Data-Driven Sentence Simplification: Survey and Benchmark

Fernando Alva-Manchego
University of Sheffield
Department of Computer Science
f.alva@sheffield.ac.uk

Carolina Scarton
University of Sheffield
Department of Computer Science
c.scarton@sheffield.ac.uk

Lucia Specia
Imperial College London
Department of Computing
l.specia@imperial.ac.uk

Sentence Simplification (SS) aims to modify a sentence in order to make it easier to read and understand. In order to do so, several rewriting transformations can be performed such as replacement, reordering, and splitting. Executing these transformations while keeping sentences grammatical, preserving their main idea, and generating simpler output, is a challenging and still far from solved problem. In this article, we survey research on SS, focusing on approaches that attempt to learn how to simplify using corpora of aligned original-simplified sentence pairs in English, which is the dominant paradigm nowadays. We also include a benchmark of different approaches on common data sets so as to compare them and highlight their strengths and limitations. We expect that this survey will serve as a starting point for researchers interested in the task and help spark new ideas for future developments.

1. Introduction

Text Simplification (TS) is the task of modifying the content and structure of a text in order to make it easier to read and understand, while retaining its main idea and approximating its original meaning. A simplified version of a text could benefit users with several reading difficulties, such as non-native speakers (Paetzold 2016), people with aphasia (Carroll et al. 1998), dyslexia (Rello et al. 2013b), or autism (Evans, Orasan, and Dornescu 2014). Simplifying a text automatically could also help improve performance

Submission received: 8 June 2018; revise d version received: 9 August 2019; accepted for publication: 15 September 2019.

https://doi.org/10.1162/COLI_a_00370

on other language processing tasks, such as parsing (Chandrasekar, Doran, and Srinivas 1996), summarization (Vanderwende et al. 2007; Silveira and Branco 2012), information extraction (Evans 2011), semantic role labeling (Vickrey and Koller 2008), and Machine Translation (MT) (Hasler et al. 2017).

Most research on TS 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 MT. It can be argued that "true" TS (i.e., document-level) cannot be achieved by simplifying sentences one at a time, and we make a call in Section 6 for the field to move in that direction. However, because the goal of this article is to review what has been done in TS so far, our survey is limited to **Sentence Simplification** (SS).

When simplifying sentences, different rewriting transformations are performed, which range from replacing complex words or phrases for simpler synonyms, to changing the syntactic structure of the sentence (e.g., splitting or reordering components). Modern SS approaches are **data-driven**; that is, they attempt to learn these transformations using parallel corpora of aligned original-simplified sentences. This results in general simplification models that could be used for any specific type of audience, depending on the data used during training. Although 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 analyze it critically and empirically to better identify areas that could be improved.

In this article, we present a survey of research on data-driven SS for English—the dominant paradigm nowadays—and complement it with a benchmark of models whose outputs on standard data sets are publicly available. Our survey differs from other SS surveys in several aspects:

- Shardlow (2014) overviews automatic SS with short notes on different approaches for the task, whereas we provide a more in-depth explanation of the mechanics of how the simplification models are learned, and review the resources used to train and test them.
- Siddharthan (2014) focuses on the motivations for TS and mostly provides details for the earliest automatic SS approaches, which are not necessarily data-driven. We review state-of-the-art models, focusing on those that learn to rewrite a text from examples available in corpora (i.e., data-driven), leaving aside approaches based on manually constructed rules.
- Saggion (2017) introduces data-driven SS in a *Learning to Simplify* book chapter. We provide a more extensive literature review of a larger number of approaches and resources. For instance, we include models based on neural sequence-to-sequence architectures (Section 4.4).

Finally, our survey introduces a benchmark with common data sets and metrics, so as to provide an empirical comparison between different approaches. This benchmark consists of commonly used evaluation metrics and novel measures of per-transformation performance.

1.1 Motivation for Sentence Simplification

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

Motivated by the potential benefits of simplified texts, research has been dedicated to developing simplification methods for specific target audiences: writers (Candido Jr. et al. 2009), low literacy readers (Watanabe et al. 2009), English learners (Petersen 2007), non-native English speakers (Paetzold 2016), children (De Belder and Moens 2010), and people suffering from aphasia (Devlin and Tait 1998; Carroll et al. 1998), dyslexia (Rello et al. 2013b), or autism (Evans, Orasan, and Dornescu 2014). Furthermore, simplifying sentences automatically could improve performance on other Natural Language Processing tasks, which has become evident in parsing (Chandrasekar, Doran, and Srinivas 1996), summarization (Siddharthan, Nenkova, and McKeown 2004; Vanderwende et al. 2007; Silveira and Branco 2012), information extraction (Klebanov, Knight, and Marcu 2004; Evans 2011), relation extraction (Niklaus et al. 2016), semantic role labeling Vickrey and Koller 2008, and MT (Mirkin, Venkatapathy, and Dymetman 2013; Mishra et al. 2014; Štajner and Popović 2016; Hasler et al. 2017). 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 Natural Language Processing applications.

1.2 Text Transformations for Simplification

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

Petersen and Ostendorf (2007) analyzed 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 categorized as dropped (1 to 0), split (1 to ≥ 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 (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 the information it contains is.

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;

and counting prepositional phrases, adjectives, adverbs, clauses, and other features. As a result, a manual for Brazilian Portuguese SS was elaborated that contains a set of rules to perform the task (Specia, Aluísio, and Pardo 2008). In addition, as part of the same project, Caseli et al. (2009) implemented a tool to aid manual SS considering the following transformations: 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 (2011a) worked with a data set of 200 news articles in Spanish with their corresponding manual simplifications. After automatically aligning the sentences, the authors determined the simplification transformations performed: change (e.g., difficult words, pronouns, voice of verb), delete (words, phrases or clauses), insert (word or phrases), split (relative clauses, coordination, etc.), proximization (add locative phrases, change from third to second person), reorder, select, and join (sentences). The first four transformations are the most common in their corpus.

1.3 Related Text Rewriting Tasks

From the definition of simplification, the task could easily be confused with **summarization**. As Shardlow (2014) points out, summarization focuses on reducing length and content by removing unimportant or redundant information. In simplification, some deletion of content can also be performed. However, we could additionally replace words by more explanatory phrases, make co-references explicit, add connectors to improve fluency, and so forth. As a consequence, a simplified text could end up being longer than its original version while still improving the readability of the text. Therefore, although summarization and simplification are related, they have 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. As such, this could be considered as a subtask of the simplification process, which also encompasses more complex transformations. **Abstractive sentence compression** (Cohn and Lapata 2013), on the other hand, does include transformations 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. Because SS could involve deletion, it would not always be able to preserve meaning. Rather, its editing decisions may remove details that could distract the reader from understanding the text's central message. As such, split-and-rephrase could be considered as another possible text transformation within simplification.

1.4 Structure of this Article

In the remainder of this article, Section 2 details the most commonly used resources for SS, with emphasis on corpora used to train SS models. Section 3 explains how the output of a simplification model is generally evaluated. The two main contributions of this article are given in Section 4, which presents a critical summary of the different approaches that have been used to train data-driven sentence models, and Section 5, which benchmarks most of these models using common metrics and data sets to compare them and establish the advantages and disadvantages of each approach. Finally, based on the literature review and analysis presented, in Section 6 we provide directions for future research in the area.

2. Corpora for Simplification

A data-driven SS model is one that learns to simplify from examples in corpora. In particular, for learning sentence-level transformations, a model requires instances of original sentences and their corresponding simplified versions. In this section, we present the most commonly used resources for SS 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 SS research. We describe resources for English in detail and give an overview of resources available for other languages.

As presented in Section 1.2, 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 remainder of this article, we use the term **simplification instance** to refer to any type of sentence alignment in a general way.

2.1 Main - Simple English Wikipedia

The Simple English Wikipedia (SEW)[1] is a version of the online English Wikipedia (EW)[2] primarily aimed at English learners, but which 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).

2.1.1 Simplification Instances. Much of the popularity of using Wikipedia for research in SS 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 term frequency–inverse document frequency (tf-idf) cosine similarity. For the **PWKP** corpus, Zhu, Bernhard, and Gurevych (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

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

Table 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, Bernhard, and Gurevych 2010)	108K	1-to-1, 1-to-N
C&K-1 (Coster and Kauchak 2011b)	137K	1-to-1, 1-to-N
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
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*

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 automatically 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.[3] 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 neighbors 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 (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 SS. Table 1 summarizes some of their characteristics.

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

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 an SS model to learn how to split sentences.

Finally, **WikiLarge** (Zhang and Lapata 2017) joins instances from four Wikipedia-based data sets: PWKP, C&K-2, AlignedWL, and RevisionWL. It is the most common corpus used for training neural sequence-to-sequence models for SS (see Sec. 4.4). However, it is not the biggest in size currently available, and can contain noisy alignments.

2.1.2 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, Kornai, and Kertész (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 automatically produced alignments used to train SS models. Coster and Kauchak (2011b) found that in their corpus (C&K-1), the majority (65%) of simple paragraphs do not align with an original one, and even between aligned paragraphs not every sentence is aligned. Also, around 27% of instances are identical, which could induce SS models to learn to not modify an original sentence, or to perform very conservative rewriting transformations. Xu, Callison-Burch, and Napoles (2015) analyzed 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. Although instances formed by identical sentence pairs are important for learning when not to simplify, misalignments add noise to the data and prevent models from learning how to perform the task accurately.

Another line of research tries to determine the simplification transformations realized 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 transformations 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, Callison-Burch, and Napoles (2015) categorized 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 toward lexical simplification and compression operations. Also, Xu, Callison-Burch, and

Napoles (2015) state that the simplifications found are not ideal, because 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 transformations can lead to suboptimal SS 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 SS. One promising direction is to devise ways to mitigate the effects of the noise in the data.

2.2 Newsela Corpus

In order to tackle some of the problems identified in EW and SEW alignments, Xu, Callison-Burch, and Napoles (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 1) shows a better presence and distribution of simplification transformations in the Newsela corpus.

