Simplicity Gain and no strong conclusions could be drawn for Structural Simplicity due to inconsistencies in the ratings. For the second dimension, **correlations do change based on the system type**. In the Simplicity-DA dataset, most metrics are better at scoring system outputs from neural sequence-to-sequence models. Whilst this difference in correlation is more significant in the Structural Simplicity dataset, it seems to be caused by low representation of sentence splitting in the data, rather than differences in system type. For the third dimension, **combining all multi-reference datasets does not significantly improve metrics' correlations over using only ASSET** in the Simplicity-DA dataset. Further analyses on the diversity of the manual references across ASSET, TurkCorpus and HSplit should be performed in order to explain this result. In addition, preliminary experiments on per-sentence reference selection based on the attempted operations showed promising results.

Based on the findings of our meta-evaluation, we designed a set of **guidelines for automatic evaluation of current simplification models**. In particular for multi-operation simplifications, we suggest to use BERTScore with references from ASSET during the development stage of simplification models, and manual evaluation for final comparisons. The main reason is that BERTScore is very good at identifying references that are similar to a system output. However, since not all references have the same simplicity level, a high similarity with a reference does not necessarily indicate high (improvements in) simplicity. Finally, we proposed a **desiderata for the characteristics of new resources for automatic evaluation**. For instance: (1) to further collect Simplicity Gain and Structural Simplicity ratings with better quality controls and diversity of system outputs; (2) to develop metrics take both the original sentence and the automatic simplification into consideration, (3) that manual references be enriched with their simplicity level; and (4) that new metrics be evaluated along several dimensions and not just overall absolute correlations with human ratings of (some form) of simplicity.

All the previously-described experiments and findings allow us to suggest an answer to our research question: Referenced-based metrics can potentially estimate the quality of automatic simplifications provided that multi-operation multi-reference evaluation datasets are available. However, the automatic scores are not reliable for all types of systems and across levels of quality, particularly due to each reference having a different level of simplicity.

Chapter 6

MulTSS: A Multi-Operation Simplifier

When simplifying sentences, professional editors apply several text rewriting operations simultaneously (Sec. 3.1), and human evaluators recognise that some sentences require multi-operation simplifications (Sec. 4.4). However, standard neural sequence-to-sequence models are prone to being conservative (Alva-Manchego et al., 2017), with limited lexical paraphrasing and compression capabilities (Sec. 3.3.2), and it is unclear if they can perform meaningful structural changes that improve the simplicity of the original sentence (Sec. 5.3). One possible cause is the data used for training these models. Most of them learn from automatic sentence alignments between related articles in English Wikipedia (EW) and Simple EW, such as WikiLarge (Zhang and Lapata, 2017). These simplification instances are known to be noisy and display limited types of rewriting (Xu et al., 2015). As such, in this Chapter, we investigate the following research question: **Can we devise models that leverage data with more types of rewriting to enhance the multi-operation capabilities of simplification systems?** We propose MulTSS, a multi-task learning approach for multi-operation Sentence Simplification.

First, we motivate our proposed multi-task approach based on its success for multilingual Machine Translation and previous applications in Sentence Simplification (Sec. 6.1). Then, we describe MulTSS, our multi-task model that leverages data from related text rewriting tasks (split-and-rephrase, extractive compression and lexical paraphrasing) to enhance the variability of operations in current training datasets for Sentence Simplification (Sec. 6.2). After detailing our experimental settings (Sec. 6.3), we show that MulTSS can simplify (our main task) using multiple operations, with higher scores in automatic metrics than strong baselines and previous work (Sec. 6.4). In addition, we present an ablation study to analyse the contribution of each auxiliary task; and also show that MulTSS can achieve competitive results on them (Sec. 6.5). Finally, we summarise our mains findings, and suggest ways to improve our current approach based on its promising results and identified limitations (Sec. 6.6).

6.1 Motivation

Multi-Task Learning (MTL) frameworks for neural sequence-to-sequence architectures were initially developed for Multilingual Machine Translation (MT). Based on the standard RNN-based encoder-decoder model for Neural MT (Bahdanau et al., 2014), Dong et al. (2015) proposed to translate from one source language to multiple target languages by having a model with a shared source encoder and multiple target decoders. This was called a *one-to-many* approach by Luong et al. (2016), who also proposed *many-to-one* (shared decoder) and *many-to-many* (multiple encoders and decoders) approaches for combining MT with other NLP tasks that could benefit it (e.g. parsing and image caption generation). Whilst several other models worked with similar ideas of sharing encoders and decoders (Firat et al., 2016; Lee et al., 2017; Zoph and Knight, 2016), a simpler modification was proposed by Johnson et al. (2017): to prepend an artificial target-language token to the source sentences, and have all languages share all parameters. Incorporating this idea into the Google's NMT system (Wu et al., 2016) was beneficial for low-resource languages, and allowed zero-shot translation, that is translating between language pairs without explicit parallel data during training.

Adding source-side tokens to the training data of NMT models had already been suggested as a way to mark *side constraints* that control the level of politeness (Sennrich et al., 2016) or active/passive voice (Yamagishi et al., 2016) of the translation. It has been later used for domain adaptation (Chu et al., 2017), improving morphological agreement of gender (Vanmassenhove et al., 2018), generating more sentiment-aware translations (Si et al., 2019), to control the readability (Marchisio et al., 2019) or complexity (Agrawal and Carpuat, 2019) of the translation output, among others. Other sequence-to-sequence tasks have also exploited this idea, such as Grammatical Error Correction (Hotate et al., 2019) and Abstractive Summarisation (Fan et al., 2018). For Sentence Simplification, it has been used to control for expected features of the simplification, such as its reading grade level (Scarton et al., 2020; Scarton and Specia, 2018), or its length and dissimilarity with the original sentence (Martin et al., 2020).

Similar to previous approaches, we enrich the training data with source-side tokens to mark a characteristic of the target simplification. However, our objective is not to control the grade level or a specific text feature of the output. Instead, we take advantage of this MTL framework to enhance Sentence Simplification training data by combining it with those from related text rewriting tasks, in order to boost the multi-operation capabilities of a model trained in this fashion. Guo et al. (2018) also proposed to use MTL for Sentence Simplification, with sentence paraphrasing and entailment as auxiliary tasks. However, our approach is simpler, since it does not modify the neural architecture; and we select auxiliary tasks that are more likely to help with performing changes to the input sentence that improve its simplicity.

6.2 Multi-Task Approach

Table 6.1 Examples of main and auxiliary tasks selected for training MulTSS. Extracted from the annotation guidelines of ASSET (Alva-Manchego et al., 2020a).

Task	Original Sentence	Rewritten Sentence
Simplification	If you are under ~~the age of~~ 18, you **are required to** complete ~~at least~~ 65 hours of **behind-the-wheel skill-building including** 10 hours of **nighttime driving**.	If you are under 18, you **must** complete 65 hours of **practice driving. This must include** at least 10 hours of **driving at night**.
Lexical Paraphrasing	From **its inception**, it was **designated** a duty-free port **and vied** with the neighboring Sultanate of Pattani for trade.	From **the start**, it was **chosen to be** a duty-free port **to compete** with the neighboring Sultanate of Pattani for trade.
Sentence Splitting	He was involved in two conversions **which** turned out to be crucial.	He was involved in two conversions**. These conversions** turned out to be crucial.
Compression	~~Think of all the ways~~ everyone in your household will benefit from your membership in Audubon.	Everyone in your household will benefit from membership in Audubon.

In this Section, we introduce MulTSS (**Mul**ti-**T**ask **S**entence **S**implification), our proposed model for multi-operation Sentence Simplification. We describe and justify our selection of rewriting tasks and model architecture.

Given an original sentence, our main task is to generate a simplified version of it. In order to complement Sentence **Simplification** information, and to allow a model to learn more diverse text transformations, we propose to use data from related rewriting tasks. Based on our findings from Sec. 3.1 and previous studies on how humans simplify (Aluísio et al., 2008; Bott and Saggion, 2011; Petersen and Ostendorf, 2007), we selected as auxiliary tasks: **Compression** (i.e. deleting words considered less relevant), Lexical **Paraphrasing** (i.e., replacing complex words/phrases with some rewriting for fluency), and Sentence **Splitting** (i.e. dividing a long sentence into several smaller ones). The examples in Table 6.1 showcase the close relationship between our selected auxiliary tasks and Sentence Simplification, further justifying the potential benefit of training a joint model. This differs from the tasks selected in (Guo et al., 2018), sentence paraphrasing and entailment generation, that focus more on generating outputs that are meaning preserving, rather than on performing text transformations that could improve the lexical or structural simplicity of the input sentence.

We implement a one-to-many (i.e. same source with different targets) sequence-to-sequence model, adopting Johnson et al. (2017)'s MTL architecture, originally for multilingual NMT. We treat each of our monolingual tasks as a different "language" pair, namely: original→simplified (Simplification), original→compressed (Compression), original→paraphrased (Lexical Paraphrasing), and original→split (Sentence Splitting). All tasks share the encoder and the decoder, and the target-task token is appended to the beginning of each input sentence.[1] In addition, different from Johnson et al. (2017), we experiment with a Transformer-based (Vaswani et al., 2017) encoder-decoder architecture instead of an RNN-based one, due to its success in recent state-of-the-art Sentence Simplification models (Martin et al., 2020; Zhao et al., 2018).

6.3 Experimental Settings

6.3.1 Training Data

For **Simplification** we used WikiLarge (Zhang and Lapata, 2017), which contains 286K simplification instances for several rewriting transformations. For **Lexical Paraphrasing** we used sscorpus (Kajiwara and Komachi, 2016), which provides automatic 1-to-1 sentence alignments from EW and Simple EW. We filtered out instances with a similarity below 0.7 to reduce noise, resulting in 183K instances. For **Splitting** we used WikiSplit (Botha et al., 2018), with 990K instances of sentence splitting (1-to-2). For **Compression** we used the Google dataset for extractive compression (Filippova et al., 2015), with 200K training instances.

6.3.2 Evaluation Metrics and Dataset

Based on the set of guidelines we proposed in Sec. 5.4, we measure the quality of automatic simplifications using BERTScore (Zhang et al., 2020) and SARI (Xu et al., 2016). To be in line with previous work, we also report BLEU (Papineni et al., 2002). For a fair comparison, we detokenised and recased all original sentences and system outputs using scripts from Moses.[2] Then, we used EASSE (Alva-Manchego et al., 2019a) to compute all metrics with the same configuration: tokenisation using SacreMoses[3] and case-sensitive calculation. In order to compute the metrics during development and testing time, we used ASSET (Chap. 4). Recall

[1] Preliminary experiments using multiple encoders and/or decoders did not yield better results.

[2] https://github.com/moses-smt/mosesdecoder/tree/master/scripts

[3] https://github.com/alvations/sacremoses

that it contains 2,359 multi-operation multi-reference simplifications, for both validation (2,000 instances) and testing (359 instances).

6.3.3 Model Configuration

We used the implementation for multilingual NMT available in `fairseq` (Ott et al., 2019). The Transformers were setup following Zhao et al. (2018): initialisation of encoder and decoder with 300-dimensional Glove vectors (Pennington et al., 2014), 4 encoder and decoder layers with 5 attention heads, and dropout rate of 0.3. We used Adam for optimisation with $\beta = (0.9, 0.98)$, and a learning rate of $lr = 0.0005$. Full training details are provided in Appendix D. Training instances were tokenised using SentencePiece (Kudo and Richardson, 2018), with a BPE size of 15,000, removing sentences with < 1 BPE tokens and > 250 BPE tokens. For generation we used beam search with a beam size of 5. After training for 30 epochs, the checkpoint with the highest BERTScore$_{\text{precision}}$ scores in the validation set was chosen.

6.3.4 Baselines and Comparison Systems

Original and Reference. We compute scores for the worst and best case scenarios. The former corresponds to repeating the original sentence as-is, i.e. not performing any changes. The latter corresponds to selecting a random reference per instance in the test set, and computing metrics' scores using the other manual simplifications as references. These scores serve as lower and upper bounds to compare against.

Single-Task Baselines. Models for Sentence Simplification trained with: (1) simplification data from WikiLarge only, or (2) pooling the training data of all tasks together.

Pipeline Baselines. We trained a model for each of the three auxiliary generation tasks, and applied them consecutively for a total of six pipeline permutations.

Previous Work. Models with publicly-available system outputs that were trained on WikiLarge, and therefore should have learned to produce multi-operation simplifications, even if they were not tested for that: DRESS-LS (Zhang and Lapata, 2017), a neural model using an RNN-based encoder-decoder architecture with attention combined with reinforcement learning; DMASS-DCSS (Zhao et al., 2018), which is based on the Transformer architecture and memory-augmentation with paraphrasing rules from the Simple Paraphrase Database (SPPDB, Pavlick and Callison-Burch, 2016); and ACCESS (Martin et al., 2020), the current state-of-the-art in TurkCorpus and ASSET, a neural model also with a Transformed-based encoder-decoder architecture that conditions the generation of simplifications on explicit desired text attributes (e.g. length and/or dissimilarity with original input). We also included SBMT-SARI (Xu et al., 2016), a syntax-based statistical MT model trained on the Paraphrase Database (PPDB, Ganitkevitch et al., 2013). Finally, we also considered Hybrid (Narayan and Gardent, 2014), which relies on phrase-based statistical MT, coupled with semantic analysis to learn to split sentences. This model was not trained on WikiLarge, but due to its potential splitting capabilities we regarded it as relevant to compare against. We could not compare against the MTL model of Guo et al. (2018) since its output is not publicly available.

