# Explainable Prediction of Text Complexity: The Missing Preliminaries for Text Simplification

**Cristina Gârbacea**[1], **Mengtian Guo**[4], **Samuel Carton**[3], **Qiaozhu Mei**[1,2]
[1]Department of EECS, University of Michigan, Ann Arbor
[2]School of Information, University of Michigan, Ann Arbor
[3]Department of CS, University of Colorado, Boulder
[4]School of Information and Library Science, University of North Carolina, Chapel Hill

## Abstract

Text simplification reduces the language complexity of professional content for accessibility purposes. End-to-end neural network models have been widely adopted to directly generate the simplified version of input text, usually functioning as a blackbox. We show that text simplification can be decomposed into a compact pipeline of tasks to ensure the transparency and explainability of the process. The first two steps in this pipeline are often neglected: 1) to predict whether a given piece of text needs to be simplified, and 2) if yes, to identify complex parts of the text. The two tasks can be solved separately using either lexical or deep learning methods, or solved jointly. Simply applying explainable complexity prediction as a preliminary step, the out-of-sample text simplification performance of the state-of-the-art, black-box simplification models can be improved by a large margin.

## 1 Introduction

Text simplification aims to reduce the language complexity of highly specialized textual content so that it is accessible for readers who lack adequate literacy skills, such as children, people with low education, people who have reading disorders or dyslexia, and non-native speakers of the language.

Mismatch between language complexity and literacy skills is identified as a critical source of bias and inequality in the consumers of systems built upon processing and analyzing professional text content. Research has found that it requires on average 18 years of education for a reader to properly understand the clinical trial descriptions on ClinicalTrials.gov, and this introduces a potential self-selection bias to those trials (Wu et al., 2016).

Text simplification has considerable potential to improve the fairness and transparency of text information systems. Indeed, the Simple English Wikipedia (`simple.wikipedia.org`) has been constructed to disseminate Wikipedia articles to kids and English learners. In healthcare, consumer vocabulary are used to replace professional medical terms to better explain medical concepts to the public (Abrahamsson et al., 2014). In education, natural language processing and simplified text generation technologies are believed to have the potential to improve student outcomes and bring equal opportunities for learners of all levels in teaching, learning and assessment (Mayfield et al., 2019).

Ironically, the definition of "text simplification" in literature has never been transparent. The term may refer to reducing the complexity of text at various linguistic levels, ranging all the way through replacing individual words in the text to generating a simplified document completely through a computer agent. In particular, *lexical simplification* (Devlin, 1999) is concerned with replacing complex words or phrases with simpler alternatives; *syntactic simplification* (Siddharthan, 2006) alters the syntactic structure of the sentence; *semantic simplification* (Kandula et al., 2010) paraphrases portions of the text into simpler and clearer variants. More recent approaches simplify texts in an end-to-end fashion, employing machine translation models in a monolingual setting regardless of the type of simplifications (Zhang and Lapata, 2017; Guo et al., 2018; Van den Bercken et al., 2019). Nevertheless, these models are limited on the one hand due to the absence of large-scale parallel (complex $\rightarrow$ simple) monolingual training data, and on the other hand due to the lack of interpretibility of their black-box procedures (Alva-Manchego et al., 2017).

Given the ambiguity in problem definition, there also lacks consensus on how to measure the goodness of text simplification systems, and automatic evaluation measures are perceived ineffective and sometimes detrimental to the specific procedure, in particular when they favor shorter but not necessar-



*Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, pages 1086–1097
August 1–6, 2021. 

ily simpler sentences (Napoles et al., 2011). While end-to-end simplification models demonstrate superior performance on benchmark datasets, their success is often compromised in out-of-sample, real-world scenarios (D'Amour et al., 2020).

Our work is motivated by the aspiration that increasing the transparency and explainability of a machine learning procedure may help its generalization into unseen scenarios (Doshi-Velez and Kim, 2018). We show that the general problem of text simplification can be formally decomposed into a compact and transparent pipeline of modular tasks. We present a systematic analysis of the first two steps in this pipeline, which are commonly overlooked: 1) *to predict whether a given piece of text needs to be simplified at all*, and 2) *to identify which part of the text needs to be simplified*. The second task can also be interpreted as an explanation of the first task: why a piece of text is considered complex. These two tasks can be solved separately, using either lexical or deep learning methods, or they can be solved jointly through an end-to-end, explainable predictor. Based on the formal definitions, we propose general evaluation metrics for both tasks and empirically compare a diverse portfolio of methods using multiple datasets from different domains, including news, Wikipedia, and scientific papers. We demonstrate that by simply applying explainable complexity prediction as a preliminary step, the out-of-sample text simplification performance of the state-of-the-art, black-box models can be improved by a large margin.