The statistics of Figure 1 show that there is still a preference toward compression and lexical substitution transformations, 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 that are not simpler than their counterparts in the previous version. This is likely to be because they did not need any further

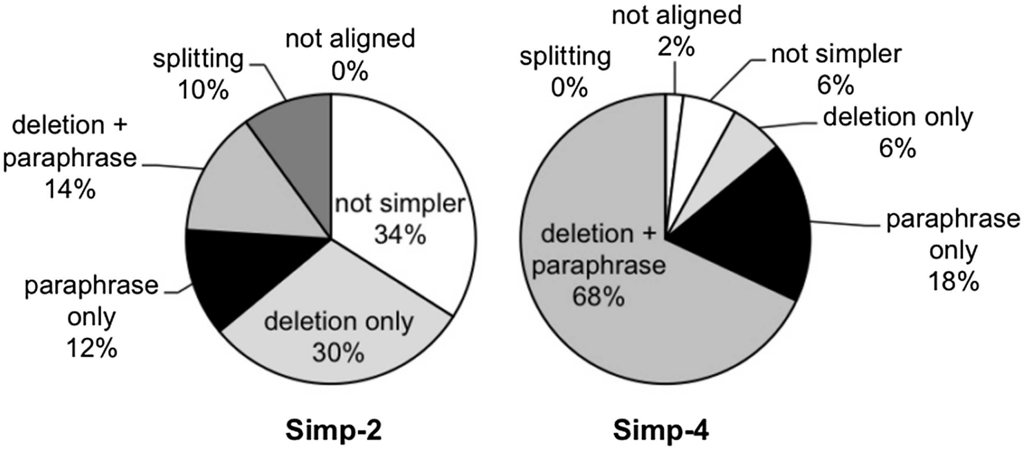


Figure 1
Manual categorization of simplification transformations in sample sentences from two simplified versions in the Newsela corpus. Simp-N means sentences from the original article (version 0) automatically aligned with sentences in version-N of the same article. Extracted from Xu, Callison-Burch, and Napoles (2015).

simplifications to comply with the readability requirements of the grade level of the current version.

Xu, Callison-Burch, and Napoles (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 its 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.

2.2.1 Simplification Instances. Newsela is a corpus that can be obtained for free for research purposes,[4] but it cannot be redistributed. As such, it is not possible to produce and release sentence alignments for the research community in SS. This is certainly a disadvantage, because it is difficult to compare SS models developed using this corpus without a common split of the data and the same document, paragraph, and sentence alignments.

Xu, Callison-Burch, and Napoles (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 judgments 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 the CATS software (Štajner et al. 2018).[5]

The vicinity-driven algorithms of Paetzold and Specia (2016) are used in Alva-Manchego et al. (2017) to generate paragraph and sentence alignments between consecutive versions of articles in the Newsela corpus. 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

[4] https://newsela.com/data/.
[5] https://github.com/neosyon/SimpTextAlign.

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, Alva-Manchego, and Specia 2017).[6]

Because 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) may learn different text transformations than those using non-consecutive versions (e.g., 0-2, 1-3, 2-4). This is important to keep in mind when learning from automatic alignments of this corpus.

2.2.2 Suitability for Simplification Research. Scarton, Paetzold, and Specia (2018b) studied automatically aligned sentences from the Newsela corpus in order to determine its suitability for SS. They first analyzed 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 data set 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 SS needs to be mindful about the types of sentence alignments to use for training models.

2.3 Other Resources for English

In this section, we describe some additional resources that are used for SS in English with very specific reasons: tuning and testing of models in general purpose (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).

2.3.1 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 2,350 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 350 instances for measuring the performance of SS models using metrics that rely on multiple references (see SARI in Sec. 3.2.3). However, the instances chosen from PWKP are those that focus on paraphrasing (1-to-1 alignments with almost similar lengths), thus limiting the range of simplification operations that SS models can be evaluated on using this multi-reference corpus. This corpus is the most commonly used to evaluate and compare SS systems trained on English Wikipedia data.

2.3.2 HSplit. Sulem, Abend, and Rappoport (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

[6] `https://github.com/ghpaetzold/massalign`.

sentence as much as possible, and (2) split only when it simplifies the original sentence. Two annotators carried out the task in both settings.

2.3.3 SimPA. Scarton, Paetzold, and Specia (2018a) introduce a corpus that differs from the previously described 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 SS models in a different domain, whereas the latter allows the analysis of the performance of SS 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.

2.3.4 OneStopEnglish. Vajjala and Lučić (2018) compiled a parallel corpus of 189 news articles that were rewritten by teachers to three levels of adult 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 generalization capabilities of models trained on bigger corpora with different target audiences.

2.3.5 WebSplit. Narayan et al. (2017) introduced split-and-rephrase, and created a data set for training and testing of models attempting this task. Extracting information from the WebNLG data set (Gardent et al. 2017), they collected WebSplit. Each entry in the data set contains: (1) a meaning representation (MR) of an original sentence, which is a set of Resource Description Framework (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 because of memorization 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 data set, with more data and following constraint (2).

2.3.6 WikiSplit. Botha et al. (2018) created a corpus for the split-and-rephrase task based on English Wikipedia edit histories. In the data set, each original sentence is only aligned with two simpler ones. A simple heuristic was used for the alignment: the trigram prefix and trigram suffix of the original sentence should match, respectively, the trigram prefix of the first simple sentence and the trigram suffix of the second simple sentence. The two simple sentences should not have the same trigram 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.

2.3.7 Paraphrase Database. Ganitkevitch, Van Durme, and Callison-Burch (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 utilized the synchronous context-free grammar formalism to collect paraphrases. Using MT 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.

2.3.8 Simple Paraphrase Database. Pavlick and Callison-Burch (2016) created the Simple PPDB, a subset of the PPDB tailored for SS. They used machine learning models to select paraphrases that generate a simplification and preserve its meaning. First, they selected 1,000 words from PPDB which also appear in the Newsela corpus. They then selected up to 10 paraphrases for each word. After that, they 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. Next, these data were used to train a multi-class logistic regression model to predict whether a paraphrase would produce simpler, more complex, or non-sense output. Finally, they applied this model to PPDB and extracted 4.5 million simplifying paraphrases.

2.4 Resources for Other Languages

The most popular (and generally larger) resources available for simplification are in English. However, some resources have been built for other languages:

- **Basque.** Gonzalez-Dios et al. (2014) collected 200 articles of science and technology texts from a science and technology magazine (complex corpus) and a Web site for children (simple corpus). They used these corpora to analyze complexity, but the articles in the data set are not parallel.
- **Brazilian Portuguese.** Caseli et al. (2009) compiled 104 newspaper articles (complex corpus), and a linguist simplified each of them following a simplification manual (Specia, Aluísio, and Pardo 2008) and an annotation editor that registers the simplification transformations performed. The corpus contains 2,116 instances.
- **Danish.** Klerke and Søgaard (2012) introduced DSim, a parallel corpus of news telegrams and their simplifications produced by trained journalists. The corpus contains 3,701 articles, out of which a total of 48,186 automatically aligned sentence pairs were selected.
- **German.** Klaper, Ebling, and Volk (2013) crawled articles from different Web sites to collect a corpus of around 7K sentences, of which close to 78% have automatic alignments.
- **Italian.** Brunato et al. (2015) collected and manually aligned two corpora. One contains 32 short novels for children and their manually simplified versions, and the other is composed of 24 texts produced and simplified by teachers. They also manually annotated the simplification transformations

performed. Tonelli, Aprosio, and Saltori (2016) introduced SIMPITIKI,[7] extracting aligned original-simplified sentences from revision histories of the Italian Wikipedia, and annotating them using the same scheme as Brunato et al. (2015). The corpus described in their paper contains 345 instances with 575 annotations of simplification tranformations. As part of SIMPITIKI, the authors also created a corpus of the Public Domain by simplifying documents from the Trento Municipality with 591 annotations.

- **Japanese.** Goto, Tanaka, and Kumano (2015) released a corpus of news articles and their simplifications, produced by teachers of Japanese as a foreign language. Their data set consists of 10,651 instances for training (automatic alignments), 723 instances for development (manual alignments), and 2,012 instances for testing (manual alignments).
- **Spanish.** Bott and Saggion (2011a) describe a corpus of 200 news articles and their simplifications, produced by trained experts and targeted at people with learning disabilities. They produced automatic sentence alignments (Bott and Saggion 2011b) and manually annotated the simplification transformations performed in only a subset of the data set. Newsela also provides a simplification corpus in Spanish which has been used in Štajner et al. (2017, 2018).

3. Evaluation of Simplification Models

The main goal in SS 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 SS models are typically evaluated, based on human ratings and/or using automatic metrics.

3.1 Human Assessment

Arguably, 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. 2016).

For **grammaticality** (sometimes referred to as *fluency*), evaluators are presented with a sentence and are asked to rate it using a Likert scale of 1–3 or 1–5 (most common). The lowest score indicates that the sentence is completely ungrammatical, whereas 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** (sometimes referred to as *adequacy*), evaluators are presented with a pair of sentences (the original and the simplification), and are 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 when compared with the original version, also using a Likert scale. Xu et al. (2016)

[7] https://github.com/dhfbk/simpitiki.

differs from this standard, asking judges to evaluate **simplicity gain**, which means to count the correct lexical and syntactic paraphrases performed. Sulem, Abend, and Rappoport (2018b) introduce the notion of **structural simplicity**, which ignores lexical simplifications and focuses on structural transformations with the question: *Is the output simpler than the input, ignoring the complexity of the words?*

3.2 Automatic Metrics

Even though human evaluation is the preferred method for assessing the quality of simplifications, they are costly to produce and may require expert annotators (linguists) or end-users of a specific target audience (e.g., children suffering from dyslexia). Therefore, researchers turn to automatic measures as a means of obtaining faster and cheaper 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; whereas others are trained on specially annotated data so as to learn to predict the quality or usefulness of the simplification being evaluated.

3.2.1 String Similarity Metrics. These metrics are mostly borrowed from the MT literature, since SS can be seen as translating a text from complex to simple. The most commonly used are BLEU and TER.

BLEU (**Bi**Lingual **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, independent 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 an 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 with the lengths of the references) could inflate this modified precision. As such, BLEU uses a Brevity Penalty (BP) factor, calculated as in Equation (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).

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

The final BLEU score is computed as in Equation (2). Traditionally, $N = 4$ and $w_n = 1/N$.