6.4 Evaluation

6.4.1 Automatic Metrics

Table 6.2 presents the results of all our baselines and MulTSS on the test set of ASSET. Repeating the original sentence achieves the highest BERTScore and BLEU scores, which could be caused by several words being kept as-is in the reference simplifications. On the other hand, a random reference reaches a much lower BLEU, the highest SARI score and a BERTScore of around 0.79. When interpreting the results for the other models, it is important to pay attention to this interaction between the values of the different automatic metrics.

When compared to the single-task baselines, MulTSS has the highest SARI score and a BERTScore that is closest to that of the Random Reference. Its BLEU score, however, points out to some conservative behaviour of the model. It should be noticed that just pooling all datasets together does not seem to result in a model capable of learning to mix the rewriting operation they exhibit. In the case of the pipeline baselines, they all score low in BERTScore, indicating low quality simplifications. Via manual inspection, we noticed that their low scores are caused by the compression model being too aggressive. This also leads to only the pipelines that perform splitting as their last step being able to divide sentences.

When compared to previous work, MulTSS has the lowest SARI score and second-highest BERTScore. Most of these models attempt to learn explicit lexical changes. For instance, SBMT-SARI and DMASS-DCSS integrate rewriting rules from PPDB and SPPDB, respectively, whilst DRESS-LS has a dedicated pre-trained model for lexical simplification in its reinforcement learning architecture. However, ACCESS, the model with the closest scores to the Random Reference, does not incorporate a specific component for lexical simplification.

Table 6.2 Automatic evaluation of baselines, previous work, and MulTSS on the Simplification task, using simplifications from the test set of ASSET as references.

	BLEU	SARI	BERTScore
Original Sentence	92.65	20.46	0.9023
Random Reference	66.73	43.95	0.7913
Single-task Baselines			
Simplification-only	87.36	32.84	**0.8590**
Data Pooling	72.55	30.98	0.7556
Pipeline Baselines			
paraphrase→split→compress	45.17	33.78	0.6469
paraphrase→compress→split	39.39	34.31	0.5005
split→paraphrase→compress	44.38	33.71	0.6504
split→compress→paraphrase	45.65	33.77	0.6553
compress→paraphrase→split	38.85	34.06	0.4971
compress→split→paraphrase	44.38	33.78	0.6049
Multi-Task Model			
MulTSS	81.33	**34.31**	0.8235
Previous Work			
Hybrid	55.68	34.50	0.6270
SBMT-SARI	69.61	36.63	0.7888
DRESS-LS	85.63	36.38	0.8491
DMASS-DCSS	70.53	38.41	0.7378
ACCESS	75.28	39.86	0.7918

As such, it appears that conditioning text attributes of the output is also beneficial. Finally, Hybrid has the lowest BLEU and BERTScore, and by manually inspecting the output it did not perform any sentence splitting, despite being explicitly trained to do so.

6.4.2 Human Judgements

We crowdsourced human ratings with the same methodology as in Sec. 4.5: workers on Amazon Mechanical Turk, Qualification Test to select raters, and eliciting continuous Direct Assessment scores (Graham et al., 2013). We randomly chose 50 original sentences from ASSET's test set and, for each of them, we sampled one simplification from the Simplification-only baseline and MulTSS. For each instance, we collected 10 ratings on fluency (i.e. grammaticality), adequacy (i.e. meaning preservation) and simplicity, using the same HIT design as in Sec. 4.4.1. For calculating inter-annotator agreement, we followed the process in Sec. 4.5. We obtained mean and standard deviations of 0.3155 ± 0.0662 for Fluency, 0.5005 ± 0.0600 for Adequacy and

Table 6.3 Results of human evaluation (standardised scores) comparing MulTSS and the Simplification-only baseline.

Model	Fluency	Adequacy	Simplicity
Simplification-only	0.128	0.147	-0.486
MulTSS	**0.284**	**0.327**	**-0.477**

0.4638 ± 0.0512 for Simplicity. All values point to a moderate level of agreement, which goes in accordance with the subjectivity of the simplification task. Results in Table 6.3 confirm that MulTSS generates better simplifications than the Simplification-only baseline, but it is apparent that the model is conservative (high adequacy).

6.5 Analysis

6.5.1 Impact of Auxiliary Tasks

Table 6.4 shows the variation in the performance of MulTSS depending on which auxiliary tasks are used for training. It is worth noticing that the model trained without splitting data has a BERTScore that is almost similar to that of the model trained with all tasks (0.8202 vs. 0.8235). It also achieves a slightly higher SARI score (34.82 vs. 34.31). However, through manual inspection, we see that it does not learn to divide sentences. This points to the necessity of developing better evaluation methods that could take into consideration multiple operations when scoring a system's output.

Table 6.4 Evaluation of MulTSS on Simplification (S), showing the impact of the auxiliary tasks: Paraphrasing (P), Splitting (Sp) and Compression (C).

Generation Tasks				Evaluation Metrics		
S	P	Sp	C	BLEU	SARI	BERTScore
✓	✓	✓	✓	81.33	34.31	**0.8235**
✓	✓			80.50	32.10	0.8065
✓		✓		80.78	33.99	0.8163
✓			✓	80.20	32.67	0.8150
✓	✓	✓		81.31	33.40	0.8215
✓	✓		✓	80.81	**34.82**	0.8202
✓		✓	✓	**81.65**	32.98	0.8224

6.5.2 Evaluation on Auxiliary Tasks

Even though our main task is Sentence Simplification, since we trained MulTSS as a one-to-many model, it is possible that it could perform the auxiliary tasks too. Table 6.5 presents an evaluation using automatic metrics in TurkCorpus (**Paraphrasing** task). MulTSS is better than the baseline, but previous work still achieved the best scores in SARI. Through manual inspection, we confirmed that MulTSS can distinguish between the types of rewriting allowed for each task. For instance, for all original sentences that it performed splitting when simplifying, it did not when lexical paraphrasing. In the case of the **Splitting** task (Table 6.6), MulTSS is not able to outperform the single-task baseline in terms of BLEU. However, it is able to split inputs in as many sentences as the reference. The decrease in BLEU scores could be caused by the models learning to paraphrase more than is required, due to information from the other tasks. In Splitting, the goal is only to divide a sentence with limited rewriting to ensure fluency. Finally, for the **Compression** task (Table 6.7), MulTSS outperforms the single-task baseline in terms of F1, and achieves a similar compression ratio as the reference.

Table 6.5 Automatic evaluation on the Lexical Paraphrasing task, using the test set of TurkCorpus.

	BLEU	SARI
Original Sentence	99.36	26.07
Random Reference	70.79	40.21
Paraphrasing-only	84.15	33.86
SBMT-SARI	71.98	39.13
DRESS-LS	80.59	36.81
DMASS-DCSS	72.26	39.67
ACCESS	75.56	**41.05**
MulTSS	83.30	35.50

Table 6.6 Automatic evaluation on the Splitting task, using the test set of WikiSplit. Metrics are: BLEU (corpus-level), sBLEU (macro-average sentence-level), #S/C (average number of simple sentences per complex one), #T/S (average number of tokens per simple sentence).

	BLEU	sBLEU	#S/C	#T/S
Original Sentence	74.17	72.86	1.0	29.04
Reference	100.00	100.00	**2.0**	15.96
Splitting-only	**76.67**	**75.99**	**2.0**	14.60
MulTSS	74.33	73.71	**2.0**	14.53

Table 6.7 Automatic evaluation on the Compression task, using the test set of Google Compression.

	F1 (Token)	Comp. Ratio
Original Sentence	59.37	1.00
Reference	100.00	0.43
Compression-only	79.86	**0.41**
MulTSS	**81.70**	**0.41**

6.6 Summary and Final Remarks

In this Chapter, we attempted to answer the research question: **Can we devise models that leverage data with more types of rewriting to enhance the multi-operation capabilities of simplification systems?**

We proposed **a multi-task approach for multi-operation simplification**. Our architecture is motivated by the multi-task learning framework of Johnson et al. (2017) for multilingual NMT and, in general, the success of using side constraints (Sennrich et al., 2016) in Sentence Simplification (Martin et al., 2020; Scarton et al., 2020; Scarton and Specia, 2018). We implemented a one-to-many model with Sentence Simplification as our main task and Lexical Paraphrasing, Compression, and Sentence Splitting as auxiliary tasks. This differs from the approach in (Guo et al., 2018) since our auxiliary tasks are better suited for inducing changes that improve the simplicity of input sentences, rather than just preserving their meaning.

Our proposed model showed **promising results for performing multi-operation simplifications**, achieving better BERTScore and SARI scores than strong baselines. In particular, we demonstrated that training a multi-operation model in a multi-task framework is more beneficial than simply pooling together all data from the main and auxiliary tasks, or executing single-task models in a pipeline fashion. Our model is competitive to some previous work through automatic and human evaluation, but shows signs of being more conservative than state-of-the art models. Furthermore, our analyses showed that sentence splitting data may not be as influential as that from other auxiliary tasks, and that our multi-task model can perform compression better than a single-task baseline.

In order to further explore the potential of our multi-task architecture and improve its performance, we could investigate how to better balance the information that models learn from each task. For instance, we could sample data from each dataset to have around the same number of training instances for each task. We could also attempt a more automatic approach by implementing temperature-based sampling methods, like in (Arivazhagan et al., 2019). Alternatively, we could filter the current datasets to reduce the number of instances where the target side is too similar to the source side. Finally, we should explore fine-tuning pre-trained sequence-to-sequence models, such as BART (Lewis et al., 2020).

All the previously-described experiments and findings allow us to suggest an answer to our research question: Multi-task models are able to combined training data from related rewriting tasks and benefit Sentence Simplification. However, in order to take full advantage, it is important to balance the contribution of each task, so that they do not overpower each other, resulting in a conservative model.

Chapter 7

Conclusions

In this Thesis, we have studied the problem of automatically simplifying sentences by applying multiple rewriting transformations. Chapter 2 presented a comprehensive review of data-driven approaches for Sentence Simplification that allowed us to identify one key limitation in the area: even though human editors perform several simplification operations simultaneously to improve the readability of sentences, it is unknown if automatic models are able to simplify in the same way. We tackled this issue in four steps.

First, Chapter 3 focused on determining which simplification operations are performed by automatic systems. We implemented novel algorithms that annotate simplification operations in given original-simplified sentence pairs at word, phrase and sentence levels. Based on these annotations, we proposed an operation-based error analysis method that complements the information provided by automatic metrics. We proved the usefulness of this evaluation approach by conducting the first benchmark of Sentence Simplification models with operation-correctness details. However, the datasets used in the benchmark are limited since they are either multi-reference but with manual single-operation simplifications (Xu et al., 2016), or have multi-operation simplifications but with a single noisy reference (Zhu et al., 2010).

Consequently, Chapter 4 introduced ASSET, a new multi-reference evaluation dataset with manual multi-operation simplification references. We detailed the crowdsourcing methodology and quality controls implemented for data collection. We also presented experiments showing that ASSET is superior to other multi-reference single-operation datasets (Sulem et al., 2018a; Xu et al., 2016) in terms of variability of rewriting operations and the simplicity of its manual references as judged by humans. Furthermore, we presented a preliminary study to analyse the correlation of human judgements and automatic metrics computed using ASSET's references. We determined that neither BLEU (Papineni et al., 2002) nor SARI (Xu et al., 2016) are good estimates of simplicity when using multi-operation simplification references.

Based on that last finding, Chapter 5 aimed to establish how to better use available evaluation metrics and data for assessing automatic multi-operation simplifications. As such, we conducted the first meta-evaluation of automatic metrics for Sentence Simplification. We studied the variation in the correlation of human judgements on simplicity with automatic metrics across three dimensions: the quality of the automatic output, the system type, and the set of simplification references. We determined that all three aspects affect the correlations and, in particular, further analysed the behaviour of simplification-specific metrics: SARI (Xu et al., 2016) and SAMSA (Sulem et al., 2018b). Based on our findings, we elaborated a set of guidelines for evaluating multi-operation simplification models.

Finally, Chapter 6 described a multi-task learning approach for training a multi-operation simplification model. We re-purposed Johnson et al. (2017)'s architecture for multilingual NMT to jointly train a model in four rewriting tasks: Sentence Simplification, Lexical Paraphrasing, Compression and Sentence Splitting. When evaluated using the guidelines from Chapter 5, our multi-task model performed better than strong single-task and pipeline baselines for Sentence Simplification. Most importantly, we showed that training in a multi-task way is better than simply joining all data together. Despite these promising results, our model was not better than most previous work. This is due to it being conservative, which motivates the need for exploring better techniques to select and balance the data of each task during training.

In this Chapter, we review and summarise our main findings and contributions (Sec. 7.1), and provide directions for future work (Sec. 7.2).