Our work presents a promising direction towards a transparent and explainable solution to text simplification in various domains.

## 2 Related Work

### 2.1 Text Simplification

#### 2.1.1 Identifying complex words

Text simplification at word level has been done through 1) **lexicon based** approaches, which match words to lexicons of complex/simple words (Deléger and Zweigenbaum, 2009; Elhadad and Sutaria, 2007), 2) **threshold based** approaches, which apply a threshold over word lengths or certain statistics (Leroy et al., 2013), 3) **human driven** approaches, which solicit the user's input on which words need simplification (Rello et al., 2013), and 4) **classification** methods, which train machine learning models to distinguish complex words from simple words (Shardlow, 2013). Complex word identification is also the main topic of SemEval 2016 Task 11 (Paetzold and Specia, 2016), aiming to determine whether a non-native English speaker can understand the meaning of a word in a given sentence. Significant differences exist between simple and complex words, and the latter on average are shorter, less ambiguous, less frequent, and more technical in nature. Interestingly, the frequency of a word is identified as a reliable indicator of its simplicity (Leroy et al., 2013).

While the above techniques have been widely employed for complex word identification, the results reported in the literature are rather controversial and it is not clear to what extent one technique outperforms the other in the absence of standardized high quality parallel corpora for text simplification (Paetzold, 2015). Pre-constructed lexicons are often limited and do not generalize to different domains. It is intriguing that classification methods reported in the literature are not any better than a "simplify-all" baseline (Shardlow, 2014).

#### 2.1.2 Readability assessment

Traditionally, measuring the level of reading difficulty is done through lexicon and rule-based metrics such as the age of acquisition lexicon (AoA) (Kuperman et al., 2012) and the Flesch-Kincaid Grade Level (Kincaid et al., 1975). A machine learning based approach in (Schumacher et al., 2016) extracts lexical, syntactic, and discourse features and train logistic regression classifiers to predict the relative complexity of a single sentence in a pairwise setting. The most predictive features are simple representations based on AoA norms. The perceived difficulty of a sentence is highly influenced by properties of the surrounding passage. Similar methods are used for fine-grained classification of text readability (Aluisio et al., 2010) and complexity (Štajner and Hulpuș, 2020).

#### 2.1.3 Computer-assisted paraphrasing

Simplification rules are learnt by finding words from a complex sentence that correspond to different words in a simple sentence (Alva-Manchego et al., 2017). Identifying simplification operations such as copies, deletions, and substitutions for words from parallel complex vs. simple corpora helps understand how human experts simplify text (Alva-Manchego et al., 2017). Machine translation has been employed to learn phrase-level alignments for sentence simplification (Wubben et al., 2012). Lexical and phrasal paraphrase rules are extracted

in (Pavlick and Callison-Burch, 2016). These methods are often evaluated by comparing their output to gold-standard, human-generated simplifications, using standard metrics (e.g., token-level precision, recall, F1), machine translation metrics (e.g., BLEU (Papineni et al., 2002) ), text simplification metrics (e.g. SARI (Xu et al., 2016) which rewards copying words from the original sentence), and readability metrics (among which Flesch-Kincaid Grade Level (Kincaid et al., 1975) and Flesch Reading Ease (Kincaid et al., 1975) are most commonly used). It is desirable that the output of the computational models is ultimately validated by human judges (Shardlow, 2014).

#### 2.1.4 End-to-end simplification

Neural encoder-decoder models are used to learn simplification rewrites from monolingual corpora of complex and simple sentences (Scarton and Specia, 2018; Van den Bercken et al., 2019; Zhang and Lapata, 2017; Guo et al., 2018). On one hand, these models often obtain superior performance on particular evaluation metrics, as the neural network directly optimizes these metrics in training. On the other hand, it is hard to interpret what exactly are learned in the hidden layers, and without this transparency it is difficult to adapt these models to new data, constraints, or domains. For example, these end-to-end simplification models tend not to distinguish whether the input text should or should not be simplified at all, making the whole process less transparent. When the input is already simple, the models tend to oversimplify it and deviate from its original meaning (see Section 5.3).