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log(p_n)\right) \tag{2}$$

In simplification research, several studies (Wubben, van den Bosch, and Krahmer 2012; Štajner, Mitkov, and Saggion 2014; Xu et al. 2016) show that BLEU has high correlation with human assessments of grammaticality and meaning preservation, but not simplicity. Also, Sulem, Abend, and Rappoport (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 SS models. In addition, because of 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), designed by Snover et al. (2006), measures the minimum number of edits necessary to change a candidate translation so that it matches perfectly to one of the references, normalized by the average length of the references. Only the reference that is closest (according to TER) is considered for the final score. The edits to be considered are insertions, deletions, substitutions of single words, and shifts (positional changes) of word sequences. TER is an edit-distance metric Equation (3), with values ranging from 0 to 100; *lower values are better*.

$$TER = \frac{\text{\# of edits}}{\text{average \# of reference words}} \tag{3}$$

In order 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 that minimizes the number of insertions, deletions, and substitutions; then (ii) calculate the optimal remaining edit distance using minimum-edit-distance and dynamic programming.

For simplification research, TER's intermediate calculations (i.e., the edits counts) have been used to show the simplification operations that an SS 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 transformations. Scarton, Paetzold, and Specia (2018b) use TER to study the differences between different simplification versions in articles of the Newsela corpus.

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 (4), with values ranging from 0 to 1 (or to 100); *higher values are better*.

$$\text{iBLEU}(s, r_s, c) = \alpha \times \text{BLEU}(c, r_s) - (1 - \alpha) \times \text{BLEU}(c, s) \tag{4}$$

After empirical evaluations, the authors recommend using a value of α between 0.7 and 0.9. For example, Mallinson, Sennrich, and Lapata (2017) experiment with a value of 0.8, while Xu et al. (2016) set it to 0.9.

3.2.2 Flesch-Based 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 (5) gives a score between 0 and 100, with *lower values indicating a higher level of difficulty*.

$$FRE = 206.835 - 1.015 \left(\frac{\text{number of words}}{\text{number of sentences}} \right) - 84.6 \left(\frac{\text{number of syllables}}{\text{number of words}} \right) \tag{5}$$

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 (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, so that *lower values indicate a lower level of difficulty*.

$$FKGL = 0.39\left(\frac{\text{number of words}}{\text{number of sentences}}\right) + 11.8\left(\frac{\text{number of syllables}}{\text{number of words}}\right) - 15.59 \tag{6}$$

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 (7); *higher values mean better simplifications*.

$$\begin{aligned} \text{FKBLEU} &= \text{iBLEU}(I, R, O) \times \text{FKGLdiff}(I, O) \\ \text{FKGLdiff} &= \text{sigmod}(\text{FKGL}(O) - \text{FKGL}(I)) \end{aligned} \tag{7}$$

Because of the way these Flesch-based metrics are computed, short sentences could obtain 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 SS models (Wubben, van den Bosch, and Krahmer 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.

3.2.3 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 an SS 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 judgments of simplicity gain. As such, this metric has become the standard measure for evaluating and comparing SS 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):

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

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

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

$$p_{del}(n) = \frac{\sum_{g \in I} \min(\#_g(I \cap \bar{O}), \#_g(I \cap \overline{R'}))}{\sum_{g \in I} \#_g(I \cap \bar{O})} \; \#_g(I \cap \bar{O}) = \max(\#_g(I) - \#_g(O), 0)$$

$$\#_g(I \cap \overline{R'}) = \max(\#_g(I) - \#_g(R)/r, 0)$$

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 (8).

$$SARI = d_1 F_{add} + d_2 F_{keep} + d_3 F_{del} \tag{8}$$

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, which only ponders the similarity of the output with the references. Although 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 transformations taken into account, restricting the evaluation to only 1-to-1 paraphrased sentences. As such, it needs to be used in conjunction with other metrics or evaluation procedures when measuring the performance of an SS 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-penalize 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, Abend, and Rappoport (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 judgments 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 transformations.

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, Abend, and Rappoport 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 because the semantic parsing is only performed in the original sentence, it prevents adding parser errors of (possibly) grammatically incorrect simplified sentences produced by the SS model being evaluated.

3.2.4 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 for MT,[8] Štajner, Mitkov, and Saggion (2014) proposed training 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. 2016), but automatic judgments 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, Mitkov, and Saggion (2014)'s work with features from Štajner, Popović, and Béchera (2016) to analyze how different feature groups correlate with human judgments on grammaticality, meaning preservation, and simplicity using data from QATS. Using Quality Estimation 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 is interested in **ranking** sentences according to their predicted reading levels. Vajjala and Meurers, (2014a,b) showed that, in the PWKP (Zhu, Bernhard, and Gurevych 2010) data set 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

[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, Scarton, and Paetzold (2018).

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 using 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, Reddy, and Steedman (2016) tested the usefulness of syntactic features extracted from an incremental parser for the task, and Howcroft and Demberg (2017) explored using more psycholinguistic features, such as idea density, surprisal, integration cost, and embedding depth.

Although not detailed in this section, some research has used METEOR (Denkowski and Lavie 2014) from the MT literature, and ROUGE (Lin 2004), borrowed from summarization research.

3.3 Discussion

In this section we have described how the outputs of SS models are evaluated using both human judgments and automatic metrics. We have attempted to not only explain these methods, but also to 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 SS model was trained. This is especially relevant when collecting simplicity judgments because of 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 judgments may arise. This is why some researchers have started to focus on developing and evaluating models for personalized simplification (Bingel, Paetzold, and Søgaard 2018; Bingel, Barrett, and Klerke 2018). 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 judgments 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 of the output. An example of such type of assessment is presented in Mandya, Nomoto, and Siddharthan (2014), where human judges had to use the automatically simplified texts in a reading comprehension test with multiple-choice questions. Then, 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 SS research are flawed. BLEU has been found to only be reliable for assessment in MT but not other Natural Language Generation tasks (Reiter 2018), and it is not adequate for most rewriting transformations in SS (Sulem, Abend, and Rappoport 2018a). SARI is only useful as a proxy for simplicity gain assessment, limited to lexical simplifications and short-distance reordering despite more text transformations 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 these automatic scores, it is essential to keep all their particular limitations in mind, to always look at all possible metrics and try to interpret them accordingly. Overall, what is the most reliable way of automatically evaluating system outputs, at the right granularity and considering all characteristics of the task, is still an open question. We comment on some possible future directions in Section 6.

4. Data-Driven Approaches to Sentence Simplification

In this section, we review research on SS aiming at learning simplifications from examples. More specifically, approaches that involve learning text transformations from parallel corpora of aligned original-simplified sentences in English. Compared with approaches based on hand-crafted rules, data-driven approaches can perform multiple simplification transformations simultaneously, as well as learn very specific and complex rewriting patterns. As a result, they make it possible to model interdependencies among different transformations 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 (Candido Jr. et al. 2009; Siddharthan 2011; Bott, Saggion, and Mille 2012), nor approaches that only learn lexical simplifications, that is, which target one-word replacements (see Paetzold and Specia [2017b] for a survey).

We classify data-driven approaches for SS as relying on statistical MT techniques (Section 4.1), induction of synchronous grammars (Section 4.2), semantics-assisted (Section 4.3), and neural sequence-to-sequence models (Section 4.4).

4.1 Monolingual Statistical Machine Translation

Several approaches treat SS as a monolingual MT task, with *original* and *simplified* as *source* and *target* languages, respectively. Whereas 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 modeled using the noisy channel framework Equation (9).

$$e^* = \arg\max_{e \in E} p(e|f) = \arg\max_{e \in E} p(f|e)p(e) \tag{9}$$

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 optimized to maximize translation quality, rather than on the generative noisy channel model. In what follows, we review the most popular approaches and explain their applications for SS.

4.1.1 Phrase-Based 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 normalized count of the number of times each possible phrase-pair occurs. These counts are extracted from parallel corpora and automatic **phrase alignments**, which, in turn, are obtained from word alignments. Decoding is a search problem: Find the sentence that maximizes 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,

Table 2
Performance of PBSMT-based sentence simplification models as reported by their authors.

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

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 with not doing any simplification. In Coster and Kauchak (2011a), the authors modified Moses to allow complex phrases to be aligned with 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 with NULL, and (2) if several complex words in a set align with 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 with NULL. Their model achieves better results than a standard Moses implementation.

PBSMT-R: Wubben, van den Bosch, and Krahmer (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 score. This trained model achieves a better BLEU score than more sophisticated approaches (Zhu, Bernhard, and Gurevych 2010; Woodsend and Lapata 2011a), explained in Section 4.2. When compared with such approaches using human evaluation, PBSMT-R is better in grammaticality and meaning preservation. However, the results are limited to paraphrasing transformations.

Table 2 summarizes the performance of the models trained using the SS approaches described, as reported in the original papers. The BLEU values are not directly comparable, since each approach used a different corpus for testing. From a transformation 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.

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

4.1.2 Syntax-Based Approaches. In Syntax-Based SMT (SBSMT), the basic unit for translation is no longer a phrase, but syntactic components in parse trees. In PBSMT, the language model and the 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, Bernhard, and Gurevych (2010) proposed a **T**ree-based **S**implification **M**odel that can perform four text transformations: splitting, dropping, reordering, and substitution (of words and phrases). Given an original sentence c, the model attempts to find a simplification s using Equation (10), with a language model $P(s)$ and a direct translation model $P(s|c)$.

$$s = \arg\max_{s} P(s|c)P(s) \tag{10}$$

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 transformation-specific and are stored in feature tables for each transformation. For each feature combination in each table, probabilities are calculated during training. We will use the splitting transformation to explain this process in more detail. A similar method is used for the other three transformations.

In TSM, sentence splitting is decomposed into two transformations: SEGMENTATION (if a sentence is to be split or not) and COMPLETION (to make the splits grammatical). The probability of a SEGMENTATION transformation is calculated using Equation (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).

$$P(seg|c) = \prod_{w:c} SFT(w|c) \tag{11}$$

COMPLETION implies deciding whether 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 (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 that is out of the scope of s, $BDFT$ is the Border Drop Feature Table (BDFT), and CFT is the Copy Feature Table.

$$P(com|seg) = \prod_{bw:s} BDFT(bw|s) \prod_{w:s} \prod_{dep:w} CFT(dep) \tag{12}$$

Finally, once similar computations are done for the other three transformations, all probabilities are combined to calculate the translation model. In Equation (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.

$$\begin{aligned} P(s|c) = \sum_{\theta:Str(\theta(c))=s} \Big(& P(seg|c)P(com|seg) \\ & \prod_{node} P(dp|node)P(ro|node)P(sub|node) \\ & \prod_{w} (sub|w) \Big) \end{aligned} \tag{13}$$

The model is trained using the Expectation-Maximization algorithm proposed by Yamada and Knight (2001). This algorithm requires constructing 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 focused on 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 (14).

$$\hat{S}(c) = \arg\max_{\forall S} P(S|c) \tag{14}$$

The language and translation models are combined using a log-linear model as in Equation (15), where $f_m(S, c)$ are feature functions on each sentence, and w_m are model parameters to be learned.

$$p(S|c) = \frac{\exp\left(\sum_{m=1}^{M} w_m f_m(S, c)\right)}{\sum_{S'} \exp\left(\sum_{m=1}^{M} w_m f_m(S', c)\right)} \tag{15}$$

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 modeling, 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 optimizing a SBSMT framework with rule-based features and tuning metrics specifically tailored for lexical simplification. Two new "light weight" metrics are also introduced: FKBLEU and SARI, both described in Section 3.2.