7.1 Summary of Findings

We now revisit the research questions that motivated each Chapter, and summarise the contributions that resulted from the process of finding their answers.

Which simplification operations do automatic systems perform accurately?

We conducted a manual study on automatically-aligned original-simplified sentence pairs from the Newsela corpus (Xu et al., 2015), and determined that the most common sentence-level simplification operations are related to lexical paraphrasing (REPLACE and SIMPLE REWRITE), compression (DELETE), addition of information (ADD), and sentence splitting (SPLIT). After that, we implemented novel annotation algorithms capable of identifying DELETE, REPLACE, ADD, REORDER, and SPLIT, using labels at word, phrase and sentence levels, accordingly. An intrinsic evaluation showed that these automatic annotations have relatively-high accuracy when compared to gold-standard human labels. As extrinsic evaluation, the automatic word-level labels were used to generate training data for SeqLab (Alva-Manchego et al., 2017), a system that models simplification as a sequence labelling problem. We showed that by explicitly indicating whether a token in a given original sentence needed to be deleted, replaced or copied, we could generate better simplifications than standard MT-based architectures. We further used the word-level annotations as a form of operation-based error analysis, based on comparing the automatic labelling of an original sentence paired with a reference simplification vs. an automatic simplification. This method was included in EASSE (Alva-Manchego et al., 2019a), a software package that also provides access to public evaluation datasets, standard implementations of automatic metrics, quality estimation features, and performance reports. With these functionalities, we elaborated the first benchmark of Sentence Simplification systems that compares models using both standard overall-performance metrics and our proposed operation-based error analysis (Alva-Manchego et al., 2020b). The latter, in particular, provided us with insights about the simplification capabilities of each system that helped better explain the metrics' scores. Most importantly, it allowed us to propose an answer to our research question: Current Sentence Simplification models are good at preserving content from the original input sentences. However, this is due to their mostly-conservative nature. We still need to provide them with the ability to accurately determine what to copy and what to delete. In addition, although progress has been made in recent years, replacing complex words with simpler synonyms is still challenging.

Do we need multi-reference multi-operation evaluation datasets to support better assessment of Sentence Simplification models?

The focus of current multi-reference datasets for automatic evaluation is on single-operation simplifications: lexical paraphrasing in TurkCorpus (Xu et al., 2016) and sentence splitting in HSplit (Sulem et al., 2018a). We argued that this prevents studying a model's ability to perform several simplification operations holistically. To counteract this limitation, we introduced ASSET (Alva-Manchego et al., 2020a), a new dataset for tuning and evaluation of Sentence Simplification models. We designed guidelines and implemented quality controls mechanisms to crowdsource 10 high-quality manual simplifications with multiple operations (compression, lexical paraphrasing and splitting) for each of the 2,359 original sentences in TurkCorpus (2,000 for the development set and 359 for the test set). By exploiting our annotation algorithms from Chapter 3 and computing a series of surface-level text features, we showed that simplifications in ASSET offer more variability in the operations performed by human editors than those in TurkCorpus and HSplit. In addition, we crowdsourced preference judgements comparing simplifications in ASSET against TurkCorpus and HSplit for all 359 sentences in the test set. Results showed that humans judged simplifications in ASSET as simpler than those in other evaluation datasets. Finally, we conducted a preliminary study on the suitability of BLEU (Papineni et al., 2002) and SARI (Xu et al., 2016) for evaluating multi-operation simplifications when computed using references in ASSET. We crowdsourced human ratings on fluency, meaning preservation and simplicity for automatic outputs of five state-of-the-art sentence simplification systems, and calculated their correlations with automatic scores. We found that none of the metrics shows a strong correlation with simplicity judgements. In particular, we argue that SARI is unsuitable for assessing automatic simplifications where multiple operations have been applied. Based on our findings, we proposed an answer to our research question: When allowed to do so, human editors tend to apply several rewriting operations when simplifying sentences, and generate various references that are simpler than those produced by executing a single simplification operation. However, current evaluation metrics are not capable of exploiting this improvement in reference quality. Therefore, these less-restricted multi-operation simplifications could serve to study and develop more suitable evaluation metrics and/or protocols that improve automatic evaluation.

Can current evaluation metrics measure the ability of automatic systems to perform multi-operation simplifications?

We extended the preliminary study from the end of Chapter 4 by collecting a new dataset of human judgements on automatic simplifications, and by performing the first meta-evaluation of automatic metrics in the area. Inspired by the Direct Assessment methodology (Graham et al., 2017), we crowdsourced 15 human ratings on fluency, meaning preservation and simplicity for 600 system outputs from 6 different simplification models. This makes our dataset (Simplicity-DA) bigger and more varied than the Simplicity Gain (Xu et al., 2016) dataset, and more reliable than the Structural Simplicity (Sulem et al., 2018c) dataset. We used all three datasets to analyse the variations of the correlations of sentence-level metrics with human judgements along three dimensions: the perceived simplicity level, the system type, and the set of references used to compute the automatic scores. For Simplicity-DA, in particular, we determined that: metrics can more reliably score low-quality simplifications; most metrics are better at scoring system outputs from neural sequence-to-sequence models; and combining all available multi-reference datasets does not significantly improve metrics' correlations over using only ASSET. We also offered some explanations on the low-to-moderate correlations achieved by simplification-specific metrics SARI (Xu et al., 2016) and SAMSA (Sulem et al., 2018b). Based on our findings, we designed a set of guidelines for automatic evaluation of current simplification models. In particular for multi-operation simplifications, we suggested to use BERTScore (Zhang et al., 2020) with references from ASSET during the development stage of simplification models, and manual evaluation for final comparisons. Finally, we proposed that future metrics should take both the original sentence and the automatic simplification into consideration, that manual references be enriched with their simplicity level, among other recommendations. Based on our meta-evaluation, we suggested an answer to our research question: Referenced-based metrics can potentially estimate the quality of automatic simplifications provided that multi-operation multi-reference evaluation datasets are available. However, the automatic scores are not reliable for all types of systems and across levels of quality, particularly due to each reference having a different level of simplicity.

Can we devise models that leverage data with more types of rewriting to enhance the multi-operation capabilities of simplification systems?

We proposed a multi-task approach for multi-operation simplification based on the framework of Johnson et al. (2017) for multilingual NMT. We implemented a one-to-many model with Sentence Simplification, Lexical Paraphrasing, Compression, and Sentence Splitting as our tasks. Our proposed model showed promising results for multi-operation simplification, achieving better BERTScore and SARI scores than strong baselines. In particular, we demonstrated that training a multi-operation model in a multi-task framework is more beneficial than simply pooling together the data from all tasks, or executing single-task models in a pipeline fashion. Even though our model is competitive to some previous work, it showed signs of being more conservative than state-of-the-art models. We hypothesised that this could be improved by better balancing of the information that the model learns from each task, perhaps through temperature-based sampling methods, like in (Arivazhagan et al., 2019). Another interesting way forwards is to explore fine-tuning pre-trained sequence-to-sequence models, such as BART (Lewis et al., 2020). Based on our results, we suggested an answer to our research question: Multi-task models are able to combined training data from related rewriting tasks and benefit Sentence Simplification. However, in order to take full advantage, it is important to balance the contribution of each task, so that they do not overpower each other, resulting in a conservative model.

7.2 Future Research

In this final Section, we look into the future and propose directions for research in Text Simplification. A previous version of this Section appears in (Alva-Manchego et al., 2020b).

Corpora Diversity

Most datasets used in Sentence Simplification research are based on Simple English Wikipedia. Its quality has been questioned (Sec. 2.2.1), but its public availability and shareability makes it popular for research purposes. The Newsela corpus offers the advantage of being produced by professionals, which ensures a higher quality across all texts available. However, the fact that common splits of the data cannot be publicly-shared hinders the development and objective comparison of models that use it. We believe that our research area would benefit from datasets that combine the positive features of both: high-quality professionally-produced data that can be publicly shared. In addition, it would be desirable that these new datasets be as diverse as possible in terms of application domains, target audiences, and simplification operations realised. It would also be valuable to follow (Alva-Manchego et al., 2020a; Sulem et al., 2018a; Xu et al., 2016) by collecting multiple references per simplification, for a fairer evaluation.

Modular Simplification

Most sequence-to-sequence approaches for training Sentence Simplification models could be considered as black boxes with respect to which simplification operations should be applied to a given sentence. That is a desirable feature for a holistic approach to the task, where the rewriting operations interact with each other, and are not necessarily applied in isolation (e.g. a sentence can be split, and some of its components deleted/reordered simultaneously). However, it could also be desirable to have a more modular approach to the problem: to first determine which simplifications should be performed in a given sentence, and then decide how to handle each operation independently (potentially using a different approach for each one). This could be specially beneficial in Personalised Simplification, since modules could be adapted to user-specific needs. For example, the simplification operation to applied may be different for a non-native speaker or for a person with autism or with a low-literacy level. Approaches based on labelling (Alva-Manchego et al., 2017; Bingel and Søgaard, 2016) could be helpful in such cases. A disadvantage, however, is collecting quality annotated data from which to learn. In addition, some simplification operations are hard to predict (e.g. addition of words that do not come from the original sentence).

Simplification Evaluation

For automatic evaluation of Sentence Simplification models, there are only two simplicity-specific metrics: SARI (focused on lexical paraphrasing) and SAMSA (focused on sentence splitting). However, as shown in Chapter 5, they have limitations and are not suitable to assess this quality aspect in multi-operation simplifications. We believe that more work needs to be done in improving how we evaluate and compare Sentence Simplification models automatically. Research on Quality Estimation has shown promising results on using reference-less metrics to evaluate generated outputs, allowing the automatic assessment to speed-up and scale. This line of work has started to be applied for Simplification (Martin et al., 2018; Štajner et al., 2016b), and we believe it needs to be explored further. In addition, human-based evaluation has been limited to three criteria: grammaticality, meaning preservation and simplicity. Are these criteria enough? Would they still be relevant if we moved to a document-level perspective? Would assessing the usefulness of the simplifications for target users (Mandya et al., 2014) be a more reliable quality measure? We believe these are questions that need to be addressed.

Document-level Approaches

Most research on Text Simplification has focused on studying simplification of individual sentences. Reducing the scope of the problem has allowed the easier collection and curation of corpora, as well as adapting methods from other text generation tasks, mainly Machine Translation. It can be argued that "true" Text Simplification (i.e. document-level) cannot be achieved by simplifying sentences one at a time. However, there is little research on tackling the problem with a document-level perspective. Woodsend and Lapata (2011b) and Mandya et al. (2014) produce simplifications at the sentence-level and try to globally optimise readability scores or length of the document. However, Siddharthan (2003) points out that syntactic alterations to sentences (especially, splitting) can affect the rhetorical relations between them, which can only be resolved going beyond sentence boundaries. This is an exciting area of research, since simplifying a complete document is a more real use-case scenario for a simplification model. This line of research should begin with identifying what makes document simplification different from sentence simplification. In (Alva-Manchego et al., 2019b), we showed that simplifying a document involves considering the interactions between individual sentences to determine which ones to include in the simplified version (content selection), if they must be combined (sentence fusion), and in which order they must appear (sentence reordering). These decisions can only be carried out by taking a document-level perspective in the simplification process. In addition, proper corpora should be curated for training and testing of data-driven models, as well as new evaluation methodologies should be devised. The Newsela corpus is a resource that could be exploited in this regard. So far, it has only been used for sentence-level simplification, even though it contains original-simplified aligned documents, with versions in several simplification levels.

Sentence Joining

Most current Sentence Simplification models perform sentence compression, in that they may delete part of the content of a sentence that could be regarded as unimportant. However, as presented in Sec. 2.1.1, studies have shown that humans tend to join sentences together while simplifying a text, and perform a form of abstractive summarisation. No state-of-the-art Text Simplification model considers this type of operation while transforming a text, perhaps because they only process one sentence at a time. Incorporating sentence joining could help in developing a document-level perspective into current Text Simplification systems.

Text Simplification is a research area with significant application potential. It can have a meaningful impact in people's lives and help create a more inclusive society. With the development of new Natural Language Processing technologies (especially neural-based models), it is starting to receive more attention in the recent years. However, there are still several open questions that pose challenges to our research community. We hope that this Thesis has contributed to advance the knowledge in the area.

References

Omri Abend and Ari Rappoport. 2013. 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. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. 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 1549–1564, Hong Kong, China. Association for Computational Linguistics.

Roee Aharoni and Yoav Goldberg. 2018. 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. Association for Computational Linguistics.

Sandra M. Aluísio, Lucia Specia, Thiago A. S. Pardo, Erick G. Maziero, Helena M. Caseli, and Renata P. M. Fortes. 2008. 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. ACM.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020a. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Louis Martin, Carolina Scarton, and Lucia Specia. 2019a. EASSE: Easier automatic sentence simplification evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong, China. Association for Computational Linguistics.

References

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

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

Marcelo Amancio and Lucia Specia. 2014. An Analysis of Crowdsourced Text Simplifications. In *Third Workshop on Predicting and Improving Text Readability for Target Reader Populations*, PITR 2014, pages 123–130, Gothenburg, Sweden.