### 2.2 Explanatory Machine Learning

Various approaches are proposed in the literature to address the explainability and interpretability of machine learning agents. The task of providing explanations for black-box models has been tackled either at a local level by explaining individual predictions of a classifier (Ribeiro et al., 2016), or at a global level by providing explanations for the model behavior as a whole (Letham et al., 2015). More recently, differential explanations are proposed to describe how the logic of a model varies across different subspaces of interest (Lakkaraju et al., 2019). Layer-wise relevance propagation (Arras et al., 2017) is used to trace backwards text classification decisions to individual words, which are assigned scores to reflect their separate contribution to the overall prediction.

LIME (Ribeiro et al., 2016) is a model-agnostic explanation technique which can approximate any machine learning model locally with another sparse linear interpretable model. SHAP (Lundberg and Lee, 2017) evaluates Shapley values as the average marginal contribution of a feature value across all possible coalitions by considering all possible combinations of inputs and all possible predictions for an instance. Explainable classification can also be solved simultaneously through a neural network, using hard attentions to select individual words into the "rationale" behind a classification decision (Lei et al., 2016). Extractive adversarial networks employs a three-player adversarial game which addresses high recall of the rationale (Carton et al., 2018). The model consists of a generator which extracts an attention mask for each token in the input text, a predictor that cooperates with the generator and makes prediction from the rationale (words attended to), and an adversarial predictor that makes predictions from the remaining words in the inverse rationale. The minimax game between the two predictors and the generator is designed to ensure all predictive signals are included into the rationale.

No prior work has addressed the explainability of text complexity prediction. We fill in this gap.

## 3 An Explainable Pipeline for Text Simplification

We propose a unified view of text simplification which is decomposed into several carefully designed sub-problems. These sub-problems generalize over many approaches, and they are logically dependent on and integratable with one another so that they can be organized into a compact pipeline.

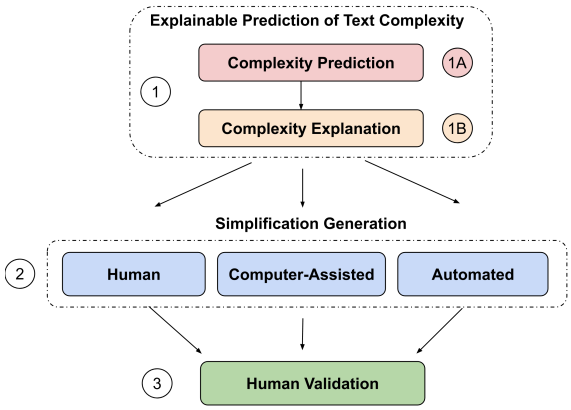


Figure 1: A text simplification pipeline. Explainable prediction of text complexity is the preliminary of any human-based, computer assisted, or automated system.

The first conceptual block in the pipeline (Figure 1) is concerned with explainable prediction of the complexity of text. It consists of two sub-tasks: 1) *prediction*: classifying a given piece of text into two categories, needing simplification or not; and 2) *explanation*: highlighting the part of the text that needs to be simplified. The second conceptual block is concerned with simplification generation, the goal of which is to generate a new, simplified version of the text that needs to be simplified. This step could be achieved through completely manual effort, or a computer-assisted approach (e.g., by suggesting alternative words and expressions), or a completely automated method (e.g., by self-translating into a simplified version). The second building block is piped into a step of human judgment, where the generated simplification is tested, approved, and evaluated by human practitioners.

One could argue that for an automated simplification generation system the first block (complexity prediction) is not necessary. We show that it is not the case. Indeed, it is unlikely that every piece of text needs to be simplified in reality, and instead the system should first decide whether a sentence needs to be simplified or not. Unfortunately such a step is often neglected by existing end-to-end simplifiers, thus their performance is often biased towards the complex sentences that are selected into their training datasets at the first place and doesn't generalize well to simple inputs. Empirically, when these models are applied to out-of-sample text which shouldn't be simplified at all, they tend to oversimplify the input and result in a deviation from its original meaning (see Section 5.3).

One could also argue that an explanation component (1B) is not mandatory in certain text simplification practices, in particular in an end-to-end neural generative model that does not explicitly identify the complex parts of the input sentence. In reality, however, it is often necessary to highlight the differences between the original sentence and the simplified sentence (which is essentially a variation of 1B) to facilitate the validation and evaluation of these black-boxes. More generally, the explainability/interpretability of a machine learning model has been widely believed to be an indispensable factor to its fidelity and fairness when applied to the real world (Lakkaraju et al., 2019). Since the major motivation of text simplification is to improve the fairness and transparency of text information systems, it is critical to explain the rationale behind the simplification decisions, even if they are made through a black-box model.