Table 3
Performance of SBSMT-based sentence simplification models as reported by their authors.

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

The proposed simplification model also relies on paraphrasing rules available in the 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 Section 2.3) on three versions of the SBSMT system, changing the tuning metric (BLEU, FKBLEU, and SARI). Evaluations using human judgments show that all three models achieved better grammatically, meaning preservation, and simplicity gain than PBSMT-R (Wubben, van den Bosch, and Krahmer 2012).

Table 3 summarizes the performance of the syntax-based models trained using the SS approaches described. These values are not directly comparable, because 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 for which it was optimized. However, SBSMT (PPDB + SARI) seems to be the overall best. From a transformations capability point of view, TSM and TriS are capable of performing splitting, which is an advantage over SBSMT variations that only generate paraphrases.

4.2 Grammar Induction

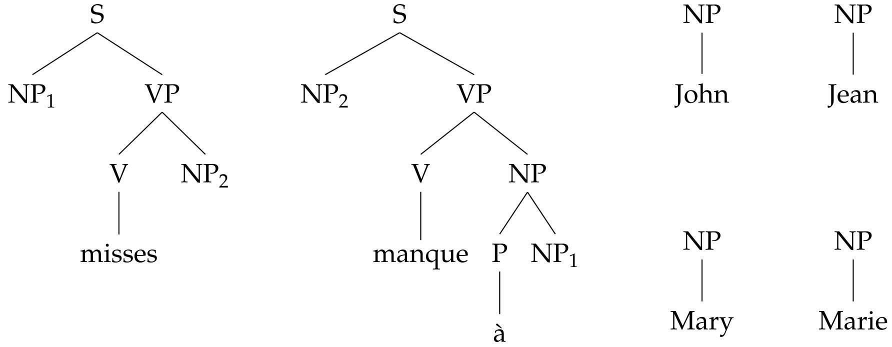
In this approach, SS is modeled as a tree-to-tree rewriting problem. Approaches typically follow a two-step process: (1) use parallel corpora of aligned original-simplified sentence pairs to extract a set of tree transformation rules, and then (2) learn how to select which rule(s) to apply to unseen 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 learned 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.

4.2.1 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 generalization of CFGs able to generate pairs of related strings and not just single strings (Chiang 2006). In a SCFG, each production has two hand-sides (source and target) that are related. For example, we show a SCFG for a sentence in English and its translation to Japanese (Chiang 2006):

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

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. This requirement is relaxed by the STSGs. 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** (QG, 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 because *anything can align to anything*.

4.2.2 Simplification Models. The formalisms explained in the previous subsection 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 (favors 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 the 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 judgments 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 uses 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). These constituent alignments are then generalized (i.e., aligned nodes are replaced with links) to extract rules. This generalization 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 the 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.

Table 4
Performance of grammar-based sentence simplification models as reported by their authors. 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	

Evaluation results using human judgments on simplicity, grammaticality, and meaning preservation showed that candidate simplifications were only encouraging for lexical simplification, almost half of the automatically simplified sentences were considered simpler, over 80% were considered grammatical, and a little over 50% were identified 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 generalized. Also, instead of controlling the rules' specificity with the recursion depth, SimpleTT augments the rules with more information, such as head's lexicalization 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 obtains the highest scores for simplicity and grammaticality among the tested models, with values comparable to those for human simplifications. However, it obtains the lowest score in meaning preservation, presumably because SimpleTT tends to simplify by deleting a sentence's elements. This approach does not explicitly model sentence splitting.

Table 4 summarizes the performance of the models trained with the SS approaches described. Unfortunately, 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 with end-to-end approaches. Even though Woodsend and Lapata (2011a) were the only ones who attempted model splitting, the other approaches could be modified in a similar way, since the formalisms allow it.

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 (1), *bricks* is involved in two events: "being resistant to cold" and "enabling the construction of permanent buildings."

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

(2) **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, because we need to identify the entity involved in both events.

Hybrid: Narayan and Gardent (2014) built an SS model by combining semantics-driven splitting and deletion, with PBSMT-based substitution and reordering. The general idea is to use semantic roles information to identify the events in a given original sentence, and those events would determine how to split the sentence. Also, deletion would 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, Clark, and Bos 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. Additionally, the probabilities for substitution and reordering are determined by a PBSMT system. For training, they used the Expectation-Maximization algorithm of Yamada and Knight (2001), in a similar fashion to Zhu, Bernhard, and Gurevych (2010), but calculating probabilities over the semantic graph produced by Boxer instead of a syntactic parse tree.

The SS 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 TS model. Their approach first uses a context-aware lexical simplifier (Biran, Brody, and Elhadad 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 score. It produces more splits than Hybrid, but also more than the reference. There is no analysis about the correctness of the splits produced. Using human evaluation, UNSUP achieves the highest values in simplicity and grammaticality.

EvLex: Štajner and Glavaš (2017) introduce an SS model that can perform sentence splitting, content reduction, and lexical simplifications. For the first two transformations, 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 tajner 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 with QG+ILP (Woodsend and Lapata 2011a), testing in two data sets, one with news stories (News) and one with Wikipedia sentences (Wiki). EvLex achieves the highest FRE score in the News data set, 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 data sets.

Table 5 summarizes the results of the models trained with the SS 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 Section 4.1.2, 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 modeling, different from standard MT-based sequence-to-sequence approaches.

4.3.1 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 data set (described in Section 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, Pham, and Manning 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

Table 5
Performance of semantics-assisted sentence simplification models as reported by their authors.

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

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 data set 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 summarization (2016; See, Liu, and Manning 2017). This model achieves the best performance in both the original and new data set 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, because WEBSPLIT was derived from the WEBNLG data set, it only contains artificial sentence splits created from the RDF triples. As such, they introduce WikiSplit, a new data set for split-and-rephrase based on English Wikipedia edit histories (see Sec. 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.

4.4 Neural Sequence-to-Sequence

In this approach, SS is modeled as a sequence-to-sequence problem, and tackled normally with an attention-based encoder-decoder architecture (Bahdanau, Cho, and Bengio 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 training of end-to-end models without needing to extract features or estimate individual model components, such as the language model. In addition, all simplification transformations can be learned simultaneously, instead of developing individual mechanisms as in previous research.

4.4.1 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 $X = (x_1, x_2, \ldots, x_{|X|})$, the model learns to predict its simplified version, $Y = (y_1, y_2, \ldots, y_{|Y|})$. It uses an encoder that transforms the source sentence X into a sequence of hidden states $(h^S_1, h^S_2, \ldots, h^S_{|X|})$, 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:

$$P(Y|X) = \prod_{t=1}^{|Y|} P(y_t|y_{1:t-1}, X) \tag{16}$$

$$P(y_{t+1}|y_{1:t}, X) = \text{softmax}(g(h^T_t, c_t)) \tag{17}$$

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

$$g(h_t^T, c_t) = W_o \tanh(U_h h_t^T + W_h c_t) \tag{18}$$

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 that summarizes $y_{1:t}$ (what has been generated so far):

$$h_t^T = \text{LSTM}(y_t, h_{t-1}^T) \tag{19}$$

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:

$$c_t = \sum_{i=1}^{|X|} \alpha_{t_i} h_i^S \qquad\qquad \alpha_{t_i} = \frac{\exp(h_t^T \cdot h_i^S)}{\sum_i \exp(h_t^T \cdot h_i^S)} \tag{20}$$

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 against 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 paraphrasing and deletion transformations. It is also apparent that choosing the second hypothesis results in less conservative simplifications.

NematuSS: Alva-Manchego et al. (2017) also tested this standard neural architecture for SS, but used the implementation provided by Nematus (Sennrich et al. 2017). They experimented with different types of original-simplified sentence alignments extracted from the Newsela corpus. When experimenting with all possible sentence alignments, the model tended to be too aggresive, mostly performing deletions. When using only 1-to-1 alignments, the model became more conservative, and the simplifications performed were restricted to deletions and one-word replacements.

targeTS: Inspired by the work of Johnson et al. (2017) on multilingual neural MT, Scarton and Specia (2018) enriched the encoder's input with information about the target audience and the (predicted) simplification transformations to be performed. Concretely, an artificial token was added to the beginning of the input sentences indicating (1) the grade level of the simplification instance, and/or (2) one of four possible text transformations: identical, elaboration, splitting, or joining. At test time, the text transformation 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 scores when using this extra information.

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 data sets, 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.

4.4.2 Modifying the Training Method. Without significantly changing the standard RNN-based architecture described before, some research has experimented with alternative learning algorithms with which the models are trained.

Dress: Zhang and Lapata (2017) use the standard attention-based encoder-decoder as an agent within a Reinforcement Learning (RL) architecture (Figure 2). An advantage of this approach is that the model can be trained end-to-end using SS-specific metrics.