Bharat Ram Ambati, Siva Reddy, and Mark Steedman. 2016. Assessing relative sentence complexity using an incremental CCG parser. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1051–1057, San Diego, California. Association for Computational Linguistics.

N. Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, M. Johnson, M. Krikun, M. Chen, Yuan Cao, G. Foster, Colin Cherry, Wolfgang Macherey, Z. Chen, and Y. Wu. 2019. Massively multilingual neural machine translation in the wild: Findings and challenges. *arXiv e-prints*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018a. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018b. Unsupervised neural machine translation. In *Proceedings of the 6th International Conference on Learning Representations (ICLR 2018)*, Vancouver, BC, Canada.

Nguyen Bach, Qin Gao, Stephan Vogel, and Alex Waibel. 2011. 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. Asian Federation of Natural Language Processing.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–55, Online. Association for Computational Linguistics.

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. 2019. 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. Association for Computational Linguistics.

Regina Barzilay and Noemie Elhadad. 2003. Sentence Alignment for Monolingual Comparable Corpora. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Beata Beigman Klebanov, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, pages 735–747, Berlin, Heidelberg. Springer Berlin Heidelberg.

Yonatan Belinkov, Nadir Durrani, Fahim Dalvi, Hassan Sajjad, and James Glass. 2017. What do neural machine translation models learn about morphology? In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 861–872, Vancouver, Canada. Association for Computational Linguistics.

Joachim Bingel, Maria Barrett, and Sigrid Klerke. 2018a. Predicting misreadings from gaze in children with reading difficulties. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 24–34, New Orleans, Louisiana. Association for Computational Linguistics.

Joachim Bingel, Gustavo Paetzold, and Anders Søgaard. 2018b. Lexi: A tool for adaptive, personalized text simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 245–258, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Joachim Bingel and Anders Søgaard. 2016. Text simplification as tree labeling. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 337–343, Berlin, Germany. Association for Computational Linguistics.

Or Biran, Samuel Brody, and Noemie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501, Portland, Oregon, USA. Association for Computational Linguistics.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017a. Findings of the 2017 conference on machine translation (WMT17). In *Proceedings of the Second Conference on Machine Translation*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016a. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 workshop on statistical machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Philipp Koehn, and Christof Monz. 2018. 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. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, and Amir Kamran. 2017b. Results of the WMT17 metrics shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 489–513, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Yvette Graham, Amir Kamran, and Miloš Stanojević. 2016b. Results of the WMT16 metrics shared task. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 199–231, Berlin, Germany. Association for Computational Linguistics.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. 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. Association for Computational Linguistics.

Stefan Bott and Horacio Saggion. 2011. Spanish text simplification: An exploratory study. *Procesamiento del Lenguaje Natural*, 47:87–95.

Stefan Bott, Horacio Saggion, and Simon Mille. 2012. Text simplification tools for Spanish. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 1665–1671, Istanbul, Turkey. European Language Resources Association (ELRA).

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Dominique Brunato, Lorenzo De Mattei, Felice Dell'Orletta, Benedetta Iavarone, and Giulia Venturi. 2018. 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. Association for Computational Linguistics.

Arnaldo Candido Jr., Erick Maziero, Caroline Gasperin, Thiago A. S. Pardo, Lucia Specia, and Sandra M. Aluisio. 2009. Supporting the adaptation of texts for poor literacy readers: A text simplification editor for brazilian portuguese. In *Proceedings of the Fourth Workshop on Innovative Use of NLP for Building Educational Applications*, BEA '09, pages 34–42, Boulder, Colorado. Association for Computational Linguistics.

John Carroll, Guido Minnen, Yvonne Canning, Siobhan Devlin, and John Tait. 1998. 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.

Helena M. Caseli, Tiago F. Pereira, Lucia Specia, Thiago A. S. Pardo, Caroline Gasperin, and Sandra M. Aluisio. 2009. Building a brazilian portuguese parallel corpus of original and simplified texts. In *Proceedings of the 10th Conference on Intelligent Text Processing and Computational Linguistics*, pages 59–70, Mexico City, Mexico.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *Proceedings of the 16th Conference on Computational Linguistics*, volume 2 of *COLING '96*, pages 1041–1044, Copenhagen, Denmark. Association for Computational Linguistics.

David Chiang. 2006. An introduction to synchronous grammars. Tutorial at ACL 2006. Available at `http://www3.nd.edu/~dchiang/papers/synchtut.pdf`.

Shamil Chollampatt and Hwee Tou Ng. 2018. A reassessment of reference-based grammatical error correction metrics. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2730–2741, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

Jacob Cohen. 1968. Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit. *Psychological Bulletin*, 70(4):213–220.

Trevor Cohn and Mirella Lapata. 2009. Sentence compression as tree transduction. *Journal of Artificial Intelligence Research*, 34:637–674.

Trevor Cohn and Mirella Lapata. 2013. An abstractive approach to sentence compression. *ACM Trans. Intell. Syst. Technol.*, 4(3):41:1–41:35.

Michael Cooper and Matthew Shardlow. 2020. CombiNMT: An exploration into neural text simplification models. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5588–5594, Marseille, France. European Language Resources Association.

William Coster and David Kauchak. 2011a. Learning to simplify sentences using wikipedia. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 1–9, Portland, Oregon. ACL.

William Coster and David Kauchak. 2011b. Simple english wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 665–669, Stroudsburg, PA, USA. Association for Computational Linguistics.

Koby Crammer and Yoram Singer. 2003. Ultraconservative Online Algorithms for Multiclass Problems. *J. Mach. Learn. Res.*, 3:951–991.

References

Scott A. Crossley, Max M. Louwerse, Philip M. McCarthy, and Danielle S. McNamara. 2007. A linguistic analysis of simplified and authentic texts. *The Modern Language Journal*, 91(1):15–30.

James R. Curran, Stephen Clark, and Johan Bos. 2007. Linguistically motivated large-scale nlp with c&c and boxer. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 33–36, Prague, Czech Republic. ACL.

Jan De Belder and Marie-Francine Moens. 2010. Text Simplification for Children. In *Proceedings of the SIGIR 2010 Workshop on Accessible Search Systems*, pages 19–26, Geneva. ACM.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Siobhan Devlin and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, pages 161–173.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1723–1732, Beijing, China. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, ACL '03, pages 205–208, Sapporo, Japan. ACL.

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

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2020. Summeval: Re-evaluating summarization evaluation. *arXiv e-prints*.

Angela Fan, David Grangier, and Michael Auli. 2018. Controllable abstractive summarization. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 45–54, Melbourne, Australia. Association for Computational Linguistics.

Dan Feblowitz and David Kauchak. 2013. Sentence simplification as tree transduction. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 1–10, Sofia, Bulgaria. Association for Computational Linguistics.

Katja Filippova, Enrique Alfonseca, Carlos A. Colmenares, Lukasz Kaiser, and Oriol Vinyals. 2015. Sentence compression by deletion with lstms. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 360–368, Lisbon, Portugal. Association for Computational Linguistics.

Katja Filippova and Michael Strube. 2008. Dependency tree based sentence compression. In *Proceedings of the Fifth International Natural Language Generation Conference*, INLG '08, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Pierre Finnimore, Elisabeth Fritzsch, Daniel King, Alison Sneyd, Aneeq Ur Rehman, Fernando Alva-Manchego, and Andreas Vlachos. 2019. Strong baselines for complex word identification across multiple languages. 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 970–977, Minneapolis, Minnesota. Association for Computational Linguistics.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875, San Diego, California. Association for Computational Linguistics.

Rudolph Flesch. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221.

Marina Fomicheva and Lucia Specia. 2019. Taking mt evaluation metrics to extremes: Beyond correlation with human judgments. *Computational Linguistics*, 45(3):515–558.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be Guilty but References are not Innocent. *arXiv e-prints*.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, Georgia. Association for Computational Linguistics.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 179–188, Vancouver, Canada. Association for Computational Linguistics.

Goran Glavaš and Jan Šnajder. 2015. Construction and evaluation of event graphs. *Natural Language Engineering*, 21(4):607–652.

Goran Glavaš and Sanja Štajner. 2013. Event-centered simplification of news stories. In *Proceedings of the Student Research Workshop associated with RANLP 2013*, pages 71–78, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

Goran Glavaš and Sanja Štajner. 2015. Simplifying lexical simplification: Do we need simplified corpora? In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 63–68, Beijing, China. Association for Computational Linguistics.

References

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2013. 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. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Alistair Moffat, and Justin Zobel. 2017. Can machine translation systems be evaluated by the crowd alone. *Natural Language Engineering*, 23(1):3–30.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640, Berlin, Germany. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. 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. Association for Computational Linguistics.

Eva Hasler, Adri de Gispert, Felix Stahlberg, Aurelien Waite, and Bill Byrne. 2017. Source sentence simplification for statistical machine translation. *Computer Speech & Language*, 45(C):221–235.

Michael Heilman and Noah A. Smith. 2010. Extracting simplified statements for factual question generation. In *Proceedings of the Third Workshop on Question Generation*, pages 11–20, Pittsburgh, PA.

Daniel Hershcovich, Omri Abend, and Ari Rappoport. 2017. 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. Association for Computational Linguistics.

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

Colby Horn, Cathryn Manduca, and David Kauchak. 2014. Learning a lexical simplifier using Wikipedia. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 458–463, Baltimore, Maryland. Association for Computational Linguistics.

Kengo Hotate, Masahiro Kaneko, Satoru Katsumata, and Mamoru Komachi. 2019. Controlling grammatical error correction using word edit rate. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 149–154, Florence, Italy. Association for Computational Linguistics.

David M. Howcroft and Vera Demberg. 2017. Psycholinguistic models of sentence processing improve sentence readability ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 958–968, Valencia, Spain. Association for Computational Linguistics.

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

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

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. 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. Association for Computational Linguistics.

Hongyan Jing. 2000. Sentence reduction for automatic text summarization. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, ANLC '00, pages 310–315, Stroudsburg, PA, USA. Association for Computational Linguistics.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Tomoyuki Kajiwara and Mamoru Komachi. 2016. 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. The COLING 2016 Organizing Committee.

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

Mert Kilickaya, Aykut Erdem, Nazli Ikizler-Cinbis, and Erkut Erdem. 2017. Re-evaluating automatic metrics for image captioning. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 199–209, Valencia, Spain. Association for Computational Linguistics.

J.P. Kincaid, R.P. Fishburne, R.L. Rogers, and B.S. Chissom. 1975. 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. 49 p.

References

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

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. 2007. 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. Association for Computational Linguistics.

Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-weighted loss and diverse reranking for sentence simplification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3137–3147, Minneapolis, Minnesota. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. 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. Association for Computational Linguistics.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

J. Richard Landis and Gary G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159–174.

Walter S. Lasecki, Luz Rello, and Jeffrey P. Bigham. 2015. Measuring text simplification with the crowd. In *Proceedings of the 12th Web for All Conference*, W4A '15, New York, NY, USA. Association for Computing Machinery.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

VI Levenshtein. 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10:707.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Edward Loper and Steven Bird. 2002. NLTK: the natural language toolkit. *CoRR*, cs.CL/0205028.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. In *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Qingsong Ma, Ondřej Bojar, and Yvette Graham. 2018. Results of the WMT18 metrics shared task: Both characters and embeddings achieve good performance. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 671–688, Belgium, Brussels. Association for Computational Linguistics.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Jonathan Mallinson, Rico Sennrich, and Mirella Lapata. 2017. Paraphrasing revisited with neural machine translation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 881–893. Association for Computational Linguistics.

Angrosh Mandya, Tadashi Nomoto, and Advaith Siddharthan. 2014. Lexico-syntactic text simplification and compression with typed dependencies. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1996–2006, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Kelly Marchisio, Jialiang Guo, Cheng-I Lai, and Philipp Koehn. 2019. Controlling the reading level of machine translation output. In *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, pages 193–203, Dublin, Ireland. European Association for Machine Translation.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Samuel Humeau, Pierre-Emmanuel Mazaré, Éric de La Clergerie, Antoine Bordes, and Benoît Sagot. 2018. Reference-less quality estimation of text simplification systems. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 29–38, Tilburg, the Netherlands. ACL.

Jana M. Mason and Janet R. Kendall. 1978. Facilitating reading comprehension through text structure manipulation. Technical Report 92, Bolt, Beranek and Newman, Inc., Cambridge, Mass.; Illinois Univ., Urbana. Center for the Study of Reading. 36 p.

References

Nitika Mathur, Timothy Baldwin, and Trevor Cohn. 2020. Tangled up in BLEU: Reevaluating the evaluation of automatic machine translation evaluation metrics. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4984–4997, Online. Association for Computational Linguistics.

Danielle S. McNamara, Arthur C. Graesser, Philip M. McCarthy, and Zhiqiang Cai. 2014. *Automated Evaluation of Text and Discourse with Coh-Metrix*. Cambridge University Press, New York, NY, USA.

Paul Mcnamee and James Mayfield. 2004. Character n-gram tokenization for european language text retrieval. *Information Retrieval*, 7(1-2):73–97.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations (ICLR 2013)*, Scottsdale, Arizona, USA.