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 maximizes the expected reward. As such, the training loss is given by the negative expected reward:

$$\mathcal{L}(\theta) = -\mathbb{E}_{(\hat{y}_1,\ldots,\hat{y}_{|\hat{Y}|}) \sim P_{RL}(\cdot|X)}[r(\hat{y}_1,\ldots,\hat{y}_{|\hat{Y}|})] \qquad (21)$$

where P_{RL} is the policy, given in our case by the distribution produced by the encoder-decoder Equation (17) and $r(\cdot)$ is the reward function. The authors followed Ranzato et al. (2016) in first pre-training the agent by minimizing the negative log-likelihood of the training source–target pairs, in order to avoid starting the process with

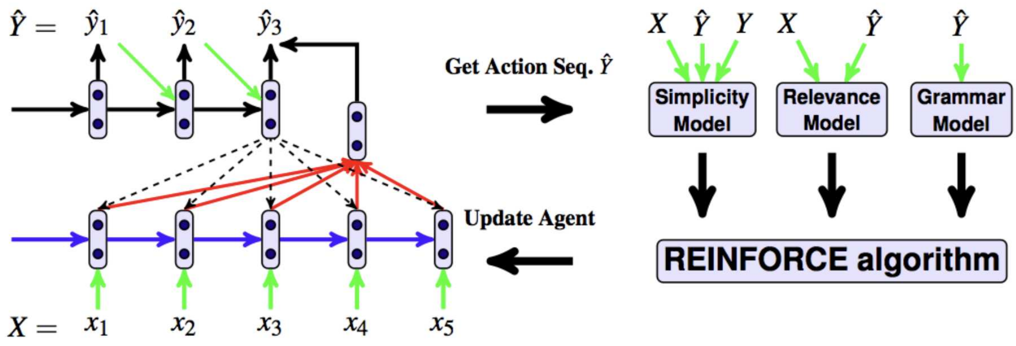


Figure 2
Model architecture for Dress. Extracted from Zhang and Lapata (2017).

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 obtain probabilistic word alignments (attention scores α_t) that help determine whether 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 (22), where P_{LS} is the probability of simplifying a word.

$$P(y_t|y_{1:t-1}, X) = (1 - \eta)P_{RL}(y_t|y_{1:t-1}, X) + \eta P_{LS}(y_t|X, \alpha_t) \tag{22}$$

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 on three different data sets: 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, Pasunuru, and Bansal (2018) worked with SS within a Multi-Task Learning (MTL) framework. Considering SS as the main task, they incorporated two auxiliary tasks to improve the 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 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., SS and paraphrasing); whereas higher-level layers are shared among tasks focused on semantics and meaning (i.e., SS and entailment). In addition, their RNN-based model is enhanced with a pointer-copy mechanism (See, Liu, and Manning 2017), which allows deciding at decoding time whether to copy a token from the input or generate one.

When training main and auxiliary tasks in parallel, a concern within MTL is how to determine the appropriate number of iterations on each task relative to the others. This is normally handled using a static hyperparameter. In contrast, Guo, Pasunuru, and Bansal (2018) proposed learning 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 SS; the SNLI (Bowman et al. 2015) and the MultiNLI (Williams, Nangia, and Bowman 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 judgments, their model scored as the best in simplicity.

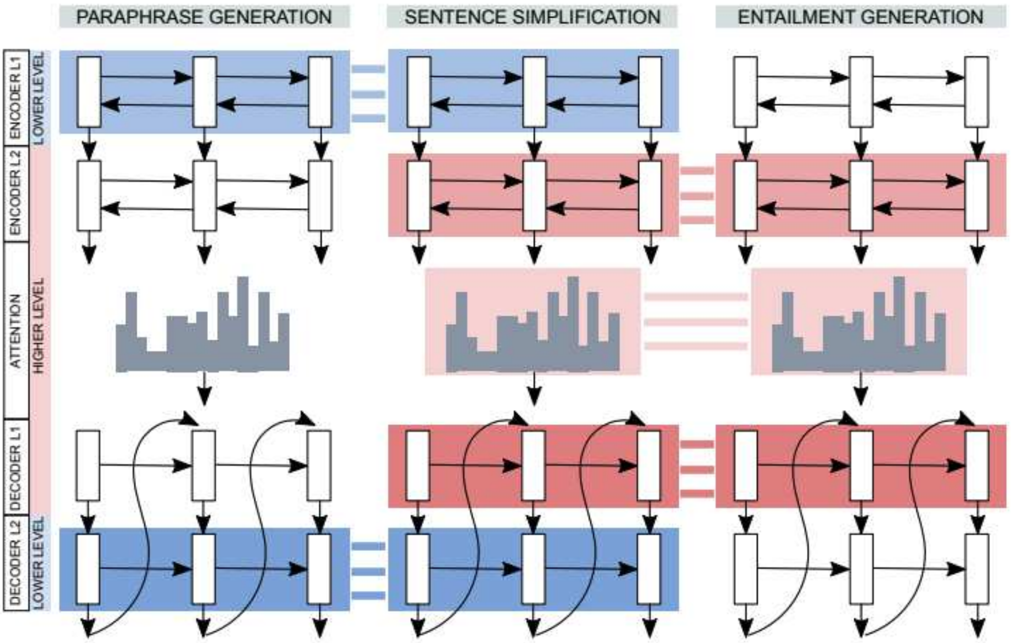


Figure 3
Model architecture for PointerCopy+MTL. Extracted from Guo, Pasunuru, and Bansal (2018).

4.4.3 Adding External Knowledge. The previously described models attempted to learn how to simplify only using information from the training data sets. Zhao et al. (2018) argued that the relatively small size of these data sets prevents models from generalizing well, considering the vast amount of possible simplification transformations 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 SPPDB were incorporated into the model using two mechanisms. In Deep Critic Sentence Simplification (DCSS), the model uses a new loss function that maximizes the probability of generating the simplified form of a word, while minimizing 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 SPPDB 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 utilization based on ground-truth from SPPDB, and showed that their models also improved compared to previous work.

4.4.4 Unsupervised Architectures. Surya et al. 2019 proposed an unsupervised approach for developing a simplification system. Their motivation was to design an architecture that could be exploited to train SS 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 data set. It interacts with G_s using an adversarial loss function $\mathcal{L}_{adv}$, in a similar fashion as GANs (Goodfellow et al. 2014). The Classifier is in charge of diversification by ensuring, through loss function $\mathcal{L}_{div}$, that both G_d and G_s attend differently to the hidden representations generated by the shared encoder. Two additional loss functions, $\mathcal{L}_{rec}$ and $\mathcal{L}_{denoi}$, are used for reconstructing sentences and denoising, respectively. The full architecture can be seen in Figure 4.

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 WebSplit (Narayan et al. 2017) data sets to train a model (UNTS+10K) with minimal supervision. Their models were compared against unsupervised systems from the MT literature (Artetxe, Labaka, and Agirre 2018; Artetxe et al. 2018), as well as SS 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 behavior 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 text transformation.

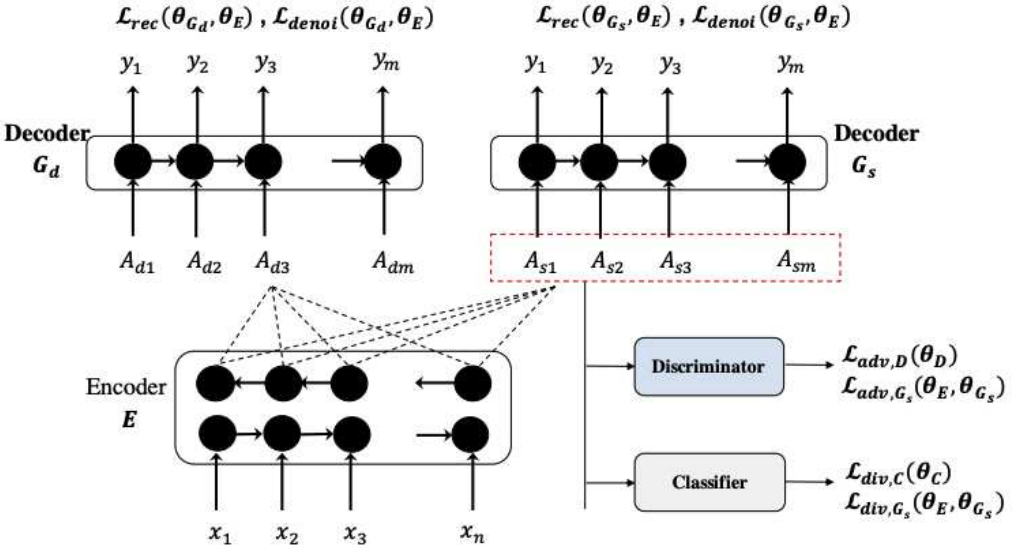

$\mathcal{L}_{rec}(\theta_{G_d}, \theta_E)$, $\mathcal{L}_{denoi}(\theta_{G_d}, \theta_E)$
$\mathcal{L}_{rec}(\theta_{G_s}, \theta_E)$, $\mathcal{L}_{denoi}(\theta_{G_s}, \theta_E)$
y_1 y_2 y_3 y_m
y_1 y_2 y_3 y_m
Decoder G_d
Decoder G_s
A_{d1} A_{d2} A_{d3} A_{dm}
A_{s1} A_{s2} A_{s3} A_{sm}
Encoder 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 4
Model architecture for UNTS. Extracted from Surya et al. (2019).

4.4.5 Simplification as Sequence Labeling. Alva-Manchego et al. (2017) model SS as a **S**equence **L**abeling problem, identifying simplification transformations at word or phrase level. They use the token-level annotation algorithms of MASSAlign (Paetzold, Alva-Manchego, and Specia 2017) to automatically generate annotated data from which an LSTM learns to predict simplification transformations; more specifically, deletions and replacements. During decoding, words labeled to be deleted are just not included in the output. To produce replacements, they use the lexical simplifier of Paetzold and Specia (2017a). The proposed approach is compared against MT-based models: Moses (Koehn et al. 2007), Nematus (Sennrich et al. 2017), and NTS+word2vec (with default settings) using data from the Newsela corpus. Alva-Manchego et al. (2017) achieve the highest SARI score in the test set, and best simplicity score with human judgments. This approach is inspired in the abstractive sentence compression model of Bingel and Søgaard (2016), who propose a tree labeling approach to remove or paraphrase syntactic units in the dependency tree of a given sentence, using a Conditional Random Fields predictor.

Most sequence-to-sequence approaches for training SS models could be considered as black boxes with respect to which simplification transformations should be applied to a given sentence. That is a desirable feature for a holistic approach to SS, 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 transformation independently (potentially using a different approach for each operation). Approaches based on labeling could be helpful in such cases. A disadvantage, however, is collecting quality annotated data from which to learn. In addition, some simplification transformations are hard to predict (e.g., insertion of words that do not come from the original sentence).

Table 6 summarizes the performance of the models trained with the SS approaches described. 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, DMASS-DCSS obtains the highest SARI score, with NseLstm-B achieving the best BLEU. Both approaches used WikiLarge for training. Regarding the transformations 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 architectures are able to perform sentence splitting.

4.5 Discussion

In this section we have presented a summary of research in data-driven automatic SS. This review has helped to understand the benefits and shortcomings of each approach to the task. Traditionally, SS has been reduced to four text transformations: *substituting* complex words or phrases, *deleting* or *reordering* sentence components, and *splitting* a complex sentence into several simpler ones (Zhu, Bernhard, and Gurevych 2010; Narayan and Gardent 2014). Table 7 lists the surveyed SS models (grouped by approach), the techniques each of them explores, and the simplification transformations that they can perform, considering the four traditional rewriting operations established.

Overall, SMT-based methods can perform substitutions, short-distance reorderings, and deletions, but fail to produce quality splits unless explicitly modeled using syntactic

Table 6
Performance of sequence-to-sequence sentence simplification models as reported by their authors.

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
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
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
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
DMASS+DCSS	WikiLarge	TurkCorpus		8.04	40.45
	WikiLarge	Newsela		5.17	27.28

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, sequence labeling seems more straightforward, and offers some flexibility in how each identified transformation could be handled individually. However, there is no available manually produced corpus with the required annotations that could allow studying the advantages and limitations of this approach. Additionally, there is little research on dealing with discourse-related issues caused by the rewriting transformations, or on considering more document-level constraints.

5. Benchmarking Sentence Simplification Models

So far in this article we have explained how several SS models were implemented, and have commented on their performance based on the results reported by their authors. However, one problem we encountered was comparing these models against one another objectively, since most authors used different corpora or (implementations of) metrics for testing their models. Some researchers have realized this problem and it is now more common to have access to their models' outputs on some data sets. Also, some have reimplemented SS approaches that were not made available by their original authors. In this section, we take advantage of this fact to use publicly available outputs of SS models on commonly used data sets, and measure their simplification performance using the same set of metrics and test data.

As shown in Section 3, in order to automatically evaluate the output of an SS model, we normally use MT-inspired metrics (e.g., BLEU), readability metrics (e.g., Flesch Kincaid), and simplicity metrics (e.g., SARI). Automatic metrics are easy to compute,

Table 7
Summary of sentence-level text simplification approaches: SMT-based (first section), grammar-based (second section), semantics-assisted (third section), and neural sequence-to-sequence (fourth section). The transformations listed are the ones the authors acknowledge that their models can perform. Transformations with * are found in some of the outputs, but not explicitly modeled by the authors.

Model	Approach	Transformations
Specia (2010)	PBSMT (Moses)	SUB, REORD
Coster and Kauchak (2011b)	PBSMT (Moses)	SUB, REORD
Coster and Kauchak (2011a)	PBSMT (Moses) + Deletion	SUB, DEL, REORD
Wubben, van den Bosch, and Krahmer (2012)	PBSMT (Moses) + Dissimilarity Ranking	SUB
Zhu, Bernhard, and Gurevych (2010)	SBSMT	SUB, DEL, REORD, SPLIT
Bach et al. (2011)	SBSMT	SPLIT
Xu et al. (2016)	SBSMT (Joshua) + SPPDB + SARI Optimization	SUB, REORD
Woodsend and Lapata (2011a)	QG + ILP	SUB, DEL, REORD, SPLIT
Paetzold and Specia (2013)	STSGs + Perplexity Ranking	SUB, DEL, REORD, SPLIT*
Feblowitz and Kauchak (2013)	STSGs Backoff + Log-linear Reranking	SUB, DEL, REORD, SPLIT*
Narayan and Gardent (2014)	Deep Semantics (Boxer) + PBSMT	SUB, DEL, REORD, SPLIT
Narayan and Gardent (2016)	Lexical Simp. + Deep Semantics (Boxer) + ILP	SUB, DEL, SPLIT
Štajner and Glavaš (2017)	Event Detection for Splitting + Unsupervised Lexical Simplification	SUB, DEL, SPLIT
Narayan et al. (2017)	Semantics-aided Splitting + NTM	SUB, REORD, SPLIT
Nisioi et al. (2017)	Seq2Seq (RNN) + PPDB + SARI	SUB, DEL, REORD
Zhang and Lapata (2017)	Seq2Seq (RNN) + RL	SUB, DEL, REORD
Vu et al. (2018)	Seq2Seq (RNN) with NSE	SUB, DEL, REORD
Guo, Pasunuru, and Bansal (2018)	Seq2Seq (RNN) + MTL	SUB, DEL, REORD
Zhao et al. (2018)	Seq2Seq (Transformer) + SPPDB	SUB, DEL, REORD
Alva-Manchego et al. (2017)	Sequence Labeling	SUB, DEL

but they only provide overall performance scores that cannot explain specific strengths and weaknesses of an SS approach. Therefore, we propose to also evaluate SS models based on how effective they are at executing specific simplification transformations. We show how this per-transformation assessment contributes to a better understanding of the automatic scores, and to an improved comparison between different SS models.

5.1 Evaluation Setting

In this section, we describe the test sets for which we collected SS models' outputs. After that, we specify how we compare them using a few of the automatic metrics previously described and automatic identifications of a set of simplification transformations.

Table 8
Characteristics of the tests sets on which SS models' outputs were collected. An instance corresponds to a source sentence with one or more possible references. Each reference can be composed of one or more sentences.

Test data set	Instances	Alignment type	References
PWKP	93	1-to-1	1
	7	1-to-N	1
TurkCorpus	359	1-to-1	8

5.1.1 Data sets. We compare the SS models' outputs in two commonly used test sets extracted from corpora based on EW and SEW: PWKP (Zhu, Bernhard, and Gurevych 2010) and TurkCorpus (Xu et al. 2016). Both test sets contain automatically generated original-simplified sentence alignments (see Sections 2.1.1 and 2.3 for details). Table 8 lists some of their characteristics.

The 1-to-N alignments in PWKP mean that some instances of sentence splitting are present in the data set. This is a limitation of TurkCorpus that only contains 1-to-1 alignments, mostly providing instances of 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, like SARI. We do not use the Newsela corpus in our benchmark because researchers are prohibited from publicly releasing models' outputs on these data.

5.1.2 Overall Performance Comparison. We first compare the models' outputs using automatic metrics so as to obtain an overall measure of simplification quality. We calculate BLEU, SARI, SAMSA, and FKGL. We compute the scores for these metrics using EASSE (Alva-Manchego et al. 2019),[10] a Python package for single access to (re)implementations of these metrics. Specifically, it uses SacreBleu (Post 2018)[11] to calculate BLEU, a re-implementation of SARI's corpus-level version in Python (it was originally available in Java), a slightly modified version of the original SAMSA implementation[12] for improved execution speed, and a re-implementation of FKGL based on publicly available scripts[13] that fixes some edge case inconsistencies.

5.1.3 Transformation-Based Performance Comparison. We are also interested in an in-depth study of the simplification capabilities of each model. In particular, we want to determine which simplification transformations each model performs more effectively.

In order to identify the simplification transformations that a model performed, it would be ideal to have corpora with such type of annotations and compare against them. However, there is no available simplification corpus with such a type of information. As a work-around, we use the `annotator` module of MASSAlign (Paetzold, Alva-Manchego, and Specia 2017). This tool provides algorithms to automatically label the simplification transformations carried out in aligned original-simplified sentences.

[10] `https://github.com/feralvam/easse`.
[11] `https://github.com/mjpost/sacreBLEU`.
[12] `https://github.com/eliorsulem/SAMSA`.
[13] `https://github.com/mmautner/readability`.

Based on word alignments, the algorithms attempt to identify copies, deletions, movements, and replacements.

Alva-Manchego et al. (2017) tested the quality of the automatically produced labels, comparing them with manual annotations for 100 sentences from the Newsela corpus. These annotations were performed by four proficient speakers of English. For 30 of those sentences annotated by the four annotators, the pairwise inter-annotator agreement between annotators yielded an average kappa value of 0.57. For all labels (excluding copies), the algorithms achieved a micro-averaged F1 score of 0.61, being especially effective at identifying deletions and replacements.

MASSAlign's annotation algorithms were integrated into EASSE, and are used to generate two sets of automatic word-level annotations: (1) between the original sentences and their reference simplifications, and (2) between the original sentences and their automatic simplifications produced by an SS system. Considering (1) as reference labels, we calculate the F1 score of each transformation in (2) to estimate their correctness. When more than one reference simplification exists, we calculate the per-transformation F1 scores of the output against each reference, and then keep the highest one as the sentence-level score. The corpus-level scores are the average of sentence-level scores.

5.2 Comparing Models in the PWKP Test Set

For the PWKP test set, the models that have publicly available outputs are: Moses (released by Zhu, Bernhard, and Gurevych 2010), PBSMT-R, QG+ILP (released by Narayan and Gardent 2014), Hybrid, TSM, UNSUP, EncDecA, Dress, and Dress-Ls. Overall scores using standard metrics are shown in Table 9 sorted by SARI.

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 Dress

Table 9
Performance measured with automatic metrics in the PWKP test set.

Model	SARI ↑	BLEU ↑	SAMSA ↑	FKGL ↓
Reference	100.00	100.00	29.91	8.07
Hybrid	**54.67**	53.94	36.04	10.29
Moses	48.99	**55.83**	34.53	11.58
Dress-Ls	40.44	36.32	29.43	8.52
Dress	40.04	34.53	28.92	8.40
TSM	39.02	37.69	37.39	**6.40**
UNSUP	38.41	38.28	35.81	7.75
PBSMT-R	35.49	46.31	35.63	12.26
QG+ILP	35.24	41.76	**41.71**	7.08
EncDecA	32.26	47.93	35.28	12.12

Table 10
Performance measured with transformation-specific F1 score in the PWKP test set.