Shachar Mirkin, Sriram Venkatapathy, and Marc Dymetman. 2013. Confidence-driven Rewriting for Improved Translation. In *XIV MT Summit*, pages 257–264, Nice.

Kshitij Mishra, Ankush Soni, Rahul Sharma, and Dipti Sharma. 2014. Exploring the effects of sentence simplification on hindi to english machine translation system. In *Proceedings of the Workshop on Automatic Text Simplification - Methods and Applications in the Multilingual Society (ATS-MA 2014)*, pages 21–29, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796, Beijing, China. Coling 2010 Organizing Committee.

Tsendsuren Munkhdalai and Hong Yu. 2017. Neural semantic encoders. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 397–407, Valencia, Spain. Association for Computational Linguistics.

Courtney Napoles and Mark Dredze. 2010. Learning simple Wikipedia: A cogitation in ascertaining abecedarian language. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics and Writing: Writing Processes and Authoring Aids*, pages 42–50, Los Angeles, CA, USA. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2014. 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. Association for Computational Linguistics.

Shashi Narayan and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference*, pages 111–120, Edinburgh, UK. Association for Computational Linguistics.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Christina Niklaus, Bernhard Bermeitinger, Siegfried Handschuh, and André Freitas. 2016. A sentence simplification system for improving relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 170–174, Osaka, Japan. The COLING 2016 Organizing Committee.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Jekaterina Novikova, Ondřej Dušek, Amanda Cercas Curry, and Verena Rieser. 2017. 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. Association for Computational Linguistics.

Franz Josef Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Charles Kay Ogden. 1930. *Basic English: A General Introduction with Rules and Grammar*. Kegan Paul, Trench, Trubner & Co.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. 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. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2013. Text simplification as tree transduction. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, STIL, pages 116–125, Fortaleza, Brazil.

Gustavo Paetzold and Lucia Specia. 2016a. SemEval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569, San Diego, California. Association for Computational Linguistics.

Gustavo Paetzold and Lucia Specia. 2017a. Lexical simplification with neural ranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 34–40, Valencia, Spain. Association for Computational Linguistics.

References

Gustavo H. Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. Massalign: Alignment and annotation of comparable documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4, Tapei, Taiwan. Association for Computational Linguistics.

Gustavo H. Paetzold and Lucia Specia. 2016b. Unsupervised lexical simplification for non-native speakers. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, page 3761–3767, Phoenix, Arizona. AAAI Press.

Gustavo H. Paetzold and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Gustavo Henrique Paetzold. 2016. *Lexical Simplification for Non-Native English Speakers*. Ph.D. thesis, University of Sheffield, Sheffield, UK.

Gustavo Henrique Paetzold and Lucia Specia. 2016c. Vicinity-driven paragraph and sentence alignment for comparable corpora. *CoRR*, abs/1612.04113.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. 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. ACL.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple ppdb: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148, Berlin, Germany. Association for Computational Linguistics.

Ellie Pavlick and Joel Tetreault. 2016. An empirical analysis of formality in online communication. *Transactions of the Association for Computational Linguistics*, 4:61–74.

David Pellow and Maxine Eskenazi. 2014a. 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. Association for Computational Linguistics.

David Pellow and Maxine Eskenazi. 2014b. 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.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Sarah E. Petersen. 2007. *Natural Language Processing Tools for Reading Level Assessment and Text Simplification for Bilingual Education*. Ph.D. thesis, University of Washington, Seattle, WA, USA. AAI3275902.

Sarah E. Petersen and Mari Ostendorf. 2007. 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. Carnegie Mellon University and ISCA Archive.

Maxime Peyrard. 2019. 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. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

Matt Post, Juri Ganitkevitch, Luke Orland, Jonathan Weese, Yuan Cao, and Chris Callison-Burch. 2013. Joshua 5.0: Sparser, better, faster, server. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 206–212, Sofia, Bulgaria. Association for Computational Linguistics.

S. P. Quigley, D. Power, and M. Steinkamp. 1977. The language structure of deaf children. *The Volta Review*, 79(2):73–84.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *Proceedings of the 4th International Conference on Learning Representations (ICLR 2016)*, San Juan, Puerto Rico.

Ehud Reiter. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Ehud Reiter and Anja Belz. 2009. An investigation into the validity of some metrics for automatically evaluating natural language generation systems. *Computational Linguistics*, 35(4):529–558.

Luz Rello, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion. 2013a. Frequent words improve readability and short words improve understandability for people with dyslexia. In Paula Kotzé, Gary Marsden, Gitte Lindgaard, Janet Wesson, and Marco Winckler, editors, *Human-Computer Interaction – INTERACT 2013: 14th IFIP TC 13 International Conference, Cape Town, South Africa, September 2-6, 2013, Proceedings, Part IV*, pages 203–219. Springer Berlin Heidelberg, Berlin, Heidelberg.

Luz Rello, Clara Bayarri, Azuki Gòrriz, Ricardo Baeza-Yates, Saurabh Gupta, Gaurang Kanvinde, Horacio Saggion, Stefan Bott, Roberto Carlini, and Vasile Topac. 2013b. "dyswebxia 2.0!: More accessible text for people with dyslexia". In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*, W4A '13, pages 25:1–25:2, Rio de Janeiro, Brazil. ACM.

N. L. Robbins and C. Hatcher. 1981. The effects of syntax on the reading comprehension of hearing-impaired children. *The Volta Review*, 83(2):105–115.

Masoud Jalili Sabet, Philipp Dufter, and Hinrich Schütze. 2020. SimAlign: High Quality Word Alignments without Parallel Training Data using Static and Contextualized Embeddings. *arXiv e-prints*.

References

Carolina Scarton, Pranava Madhyastha, and Lucia Specia. 2020. Deciding when, how and for whom to simplify. In *Proceedings of the 24th European Conference on Artificial Intelligence (ECAI)*, Santiago de Compostela, Spain.

Carolina Scarton, Gustavo H. Paetzold, and Lucia Specia. 2018a. Simpa: A sentence-level simplification corpus for the public administration domain. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 4333–4338, Miyazaki, Japan. European Language Resources Association (ELRA).

Carolina Scarton, Gustavo H. Paetzold, and Lucia Specia. 2018b. Text simplification from professionally produced corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3504–3510, Miyazaki, Japan. European Language Resources Association (ELRA).

Carolina Scarton, Alessio Palmero Aprosio, Sara Tonelli, Tamara Martín Wanton, and Lucia Specia. 2017. MUSST: A multilingual syntactic simplification tool. In *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 25–28, Tapei, Taiwan. Association for Computational Linguistics.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Max Schwarzer and David Kauchak. 2018. Human evaluation for text simplification: The simplicity-adequacy tradeoff. In *Southern California Natural Language Processing Symposium*.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. 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, Vancouver, Canada. Association for Computational Linguistics.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel L´aubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 65–68, Valencia, Spain. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications(IJACSA), Special Issue on Natural Language Processing 2014*, 4(1).

Matthew Shardlow, Michael Cooper, and Marcos Zampieri. 2020. CompLex — a new corpus for lexical complexity prediction from Likert Scale data. In *Proceedings of the 1st Workshop on Tools and Resources to Empower People with REAding DIfficulties (READI)*, pages 57–62, Marseille, France. European Language Resources Association.

Cynthia M. Shewan. 1985. Auditory comprehension problems in adult aphasic individuals. *Human Communication Canada*, 9(5):151–155.

Chenglei Si, Kui Wu, Ai Ti Aw, and Min-Yen Kan. 2019. Sentiment aware neural machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 200–206, Hong Kong, China. Association for Computational Linguistics.

Advaith Siddharthan. 2003. Preserving Discourse Structure when Simplifying Text. In *Proceedings of the 2003 European Natural Language Generation Workshop*, ENLG 2003, pages 103–110, Budapest, Hungary.

Advaith Siddharthan. 2011. Text simplification using typed dependencies: A comparison of the robustness of different generation strategies. In *Proceedings of the 13th European Workshop on Natural Language Generation*, ENLG '11, pages 2–11, Stroudsburg, PA, USA. Association for Computational Linguistics.

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

Advaith Siddharthan and Angrosh Mandya. 2014. Hybrid text simplification using synchronous dependency grammars with hand-written and automatically harvested rules. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 722–731, Gothenburg, Sweden. Association for Computational Linguistics.

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

Sara Botelho Silveira and António Branco. 2012. Enhancing multi-document summaries with sentence simplificatio. In *Proceedings of the 14th International Conference on Artificial Intelligence*, ICAI 2012, pages 742–748, Las Vegas, USA.

Simple Wikipedia. 2017a. Wikipedia:How to write Simple English pages. From `https://simple.wikipedia.org/wiki/Wikipedia:How_to_write_Simple_English_pages`. Retrieved January 23, 2017.

Simple Wikipedia. 2017b. Wikipedia:Simple English Wikipedia. From `https://simple.wikipedia.org/wiki/Wikipedia:Simple_English_Wikipedia`. Retrieved January 23, 2017.

David A. Smith and Jason Eisner. 2006. Quasi-synchronous grammars: Alignment by soft projection of syntactic dependencies. In *Proceedings of the Workshop on Statistical Machine Translation*, StatMT '06, pages 23–30, New York City, New York. ACL.

References

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

Lucia Specia. 2010. Translating from complex to simplified sentences. In *Proceedings of the 9th International Conference on Computational Processing of the Portuguese Language*, PROPOR'10, pages 30–39, Porto Alegre, RS, Brazil. Springer-Verlag.

Lucia Specia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Lúcia Specia, Sandra Maria Aluísio, and Thiago A. Salgueiro Pardo. 2008. Manual de simplificação sintática para o português. Technical Report NILC-TR-08-06, NILC–ICMC–USP, São Carlos, SP, Brasil. Available in `http://www.nilc.icmc.usp.br/nilc/download/NILC_TR_08_06.pdf`.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018a. BLEU is not suitable for the evaluation of text simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 738–744, Brussels, Belgium. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018b. 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. Association for Computational Linguistics.

Elior Sulem, Omri Abend, and Ari Rappoport. 2018c. Simple and effective text simplification using semantic and neural methods. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 162–173, Melbourne, Australia. Association for Computational Linguistics.

Md Sultan, Steven Bethard, and Tamara Sumner. 2014. Back to basics for monolingual alignment: Exploiting word similarity and contextual evidence. *Transactions of the Association for Computational Linguistics*, 2:219–230.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual smt system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 38–42, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2014a. Assessing the relative reading level of sentence pairs for text simplification. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 288–297, Gothenburg, Sweden. Association for Computational Linguistics.

Sowmya Vajjala and Detmar Meurers. 2014b. Readability assessment for text simplification: From analysing documents to identifying sentential simplifications. *ITL - International Journal of Applied Linguistics*, 165(2):194–222.

Sowmya Vajjala and Detmar Meurers. 2015. Readability-based sentence ranking for evaluating text simplification. Technical report, Iowa State University.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Information Processing and Management*, 43(6):1606–1618.

Eva Vanmassenhove, Christian Hardmeier, and Andy Way. 2018. Getting gender right in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3003–3008, Brussels, Belgium. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. 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.

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

Tu Vu, Baotian Hu, Tsendsuren Munkhdalai, and Hong Yu. 2018. Sentence simplification with memory-augmented neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 79–85, New Orleans, Louisiana. Association for Computational Linguistics.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. 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. ACM.

John Wieting and Kevin Gimpel. 2018. 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. Association for Computational Linguistics.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. 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. Association for Computational Linguistics.

Evan James Williams. 1959. *Regression Analysis*, volume 14. Wiley, New York.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Mach. Learn.*, 8(3-4):229–256.

Kristian Woodsend and Mirella Lapata. 2011a. 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. Association for Computational Linguistics.

Kristian Woodsend and Mirella Lapata. 2011b. WikiSimple: Automatic Simplification of Wikipedia Articles. In *Proceedings of the 25th National Conference on Artificial Intelligence*, pages 927–932, San Francisco, CA.

Y. Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, G. Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. S. Corrado, Macduff Hughes, and J. Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv e-prints*.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. 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. Association for Computational Linguistics.

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

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics*, ACL '01, pages 523–530, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

Taha Yasseri, András Kornai, and János Kertész. 2012. A practical approach to language complexity: A wikipedia case study. *PLOS ONE*, 7(11):1–8.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. 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. Association for Computational Linguistics.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017a. CWIG3G2 - complex word identification task across three text genres and two user groups. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 401–407, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Seid Muhie Yimam, Sanja Štajner, Martin Riedl, and Chris Biemann. 2017b. Multilingual and cross-lingual complex word identification. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 813–822, Varna, Bulgaria. INCOMA Ltd.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 595–605, Copenhagen, Denmark. Association for Computational Linguistics.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3164–3173, Brussels, Belgium. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. 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. Association for Computational Linguistics.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 30–34, San Diego, California. Association for Computational Linguistics.

Sanja Štajner, Marc Franco-Salvador, Simone Paolo Ponzetto, Paolo Rosso, and Heiner Stuckenschmidt. 2017. Sentence alignment methods for improving text simplification systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 97–102, Vancouver, Canada. Association for Computational Linguistics.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. CATS: A tool for customized alignment of text simplification corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, pages 3895–3903, Miyazaki, Japan. European Language Resources Association (ELRA).