Model	Delete	Move	Replace	Copy
Moses	31.36	2.47	**17.17**	**70.27**
Hybrid	34.01	2.84	16.06	69.83
PBSMT-R	12.05	0.00	8.17	66.59
TSM	29.15	0.62	6.57	62.22
EncDecA	8.59	0.34	5.23	66.09
UNSUP	25.58	1.40	4.27	62.83
Dress	**35.65**	0.38	3.40	59.02
Dress-Ls	35.23	0.36	2.33	59.97
QG+ILP	16.77	**3.90**	2.00	56.47

obtains the lowest BLEU score, its FKGL value is not reliable. In terms of SAMSA, the best model is QG+ILP, followed by TSM and Hybrid. Because this metric focuses on assessing sentence splitting, it is expected that the models that explicitly perform this transformation scored best. From these results, we could conclude that Hybrid is the overall best model, since it achieves the highest SARI, BLEU, and SAMSA scores close to the highest, and a FKGL 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. Because 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:

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

(4) **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 10 presents the results of our transformation-based performance measures for models' outputs on PWKP. From Table 9, we saw that Hybrid got the highest SARI score. With the results on Table 10 we can understand better why that happened. Because 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 replacements and copying operations are among the most accurate. EncDecA, which obtained the worst SARI score, is only average in replacement and copy, and has the lowest deletion accuracy, which points out to them mostly repeating the original sentence without much modifications. Finally, Dress and Dress-Ls are the best ones in deleting content, which explains their low (and "good") FKGL scores.

5.3 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) and NTS+SARI. Because the TurkCorpus has multiple simplified references for each original sentence,

Table 11
Performance measured with automatic metrics in the TurkCorpus test set.

	TurkCorpus		HSplit	
Model	SARI ↑	BLEU ↑	SAMSA ↑	FKGL ↓
Reference	49.88	97.41	54.00	8.76
DMASS-DCSS	**40.42**	73.29	35.45	7.66
SBSMT(PPDB+SARI)	39.96	73.08	41.41	7.89
PBSMT-R	38.56	**81.11**	**47.59**	8.78
NTS+SARI	37.30	79.82	45.52	8.11
Dress-Ls	37.27	80.12	45.94	7.58
Dress	37.08	77.18	44.47	7.45
Hybrid	31.40	48.97	46.68	**5.12**

Table 12
Performance measured with transformation-specific F1 score in the TurkCorpus test set.

Model	Delete	Move	Replace	Copy
PBSMT-R	34.18	2.64	23.65	**93.50**
Hybrid	**49.46**	**7.37**	1.03	70.73
SBSMT-SARI	28.42	1.26	**37.21**	92.89
NTS-SARI	31.10	1.58	23.88	88.19
Dress-Ls	40.31	1.43	12.62	86.76
DMASS-DCSS	38.03	5.10	34.79	86.70

we use all of them for measuring BLEU and SARI. For calculating reference values, we sample one of the eight human references for each instance as others have done (Zhang and Lapata 2017). When reporting SAMSA scores, we only use the 70 sentences of TurkCorpus that also appear in HSplit.[14] This allows us to compute reference scores for instances that contain structural simplifications (i.e., sentence splits). We calculate SAMSA for each of the four manual simplifications in HSplit, and choose the highest as an upper-bound. Results are presented in Table 11 sorted by SARI score.

In this test set, the models that used external knowledge from Simple PPDB achieved the highest scores in terms of SARI: DMASS-DCSS and SBSMT (PPDB+SARI). The most fluent model according to BLEU is PBSMT-R, closely followed by Dress-Ls. This could be due to MT-based models being capable of generating grammatical output of high quality. Contrary to what was observed in the PWKP test set, Hybrid achieved the worst scores in SARI and SAMSA. 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 behavior is penalized since most references were only rewritten considering paraphrases of words and phrases.

Table 12 presents the results of our transformation-based performance measures for models' outputs on the TurkCorpus test set. Similarly to the PWKP test set, the effectiveness results on TurkCorpus support the scores of the automatic metrics (Table 11).

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

SBSMT (PPDB+SARI) is the best at performing replacements, which explains its high SARI score. PBSMT-R obtained the best BLEU score, which is explained by the above-average copy transformations. Even though Hybrid achieved the lowest FKGL value, it is the best in deletions, has a below-average copying, and does not produce accurate 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 paraphrases, that is, mostly 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 data set.

6. Conclusions and Future Directions

In this article we presented a survey of research in Text Simplification, focusing on models that perform the task at sentence-level. We limited our review to models that learn to produce simplifications by using parallel corpora of original-simplified sentences. We reviewed the main resources exploited by these approaches, detailed how models are trained using these data sets, and explained how they are evaluated. Finally, we compared several SS models using standard overall-performance metrics, and proposed a new operation-specific method for qualifying the simplification transformations that each SS model performs. The former analysis provided us with insights about the simplification capabilities of each approach, which help better explain the initial automatic scores. Based on our review, we suggest the following as areas that are worth exploring.

Corpora Diversity. Most data sets used in data-driven SS research are based on EW and SEW. Its quality has been questioned (see Section 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 data sets 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 data sets be as diverse as possible in terms of application domains, target audiences, and text transformations realized. Finally, it would also be valuable to follow Xu et al. (2016) by collecting multiple simplification references per simplification, for a fairer evaluation.

Explanation Generation. Even though most current SS approaches do not focus on specific simplification transformations, it is safe to say that they tackle the four main ones: deletion, substitution, reordering, and splitting (see Table 7). However, by definition, simplifying a text could also involve further explaining complex terms or concepts. This is not merely replacing a word or phrase for a simpler synonym or its definition, but to elaborate on the concept in a natural way that keeps the text grammatical, is meaning preserving, and is simple. This is an important research area in SS where limited work has been published (Damay et al. 2006; Watanabe et al. 2009; Kandula, Curtis, and Zeng-Treitler 2010; Eom, Dickinson, and Sachs 2012).

Personalized Simplification. All SS approaches reviewed in this article are general purpose, that is, without a specific target audience. This is because current SS research focuses on how to learn the simplification operations that are realized in the corpus being used, and not on the end-user of the model's output. The simplification needs of a non-native speaker are different from those of a person with autism or with a low-literacy level. One possible solution could be to create target-audience-specific corpora to learn from. However, even within the same target group, individuals have specific simplification needs and preferences. We believe it would be meaningful to develop approaches that are capable of handling the specific needs of its user, and possibly learn from its interactions as a way to generate more helpful simplifications for that particular person.

Document-level Approaches. Most TS models focus on simplifying sentences individually, and there is little research on tackling the problem with a document-level perspective. Woodsend and Lapata (2011b) and Mandya, Nomoto, and Siddharthan (2014) produce simplifications at the sentence-level and try to globally optimize 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. It is likely that transformations that span multiple sentences are performed, which could never be tackled by a sentence-level model. In addition, proper corpora should be curated for training and testing of data-driven models, and 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 TS, even though it contains original-simplified aligned documents, with versions in several simplification levels.

Sentence Joining. Most current SS 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 Section 1.2, studies have shown that humans tend to join sentences together while simplifying a text, and perform a form of abstractive summarization. No state-of-the-art TS 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 TS systems.

Simplification Evaluation. For automatic evaluation of SS models, there are currently only two simplification-specific metrics: SARI (focused on paraphrasing) and SAMSA (focused on sentence splitting). However, as mentioned in Section 1.2, humans perform several more transformations that we are not currently measured when assessing a model's output. The transformation-specific evaluation presented in this article is merely a method for better understanding what the SS models are doing, but it is not a metric in its whole sense. We believe that more work needs to be done in improving how we evaluate and compare SS 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 (Štajner et al. 2016; Martin et al. 2018), 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, Nomoto, and Siddharthan 2014) be a more reliable quality measure? We believe these are questions that need to be addressed.

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 recent years. However, there are still several open questions that pose challenges to our research community. We hope that this article has helped provide an understanding of what has been done so far in the area and, more importantly, has motivated more people to advance the current state of the art.

References

Abend, Omri 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.

Aharoni, Roee 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.

Aluísio, Sandra M., 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.

Alva-Manchego, Fernando, 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.

Alva-Manchego, Fernando, Louis Martin, Carolina Scarton, and Lucia Specia. 2019. EASSE: Easier Automatic Sentence Simplification Evaluation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 49–54, Hong Kong.

Amancio, Marcelo 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.

Ambati, Bharat Ram, 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, CA.

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

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

Bach, Nguyen, 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.

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

Barzilay, Regina 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.

Belinkov, Yonatan, 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.

Bingel, Joachim, Maria Barrett, and Sigrid Klerke. 2018. 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, LA.

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

Bingel, Joachim 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.

Biran, Or, 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, OR.

Botha, Jan A., 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.

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

Bott, Stefan and Horacio Saggion. 2011b. An unsupervised alignment algorithm for text simplification corpus construction. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, MTTG '11, pages 20–26, Stroudsburg, PA.

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

Bowman, Samuel R., 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.

Brunato, Dominique, Felice Dell'Orletta, Giulia Venturi, and Simonetta Montemagni. 2015. Design and annotation of the first Italian corpus for text simplification. In *Proceedings of the 9th Linguistic Annotation Workshop*, LAW IX, pages 31–41, Denver, CO.

Candido Jr., Arnaldo, 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, CO.

Carroll, John, 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, WI.

Caseli, Helena M., 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.

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

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

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

Cohn, Trevor and Mirella Lapata. 2013. An abstractive approach to sentence compression. *ACM Transactions on Intelligent Systems and Technology*, 4(3):41:1–41:35.

Coster, William 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, OR.

Coster, William 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.

Crammer, Koby and Yoram Singer. 2003. Ultraconservative Online Algorithms for Multiclass Problems. *Journal of Machine Learning Research*, 3:951–991.

Crossley, Scott A., 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.

Curran, James R., 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.

Damay, Jerwin Jan S., Gerard Jaime D. Lojico, Kimberly Amanda L. Lu, Dex B. Tarantan, and Ethel C. Ong. 2006. SIMTEXT text simplification of medical literature. In *Proceedings of the 3rd National Natural Language Processing Symposium - Building Language Tools and Resources*, pages 34–38, Manila.

De Belder, Jan and Marie-Francine Moens. 2010. Text simplification for children. In *Proceedings of the SIGIR 2010 Workshop on Accessible Search Systems*, pages 19–26, Geneva.

Denkowski, Michael and Alon Lavie. 2014. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, pages 376–380, Baltimore, MD.

Devlin, Siobhan and John Tait. 1998. The use of a psycholinguistic database in the simplification of text for aphasic readers. *Linguistic Databases*, 161–173, Stanford, CA.

Eisner, Jason. 2003. Learning non-isomorphic tree mappings for machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics - Volume 2*, ACL '03, pages 205–208, Sapporo.

Eom, Soojeong, Markus Dickinson, and Rebecca Sachs. 2012. Sense-specific lexical information for reading assistance. In *Proceedings of the Seventh Workshop on Building Educational Applications Using NLP*, pages 316–325, Montréal.