References

Sanja Štajner and Goran Glavaš. 2017. Leveraging event-based semantics for automated text simplification. *Expert Systems with Applications*, 82:383 – 395.

Sanja Štajner, Ruslan Mitkov, and Horacio Saggion. 2014. One step closer to automatic evaluation of text simplification systems. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 1–10, Gothenburg, Sweden. Association for Computational Linguistics.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242, Riga, Latvia.

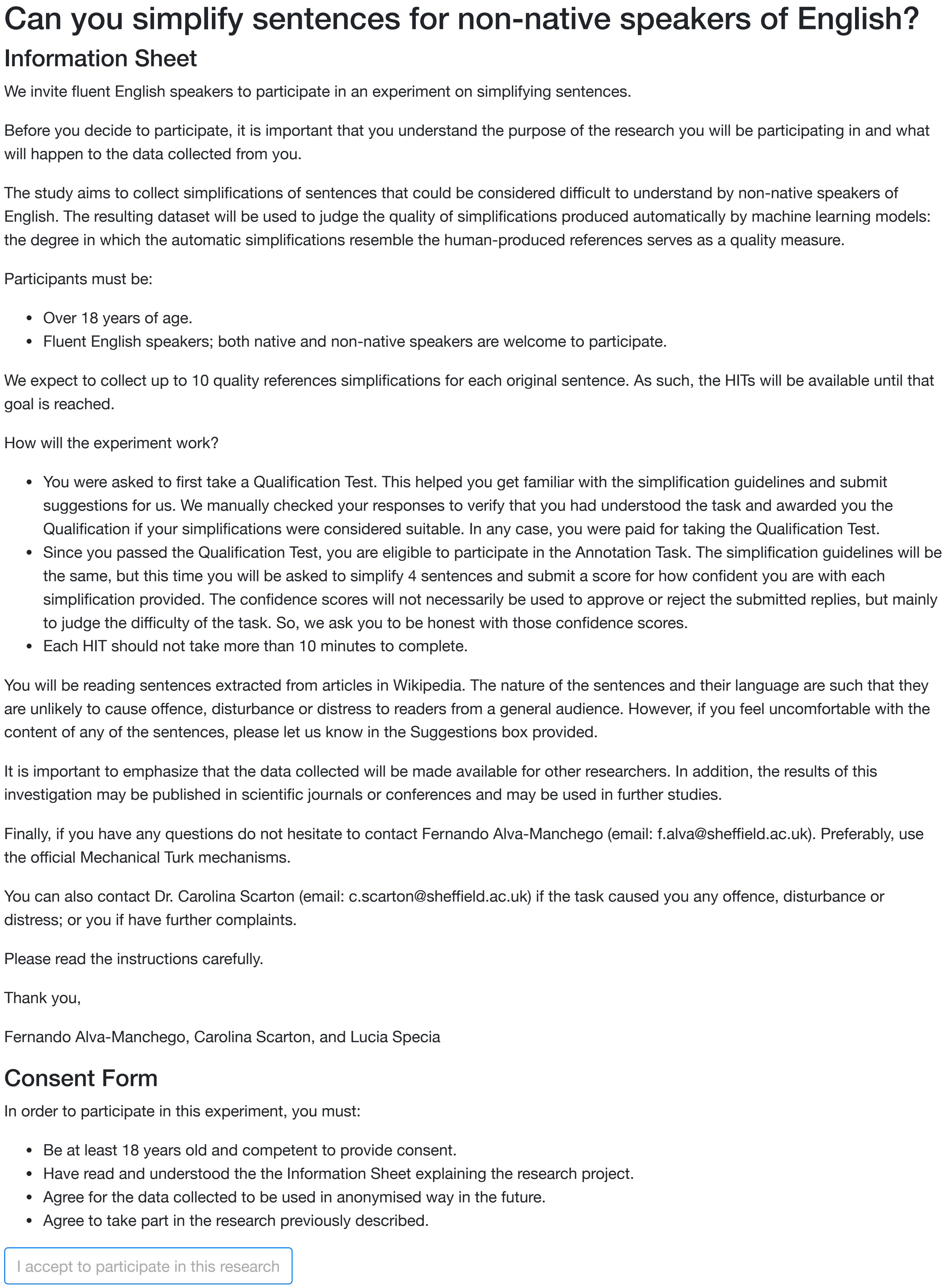
Sanja Štajner, Maja Popović, and Hannah Béchera. 2016a. Quality estimation for text simplification. In *Proceeding of the Workshop on Quality Assessment for Text Simplification - LREC 2016*, QATS 2016, pages 15–21, Portorož, Slovenia. European Language Resources Association (ELRA).

Sanja Štajner, Maja Popović, Horacio Saggion, Lucia Specia, and Mark Fishel. 2016b. 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. European Language Resources Association (ELRA).

Appendix A

Consent Forms for Participation in Data Collection for ASSET

Consent forms that AMT workers had to “sign” before participating in the corresponding tasks during the data collection for ASSET.



Can you simplify sentences for non-native speakers of English?

Information Sheet

We invite fluent English speakers to participate in an experiment on simplifying sentences.

Before you decide to participate, it is important that you understand the purpose of the research you will be participating in and what will happen to the data collected from you.

The study aims to collect simplifications of sentences that could be considered difficult to understand by non-native speakers of English. The resulting dataset will be used to judge the quality of simplifications produced automatically by machine learning models: the degree in which the automatic simplifications resemble the human-produced references serves as a quality measure.

Participants must be:

- Over 18 years of age.
- Fluent English speakers; both native and non-native speakers are welcome to participate.

We expect to collect up to 10 quality references simplifications for each original sentence. As such, the HITs will be available until that goal is reached.

How will the experiment work?

- You were asked to first take a Qualification Test. This helped you get familiar with the simplification guidelines and submit suggestions for us. We manually checked your responses to verify that you had understood the task and awarded you the Qualification if your simplifications were considered suitable. In any case, you were paid for taking the Qualification Test.
- Since you passed the Qualification Test, you are eligible to participate in the Annotation Task. The simplification guidelines will be the same, but this time you will be asked to simplify 4 sentences and submit a score for how confident you are with each simplification provided. The confidence scores will not necessarily be used to approve or reject the submitted replies, but mainly to judge the difficulty of the task. So, we ask you to be honest with those confidence scores.
- Each HIT should not take more than 10 minutes to complete.

You will be reading sentences extracted from articles in Wikipedia. The nature of the sentences and their language are such that they are unlikely to cause offence, disturbance or distress to readers from a general audience. However, if you feel uncomfortable with the content of any of the sentences, please let us know in the Suggestions box provided.

It is important to emphasize that the data collected will be made available for other researchers. In addition, the results of this investigation may be published in scientific journals or conferences and may be used in further studies.

Finally, if you have any questions do not hesitate to contact Fernando Alva-Manchego (email: f.alva@sheffield.ac.uk). Preferably, use the official Mechanical Turk mechanisms.

You can also contact Dr. Carolina Scarton (email: c.scarton@sheffield.ac.uk) if the task caused you any offence, disturbance or distress; or you if have further complaints.

Please read the instructions carefully.

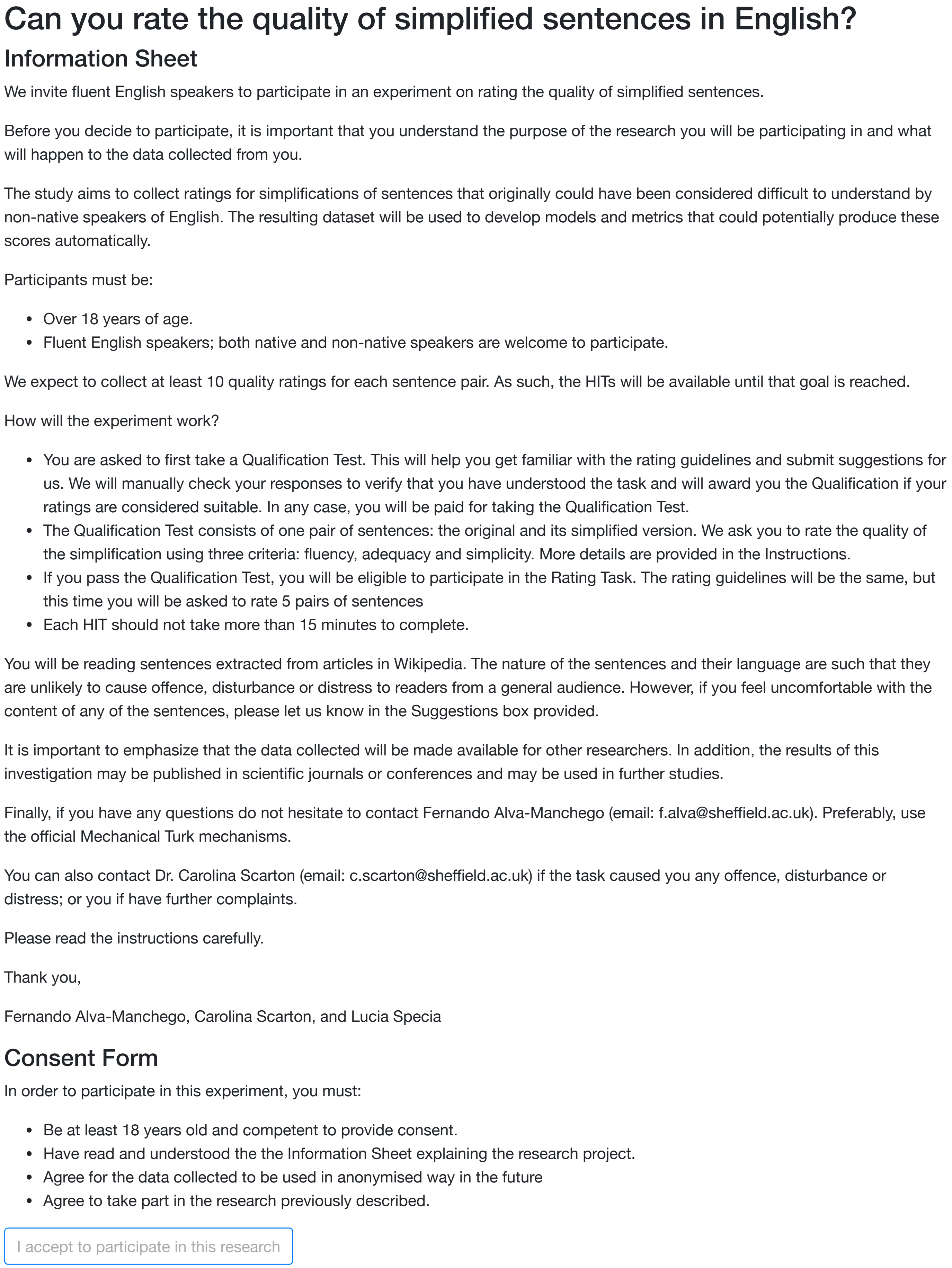
Thank you,

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia

Consent Form

In order to participate in this experiment, you must:

- Be at least 18 years old and competent to provide consent.
- Have read and understood the the Information Sheet explaining the research project.
- Agree for the data collected to be used in anonymised way in the future.
- Agree to take part in the research previously described.



Can you rate the quality of simplified sentences in English?

Information Sheet

We invite fluent English speakers to participate in an experiment on rating the quality of simplified sentences.

Before you decide to participate, it is important that you understand the purpose of the research you will be participating in and what will happen to the data collected from you.

The study aims to collect ratings for simplifications of sentences that originally could have been considered difficult to understand by non-native speakers of English. The resulting dataset will be used to develop models and metrics that could potentially produce these scores automatically.

Participants must be:

- Over 18 years of age.
- Fluent English speakers; both native and non-native speakers are welcome to participate.

We expect to collect at least 10 quality ratings for each sentence pair. As such, the HITs will be available until that goal is reached.

How will the experiment work?

- You are asked to first take a Qualification Test. This will help you get familiar with the rating guidelines and submit suggestions for us. We will manually check your responses to verify that you have understood the task and will award you the Qualification if your ratings are considered suitable. In any case, you will be paid for taking the Qualification Test.
- The Qualification Test consists of one pair of sentences: the original and its simplified version. We ask you to rate the quality of the simplification using three criteria: fluency, adequacy and simplicity. More details are provided in the Instructions.
- If you pass the Qualification Test, you will be eligible to participate in the Rating Task. The rating guidelines will be the same, but this time you will be asked to rate 5 pairs of sentences
- Each HIT should not take more than 15 minutes to complete.

You will be reading sentences extracted from articles in Wikipedia. The nature of the sentences and their language are such that they are unlikely to cause offence, disturbance or distress to readers from a general audience. However, if you feel uncomfortable with the content of any of the sentences, please let us know in the Suggestions box provided.

It is important to emphasize that the data collected will be made available for other researchers. In addition, the results of this investigation may be published in scientific journals or conferences and may be used in further studies.

Finally, if you have any questions do not hesitate to contact Fernando Alva-Manchego (email: f.alva@sheffield.ac.uk). Preferably, use the official Mechanical Turk mechanisms.