Evans, Richard, 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.

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

Feblowitz, Dan 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.

Filippova, Katja 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.

Flesch, Rudolph. 1948. A new readability yardstick. *Journal of Applied Psychology*, 32(3):221.

Ganitkevitch, Juri, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of NAACL-HLT*, pages 758–764, Atlanta, GA.

Gardent, Claire, 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.

Glavaš, Goran and Jan Šnajner. 2015. Construction and evaluation of event graphs. *Natural Language Engineering*, 21(4):607–652.

Glavaš, Goran 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.

Glavaš, Goran 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.

Gonzalez-Dios, Itziar, María Jesús Aranzabe, Arantza Díaz de Ilarraza, and Haritz Salaberri. 2014. Simple or complex? Assessing the readability of basque texts. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 334–344, Dublin.

Goodfellow, Ian, 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*. Curran Associates, Inc., pages 2672–2680.

Goto, Isao, Hideki Tanaka, and Tadashi Kumano. 2015. Japanese news simplification: Task design, data set construction, and analysis of simplified text. In *Proceedings of Machine Translation Summit XV, Vol. 1: MT Researchers' Track*, pages 17–31, Miami, FL.

Gu, Jiatao, 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.

Guo, Han, 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, NM.

Hasler, Eva, 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.

Heilman, Michael 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.

Hershcovich, Daniel, 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.

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

Howcroft, David M. 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.

Hwang, William, 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, CO.

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

Jing, Hongyan. 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.

Johnson, Melvin, 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.

Kajiwara, Tomoyuki 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.

Kandula, Sasikiran, Dorothy Curtis, and Qing Zeng-Treitler. 2010. A semantic and syntactic text simplification tool for health content. In *AMIA Annual Symposium Proceedings*, pages 366–370, Washington, DC.

Kauchak, David. 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.

Kincaid, J. P., 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.

Klaper, David, Sarah Ebling, and Martin Volk. 2013. Building a German/simple German parallel corpus for automatic text simplification. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 11–19, Sofia.

Klebanov, Beata Beigman, Kevin Knight, and Daniel Marcu. 2004. Text simplification for information-seeking applications. In *Proceedings of On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE*, LNCS, number 3290, Springer Berlin Heidelberg, Berlin Heidelberg, pages 735–747.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M. Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. *CoRR*, abs/1701.02810.

Klerke, Sigrid and Anders Søgaard. 2012. DSIM, a Danish parallel corpus for text simplification. In *Proceedings of the Eight International Conference on Language Resources and Evaluation*, LREC'12, pages 4015–4018, Istanbul.

Koehn, Philipp, 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.

Lin, Chin Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Workshop on Text Summarization Branches Out*, pages 74–81, Barcelona.

Luong, Thang, 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.

Mallinson, Jonathan, 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, Valencia.

Mandya, Angrosh, 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.

Martin, Louis, 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.

Mason, Jana M. and Janet R. Kendall. 1978. Facilitating reading comprehension through text structure manipulation, Bolt, Beranek and Newman, Inc., Cambridge, MA; Illinois University, Urbana. Center for the Study of Reading.

McNamara, Danielle S., 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.

Mcnamee, Paul and James Mayfield. 2004. Character n-gram tokenization for European language text retrieval. *Information Retrieval*, 7(1–2):73–97.

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at 2013 International Conference on Learning Representations*, Scottsdale, AZ.

Mirkin, Shachar, Sriram Venkatapathy, and Marc Dymetman. 2013. Confidence-driven rewriting for improved translation. In *XIV MT Summit*, pages 257–264, Nice.

Mishra, Kshitij, 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.

Munkhdalai, Tsendsuren 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.

Napoles, Courtney 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.

Narayan, Shashi 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, MD.

Narayan, Shashi and Claire Gardent. 2016. Unsupervised sentence simplification using deep semantics. In *Proceedings of the 9th International Natural Language Generation conference*, pages 111–120, Edinburgh.

Narayan, Shashi, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. *CoRR*, abs/1707.06971.

Niklaus, Christina, 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.

Nisioi, Sergiu, 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.

Och, Franz Josef and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

Ogden, Charles Kay. 1930. *Basic English: A General Introduction with Rules and Grammar*. Kegan Paul, Trench, Trubner & Co.

Paetzold, Gustavo and Lucia Specia. 2013. Text simplification as tree transduction. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, pages 116–125, Fortaleza.

Paetzold, Gustavo 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.

Paetzold, Gustavo H., 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, Taipei.

Paetzold, Gustavo H. and Lucia Specia. 2017b. A survey on lexical simplification. *Journal of Artificial Intelligence Research*, 60:549–593.

Paetzold, Gustavo Henrique. 2016. Lexical Simplification for Non-Native English Speakers. Ph.D. thesis, University of Sheffield, Sheffield, UK.

Paetzold, Gustavo Henrique and Lucia Specia. 2016. Vicinity-driven paragraph and sentence alignment for comparable corpora. *CoRR*, abs/1612.04113.

Papineni, Kishore, 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, PA.

Pavlick, Ellie 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.

Petersen, Sarah E. 2007. Natural Language Processing Tools for Reading Level Assessment and Text Simplification for Bilingual Education. Ph.D. thesis, University of Washington, AAI3275902.

Petersen, Sarah E. and Mari Ostendorf. 2007. Text simplification for language learners: A corpus analysis. In *Proceedings of the Speech and Language Technology for Education Workshop*, SLaTE 2007, pages 69–72, Farmington, PA.

Post, Matt. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels.

Post, Matt, 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.

Quigley, S. P., D. Power, and M. Steinkamp. 1977. The language structure of deaf children. *The Volta Review*, 79(2):73–84.

Ranzato, Marc'Aurelio, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016*, San Juan.

Reiter, Ehud. 2018. A structured review of the validity of BLEU. *Computational Linguistics*, 44(3):393–401.

Rello, Luz, Ricardo Baeza-Yates, Laura Dempere-Marco, and Horacio Saggion. 2013a. Frequent words improve readability and short words improve understandability for people with dyslexia. In *Human-Computer Interaction – INTERACT 2013: 14th IFIP TC 13 International Conference*, pages 203–219, Cape Town.

Rello, Luz, 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.

Robbins, N. L. and C. Hatcher. 1981. The effects of syntax on the reading comprehension of hearing-impaired children. *The Volta Review*, 83(2):105–115.

Saggion, Horacio. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*, 10(1):1–137.

Scarton, Carolina, 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.

Scarton, Carolina, 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.

Scarton, Carolina 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.

See, Abigail, 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.

Sennrich, Rico, 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.

Shardlow, Matthew. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications (IJACSA), Special Issue on Natural Language Processing 2014*, pages 58–70, West Yorkshire, UK.

Shewan, Cynthia M. 1985. Auditory comprehension problems in adult aphasic individuals. *Human Communication Canada*, 9(5):151–155.

Siddharthan, Advaith. 2003. Preserving discourse structure when simplifying text. In *Proceedings of the 2003 European Natural Language Generation Workshop*, ENLG 2003, pages 103–110, Budapest.

Siddharthan, Advaith. 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.

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

Siddharthan, Advaith, 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.

Silveira, Sara Botelho and Antnio 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, NV.

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.

Smith, David A. 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, NY.

Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, MA.

Specia, Lúcia. 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.

Specia, Lúcia, Carolina Scarton, and Gustavo Henrique Paetzold. 2018. Quality estimation for machine translation. *Synthesis Lectures on Human Language Technologies*, 11(1):1–162.

Specia, Lúcia, Sandra Maria Aluísio, and Thiago A. Salgueiro Pardo. 2008. Manual de simplificação sintática para o português, NILC–ICMC–USP, São Carlos, SP, Brasil. Available at `http://www.nilc.icmc.usp.br/nilc/download/NILC_TR_08_06.pdf`

Sulem, Elior, 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.

Sulem, Elior, 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, LA.

Sun, Hong 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.

Surya, Sai, 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.

Štajner, Sanja, 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.

Štajner, Sanja, 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.

Štajner, Sanja and Goran Glavaš. 2017. Leveraging event-based semantics for automated text simplification. *Expert Systems with Applications*, 82:383–395.

Štajner, Sanja, 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.

Štajner, Sanja 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.

Štajner, Sanja, Maja Popović, and Hannah Béchera. 2016. Quality estimation for text simplification. In *Proceedings of the Workshop on Quality Assessment for Text Simplification - LREC 2016*, QATS 2016, pages 15–21, Portorož.

Štajner, Sanja, Maja Popović, Horacio Saggion, Lucia Specia, and Mark Fishel. 2016. Shared task on quality assessment for text simplification. In *Proceedings of the Workshop on Quality Assessment for Text Simplification - LREC 2016*, QATS 2016, pages 22–31, Portorož.

Tonelli, Sara, Alessio Palmero Aprosio, and Francesca Saltori. 2016. SIMPITIKI: A Simplification corpus for Italian. In *Proceedings of Third Italian Conference on Computational Linguistics (CLiC-it 2016) & Fifth Evaluation Campaign of Natural Language Processing and Speech Tools for Italian. Final Workshop (EVALITA 2016)*, Napoli.

Vajjala, Sowmya 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, LA.

Vajjala, Sowmya 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.

Vajjala, Sowmya 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.

Vajjala, Sowmya and Detmar Meurers. 2015. Readability-based sentence ranking for evaluating text simplification, Iowa State University.

Vanderwende, Lucy, 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.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz 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*, Curran Associates, Inc., pages 5998–6008.

Vickrey, David and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, OH.

Vu, Tu, 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, LA.

Watanabe, Willian Massami, 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, IN.

Wieting, John 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.

Williams, Adina, 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, LA.

Williams, Ronald J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256.

Woodsend, Kristian 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.

Woodsend, Kristian 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.

Wubben, Sander, 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.

Xu, Wei, 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.

Xu, Wei, 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.

Yamada, Kenji and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, ACL '01, pages 523–530, Stroudsburg, PA.

Yasseri, Taha, András Kornai, and János Kertsz. 2012. A practical approach to language complexity: A Wikipedia case study. *PLOS ONE*, 7(11):1–8.

Yatskar, Mark, 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, CA.

Zhang, Xingxing 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.

Zhao, Sanqiang, 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.

Zhu, Zhemin, 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.

Zoph, Barret 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, CA.