You can also contact Dr. Carolina Scarton (email: c.scarton@sheffield.ac.uk) if the task caused you any offence, disturbance or distress; or you if have further complaints.

Please read the instructions carefully.

Thank you,

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia

Consent Form

In order to participate in this experiment, you must:

- Be at least 18 years old and competent to provide consent.
- Have read and understood the the Information Sheet explaining the research project.
- Agree for the data collected to be used in anonymised way in the future
- Agree to take part in the research previously described.

Appendix B

Simplification Instructions for Data Collection in ASSET

Instructions given to AMT workers for both the Qualification Test and Simplification Task, during the collection of manual simplification references for ASSET.

Simplification Instructions

We ask you to rewrite each original sentence in order to make it easier to understand by non-native speakers of English. You can do so by replacing complex words with simpler synonyms (i.e. paraphrasing), deleting unimportant information (i.e. compression), and/or splitting a long complex sentence into several simpler ones. The final simplified sentences need to be grammatical, fluent, and retain the main ideas of their original counterparts without altering their meanings.

DETAILS:

You are given **four sentences** that need to be rewritten so that they use **simpler English**. This means that you should reduce the number of difficult words or idioms, simplify complex phrasing, delete information that may not be relevant, and make the sentence more straight-forward. This could be accomplished by applying different transformations to the original sentences. In this task, we ask you to use **paraphrasing, compression and/or sentence splitting**. We explain each of these transformations below and provide examples.

- Paraphrasing
This involves changing complex words or phrases for simpler synonyms. For example:

Original	**Simplification**
From **its inception**, it was **designated** a duty-free port **and vied** with the neighbouring Sultanate of Pattani for trade.	From **the start**, it was **chosen to be** a duty-free port **to compete** with the neighbouring Sultanate of Pattani for trade.

*the bold facing here is for ease of readability and will not be used in the task.

When paraphrasing, try to preserve as much of the original meaning as possible. Proper names (e.g. John, Microsoft, Apple), geographical locations (e.g. Northern Europe, Sultanate of Pattani), specialized technical terms (e.g. polymorphism, electro-pneumatic) or any word you don’t know should be kept whenever possible. You should use simpler concepts/words/phrases in your paraphrasing. This means your sentence could potentially have more words than the original.

- Compression
This involves deleting information that may not be relevant to understand the main idea of the sentence. For example:

Original	**Simplification**
Think of all the ways everyone in your household will benefit from your membership in Audubon.	Everyone in your household will benefit from membership in Audubon.
Mr Usta was examined by Dr Raymond Crockett, a **Harley Street** physician specialising in kidney disease.	Mr Usta was examined by Dr Raymond Crockett, a physician specialising in kidney disease.

*the bold facing here is for ease of readability and will not be used in the task.

When compressing, it is important that the resulting simplification does not alter the original meaning. Also, it is possible to make other small changes to keep the resulting sentence grammatical and fluent.

- Sentence Splitting
This involves splitting a sentence as much as possible into several sentences in places where such a splitting could make the sentence simpler. The resulting sentences, taken together, should mean the same as the original sentence, be grammatical and fluent. For example:

Original	**Simplification**
He was involved in two conversions which turned out to be crucial.	He was involved in two conversions. These conversions turned out to be crucial.
The federal government suspended sales of U.S. savings bonds because Congress hasn't lifted the ceiling on government debt.	Congress hasn't lifted the ceiling on government debt. So, the federal government suspended sales of U.S. savings bonds.

As in the example shown, when splitting, it is also possible to make other small changes to keep the resulting text grammatical. Also, if no splitting is possible in a given sentence, leave the sentence as it. The examples provided only show a few possible ways in which a sentence could be split. Feel free to try out other ways.

- Putting it all together
All the three transformations explained before should be applied whenever possible to simplify the given original sentences. We provide an example where all three transformations were performed:

Original	**Simplification**
If you are under the age of 18, you are required to complete at least 65 hours of behind-the-wheel skill-building including 10 hours of nighttime driving.	If you are under 18, you must complete 65 hours of practice driving. This must include at least 10 hours of driving at night.

The transformations performed were:

Original	**Simplification**	**Transformation**
under the age of 18	*under 18*	compression
behind-the-wheel skill-building	*practice driving*	paraphrasing
nighttime driving	*driving at night*	paraphrasing
you are required to	*you must*	paraphrasing
including	*. This must include*	splitting

We will manually check a fraction of your answers and spammers will be rejected. Read the provided instructions carefully and make sure you understand the task before beginning.

Have fun!

Appendix C

Significance Tests in Meta-Evaluation

C.1 Metrics across Simplicity Quality Levels

Simplicity-DA Dataset

Table C.1 Pearson correlations between **Simplicity-DA** judgements metrics scores computed using **ASSET**, for **low/high/all quality splits**. Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	Low	High	All
Reference-based	BERTScore$_{\text{Precision}}$	0.512	0.287	**0.617**
	BERTScore$_{\text{F1}}$	0.518	0.224	0.573
	iBLEU	0.398	0.253	0.504
	BLEU-SARI (AM)	0.417	0.239	0.503
	BERTScore$_{\text{Recall}}$	0.471	0.172	0.500
	BLEU	0.405	0.235	0.496
	BLEU-SARI (GM)	0.408	0.215	0.476
	BLEU-SARI-SAMSA (AM)	0.388	0.170	0.419
	BLEU-SAMSA (AM)	0.355	0.148	0.380
	SARI	0.336	0.139	0.359
	BLEU-SARI-SAMSA (GM)	0.355	0.075	0.328
	BLEU-SAMSA (GM)	0.324	0.074	0.300
	SARI-SAMSA (AM)	0.203	0.050	0.166
	SARI-SAMSA (GM)	0.222	0.024	0.156
	FKBLEU	0.131	0.006	0.098
Non-Reference-based	FKGL	0.272	0.093	0.117
	SAMSA	0.103	0.010	0.058

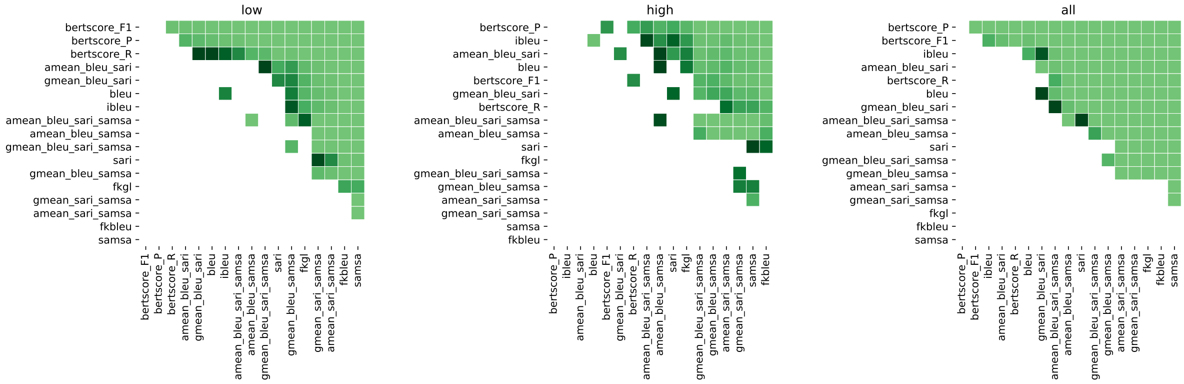


Figure C.1 Significance test for differences in correlations between metrics scores computed using **ASSET**, for **low/high/all quality splits**. Green cells denote a statistically-significant increase for the metric in a given row over the metric in a given column according to Williams test.

Simplicity Gain Dataset

Table C.2 Pearson correlations between **Simplicity Gain** human judgements and automatic metrics scores computed using manual references from **TurkCorpus**, for **low/high/all quality splits**. Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	Low	High	All
Reference-based	SARI	0.292	0.240	**0.331**
	BERTScore$_{\text{F1}}$	0.215	0.236	0.247
	BERTScore$_{\text{Recall}}$	0.221	0.217	0.241
	BERTScore$_{\text{Precision}}$	0.209	0.231	0.241
	BLEU-SARI (GM)	0.246	0.177	0.214
	BLEU-SARI (AM)	0.223	0.172	0.187
	BLEU-SARI-SAMSA (GM)	0.204	0.144	0.151
	SARI-SAMSA (GM)	0.189	0.117	0.141
	BLEU-SARI-SAMSA (AM)	0.207	0.139	0.128
	iBLEU	0.181	0.136	0.128
	BLEU	0.178	0.132	0.123
	SARI-SAMSA (AM)	0.183	0.104	0.099
	FKBLEU	0.041	0.007	0.092
	BLEU-SAMSA (AM)	0.175	0.107	0.081
	BLEU-SAMSA (GM)	0.166	0.098	0.075
Non-Reference-based	FKGL	0.045	0.101	0.147
	SAMSA	0.120	0.042	0.013

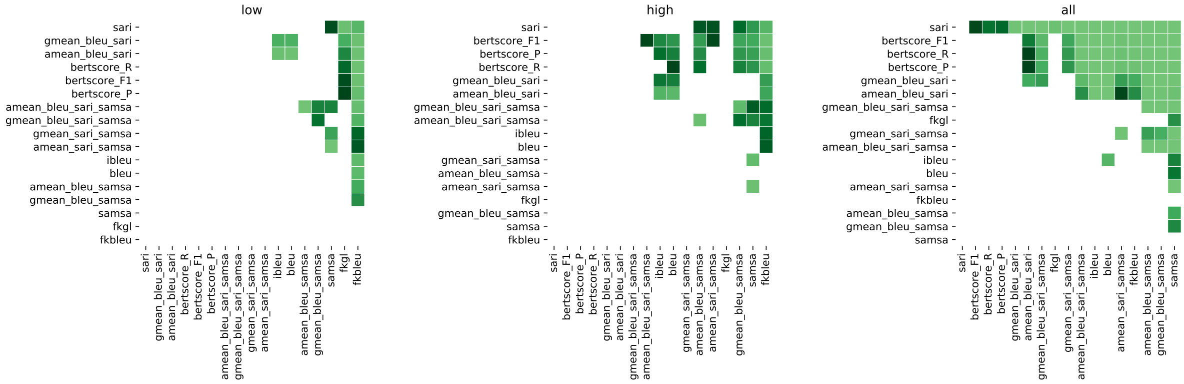


Figure C.2 Significance test results for differences in correlations between **Simplicity Gain** human judgements and automatic metrics scores computed using manual references from **TurkCorpus**, for **low/high/all quality splits**. Green cells denote a statistically-significant increase for the metric in a given row over the metric in a given column according to Williams test.

Structural Simplicity Dataset

Table C.3 Pearson correlations between **Structural Simplicity** human judgements and automatic metrics scores computed using manual references from **HSplit**, for **low/high/all quality splits**. Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	Low	High	All
Reference-based	BLEU	0.421	**0.643**	0.443
	BLEU-SARI (GM)	0.329	0.589	0.438
	iBLEU	0.408	0.635	0.436
	BLEU-SARI (AM)	0.346	0.599	0.431
	BERTScore$_{\text{Recall}}$	0.411	0.601	0.430
	BLEU-SARI-SAMSA (AM)	0.281	0.601	0.428
	BLEU-SAMSA (AM)	0.289	0.608	0.420
	BLEU-SAMSA (GM)	0.293	0.569	0.370
	BLEU-SARI-SAMSA (GM)	0.291	0.553	0.366
	FKBLEU	0.395	0.608	0.364
	SARI-SAMSA (AM)	0.140	0.495	0.347
	BERTScore$_{\text{F1}}$	0.483	0.529	0.325
	SARI	0.137	0.418	0.313
	SARI-SAMSA (GM)	0.171	0.468	0.302
	BERTScore$_{\text{Precision}}$	**0.552**	0.310	0.090
Non-Reference-based	SAMSA	0.103	0.431	0.284
	FKGL	0.070	0.165	0.228

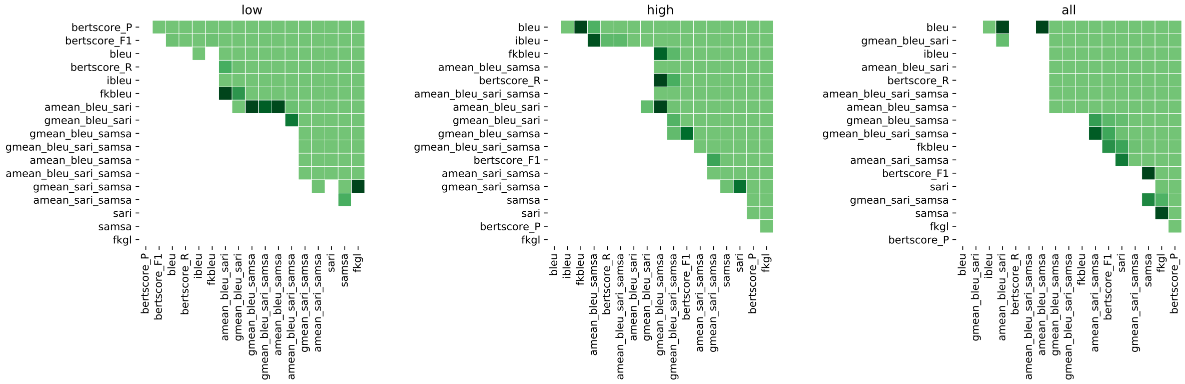


Figure C.3 Significance test results for differences in correlations between **Structural Simplicity** human judgements and automatic metrics scores computed using manual references from **HSplit**, for **low/high/all quality splits**. Green cells denote a statistically-significant increase for the metric in a given row over the metric in a given column according to Williams test.

C.2 Metrics across Types of Systems

Simplicity-DA Dataset

Table C.4 Pearson correlations between **Simplicity-DA** human judgements and automatic metrics scores computed using manual references from **ASSET**, for **system type splits**. Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

	Metric	S2S	PBMT	SBMT	Sem+PBMT
Reference-based	BERTScore$_{\text{Precision}}$	**0.650**	0.459	0.537	**0.624**
	BERTScore$_{\text{F1}}$	0.588	0.400	0.528	0.568
	IBLEU	0.558	0.355	0.306	0.329
	BERTScore$_{\text{Recall}}$	0.484	0.375	0.527	0.470
	BLEU	0.546	0.347	0.295	0.333
	BLEU-SARI (AM)	0.536	0.336	0.315	0.335
	BLEU-SARI (GM)	0.508	0.320	0.298	0.308
	BLEU-SARI-SAMSA (AM)	0.456	0.317	0.331	0.412
	BLEU-SAMSA (AM)	0.439	0.300	0.302	0.412
	BLEU-SARI-SAMSA (GM)	0.317	0.222	0.324	0.371
	SARI	0.310	0.173	0.228	0.240
	BLEU-SAMSA (GM)	0.303	0.229	0.295	0.391
	SARI-SAMSA (AM)	0.209	0.121	0.243	0.291
	SARI-SAMSA (GM)	0.190	0.080	0.250	0.333
	FKBLEU	0.092	0.058	0.006	0.138
Non-Reference-based	SAMSA	0.126	0.067	0.184	0.248
	FKGL	0.104	0.063	0.055	0.062

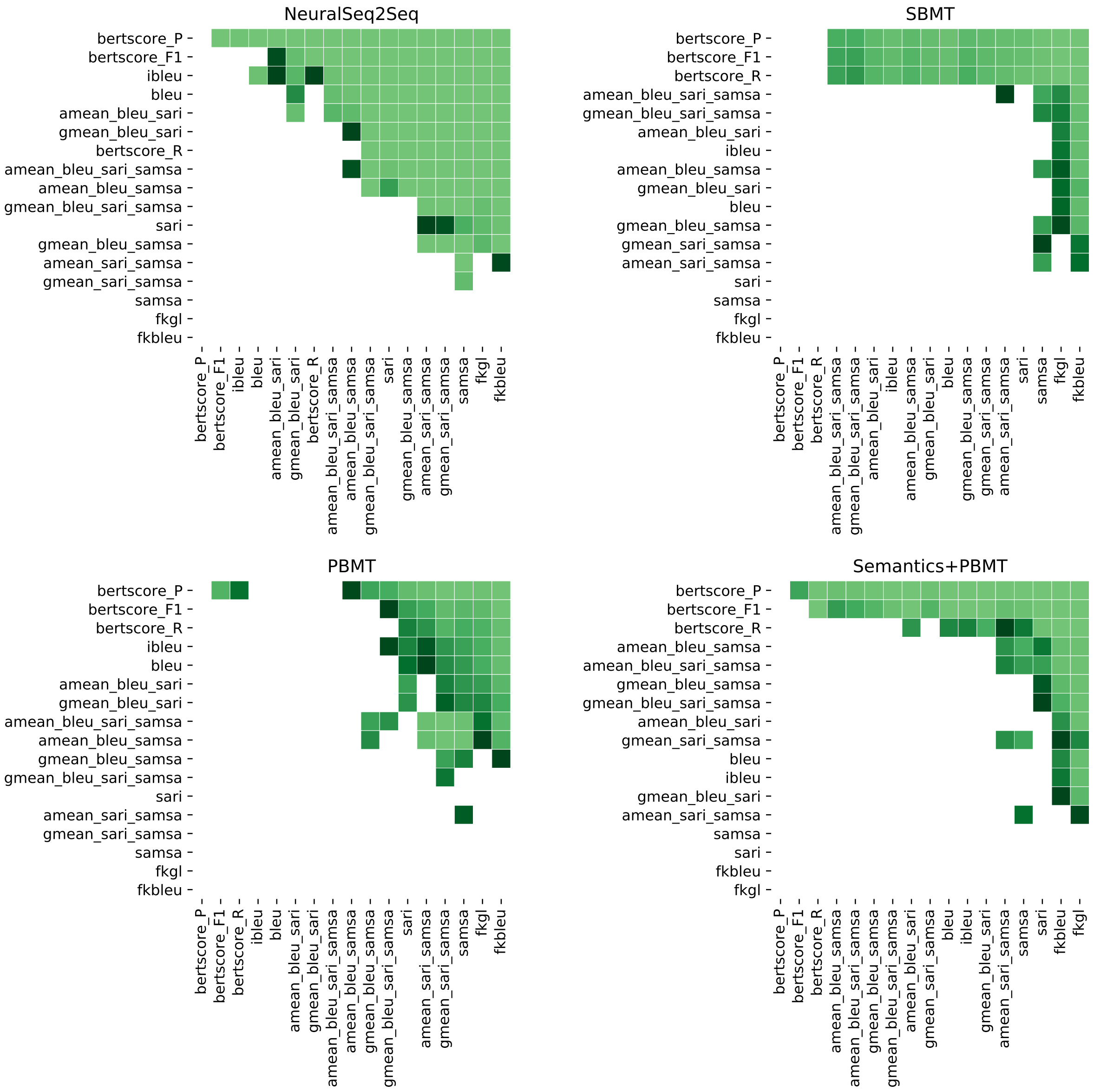


Figure C.4 Significance test results for differences in correlations between **Simplicity-DA** human judgements and automatic metrics scores computed using manual references from **ASSET**, for **system type splits**. Green cells denote a statistically-significant increase for the metric in a given row over the metric in a given column according to Williams test.

Structural Simplicity Dataset

Table C.5 Pearson correlations between **Structural Simplicity** judgements and metrics scores computed using **HSplit**, for **system type splits**. Correlations of metrics not significantly outperformed by any other in the quality split are boldfaced.

Metric	PBMT	SBMT	S2S	Sem	Sem+PBMT	Sem+S2S
BERTScore$_{\text{Precision}}$	0.501	**0.571**	0.292	**0.330**	0.096	0.111
BERTScore$_{\text{F1}}$	0.405	0.497	0.553	0.180	0.049	0.362
BLEU	0.284	0.380	0.661	0.130	0.147	0.540
iBLEU	0.252	0.380	0.642	0.130	0.145	0.536
BLEU-SARI-SAMSA (GM)	0.185	0.292	0.567	0.111	0.087	0.551
BLEU-SAMSA (GM)	0.216	0.279	0.563	0.109	0.075	0.561
BLEU-SARI (AM)	0.184	0.364	0.603	0.100	0.175	0.507
BLEU-SARI (GM)	0.157	0.341	0.589	0.097	0.185	0.515
BLEU-SAMSA (AM)	0.240	0.334	0.603	0.095	0.072	0.573
BLEU-SARI-SAMSA (AM)	0.204	0.351	0.591	0.092	0.111	0.554
SARI-SAMSA (GM)	0.123	0.253	0.446	0.081	0.025	0.505
BERTScore$_{\text{Recall}}$	0.339	0.418	0.635	0.066	0.134	0.480
SARI-SAMSA (AM)	0.121	0.271	0.425	0.056	0.063	0.502
SARI	0.015	0.286	0.330	0.028	0.166	0.355
FKBLEU	0.215	0.344	0.617	0.009	0.119	0.539
SAMSA	0.141	0.177	0.368	0.052	0.009	0.497
FKGL	0.205	0.016	0.251	0.083	0.155	0.242

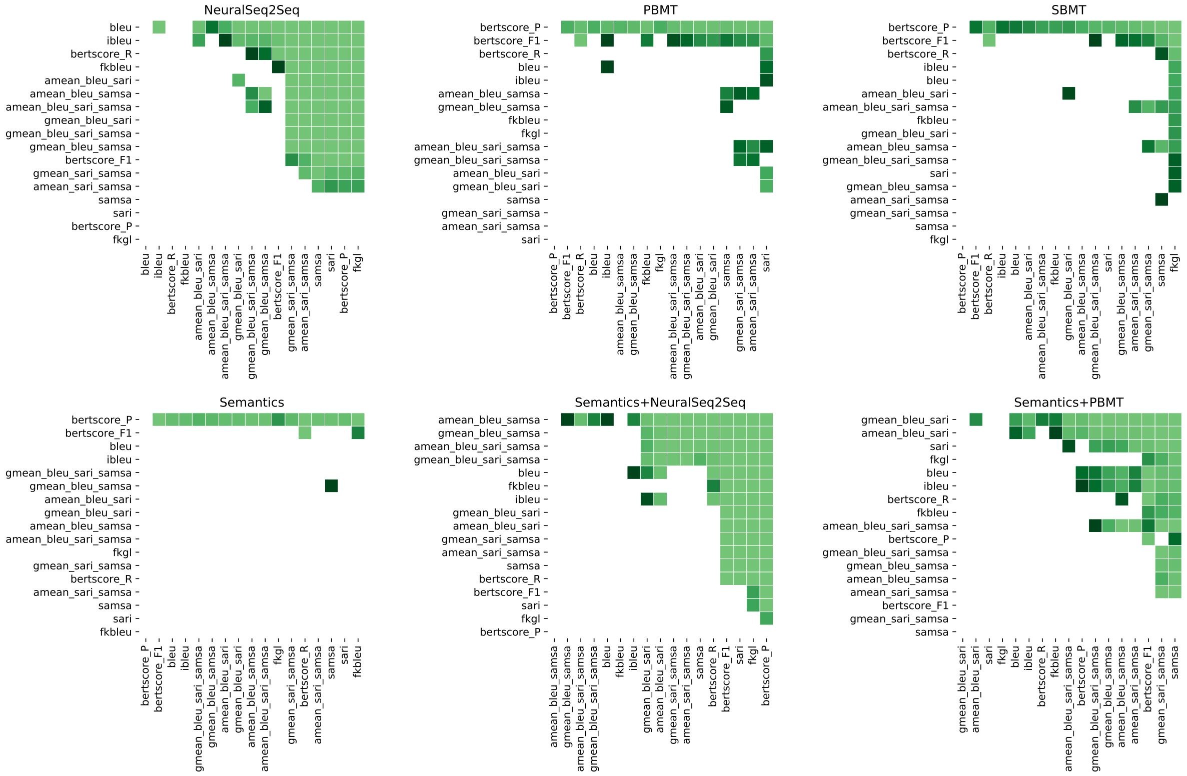


Figure C.5 Significance test results for differences in correlations between **Structural Simplicity** judgements and metrics scores computed using **HSplit**, for **system type splits**. Green cells denote a statistically-significant increase for the metric in a given row over the metric in a given column according to Williams test.

Appendix D

Training Details for MulTSS

Preprocessing. Datasets were recased and detokenised (if necessary) using Moses. This was performed to ensure that all data had the same format when being input to the models. All instances were tokenised using sentencepiece[1] before training. We set a BPE size of 15000, and removed sentences with $<$ 1 BPE tokens and $>$ 250 BPE tokens.

Training. For encoder and decoders based on Transformer, we followed Zhao et al. (2018). All singletask and multitask models had similar configurations. For instance, MulTSS was trained with the following command:

```
fairseq-train "${data_dir}/bin" \
  --ddp-backend=no_c10d \
  --task multilingual_translation
  --lang-pairs orig-simp,orig-para,orig-split,orig-comp \
  --arch multilingual_transformer \
  --share-encoders --share-decoders \
  --encoder-langtok tgt\
  --encoder-embed-path "${glove}" \
  --encoder-embed-dim 300 \
  --encoder-ffn-embed-dim 300 \
  --decoder-embed-path "${glove}" \
  --decoder-embed-dim 300 \
  --decoder-ffn-embed-dim 300 \
  --encoder-attention-heads 5 \
  --decoder-attention-heads 5 \
  --encoder-layers 4 \
  --decoder-layers 4 \
  --optimizer adam \
```

[1] https://github.com/google/sentencepiece

```
--adam-betas '(0.9, 0.98)' \
--lr 0.0005 \
--lr-scheduler inverse_sqrt \
--min-lr '1e-09' \
--label-smoothing 0.1 \
--dropout 0.3 \
--weight-decay 0.0001 \
--criterion label_smoothed_cross_entropy \
--max-epoch 30 \
--warmup-updates 4000 \
--warmup-init-lr '1e-07' \
--max-tokens 4000 --update-freq 4 \
--save-dir "${model_dir}" \
--tensorboard-logdir "${log_dir}" \
```

All singletask models were trained for 50 epoch, and MulTSS for 30. We chose the best models in the respective validation set depending on the task using the appropriate metric.