Without loss of generality, we can formally define the sub-tasks 1A, 1B, and 2- in the pipeline:

**Definition 3.1.** (Complexity Prediction). Let text $d \in D$ be a sequence of tokens $w_1 w_2 ... w_n$. The task of complexity prediction is to find a function $f : D \to \{0, 1\}$ such that $f(d) = 1$ if $d$ needs to be simplified, and $f(d) = 0$ otherwise.

**Definition 3.2.** (Complexity Explanation). Let $d$ be a sequence of tokens $w_1 w_2 ... w_n$ and $f(d) = 1$. The task of complexity explanation/highlighting is to find a function $h : D \to \{0, 1\}^n$ s.t. $h(d) = c_1 c_2 ... c_n$, where $c_i = 1$ means $w_i$ will be highlighted as a complex portion of $d$ and $c_i = 0$ otherwise. We denote $d|h(d)$ as the highlighted part of $d$ and $d|\neg h(d)$ as the unhighlighted part of d.

**Definition 3.3.** (Simplification Generation). Let $d$ be a sequence of tokens $w_1 w_2 ... w_n$ and $f(d) = 1$. The task of simplification generation is to find a function $g : D \to D'$ s.t. $g(d, f(d), h(d)) = d'$, where $d' = w'_1 w'_2 ... w'_m$ and $f(d') = 0$, subject to the constraint that $d'$ preserves the meaning of $d$.

In this paper, we focus on an empirical analysis of the first two sub-tasks of explainable prediction of text complexity (1A and 1B), which are the preliminaries of any reasonable text simplification practice. We leave aside the detailed analysis of simplification generation (2-) for now, as there are many viable designs of $g(\cdot)$ in practice, spanning the spectrum between completely manual and completely automated. Since this step is not the focus of this paper, we intend to leave the definition of simplification generation highly general.

Note that the definitions of complexity prediction and complexity explanation can be naturally extended to a continuous output, where $f(\cdot)$ predicts the complexity level of $d$ and $h(\cdot)$ predicts the complexity weight of $w_i$. The continuous output would align the problem more closely to readability measures (Kincaid et al., 1975). In this paper, we stick to the binary output because a binary action (to simplify or not) is almost always necessary in reality even if a numerical score is available.

Note that the definition of complexity explanation is general enough for existing approaches. In lexical simplification where certain words in a complex vocabulary $V$ are identified to explain the complexity of a sentence, it is equivalent to highlighting every appearance of these words in $d$, or $\forall w_i \in V, c_i = 1$. In automated simplification

where there is a self-translation function $g(d) = d'$, $h(d)$ can be simply instantiated as a function that returns a sequence alignment of $d$ and $d'$. Such reformulation helps us define unified evaluation metrics for complexity explanation (see Section 4).

It is also important to note that the dependency between the components, especially complexity prediction and explanation, does not restrict them to be done in isolation. These sub-tasks can be done either separately, or jointly with an end-to-end approach as long as the outputs of $f, h, g$ are all obtained (so that transparency and explainability are preserved). In Section 4, we include both separate models and end-to-end models for explanatory complexity predication in one shot.

## 4 Empirical Analysis of Complexity Prediction and Explanation

With the pipeline formulation, we are able to compare a wide range of methods and metrics for the sub-tasks of text simplification. We aim to understand how difficult they are in real-world settings and which method performs the best for which task.

### 4.1 Complexity Prediction

#### 4.1.1 Candidate Models

We examine a wide portfolio of deep and shallow binary classifiers to distinguish complex sentences from simple ones. Among the shallow models we use Naive Bayes (NB), Logistic Regression (LR), Support Vector Machines (SVM) and Random Forests (RF) classifiers trained with unigrams, bigrams and trigrams as features. We also train the classifiers using the lexical and syntactic features proposed in (Schumacher et al., 2016) combined with the $n$-gram features (denoted as "enriched features"). We include neural network models such as word and char-level Long Short-Term Memory Network (LSTM) and Convolutional Neural Networks (CNN). We also employ a set of state-of-the-art pre-trained neural language models, fine-tuned for complexity prediction; we introduce them below.

ULMFiT (Howard and Ruder, 2018) a language model on a large general corpus such as WikiText-103 and then fine-tunes it on the target task using slanted triangular rates, and gradual unfreezing. We use the publicly available implementation[1] of the model with two fine-tuning epochs for each dataset and the model quickly adapts to a new task.

[1] https://docs.fast.ai/tutorial.text.html, retrieved on 5/31/2021.

BERT (Devlin et al., 2019) trains deep bidirectional language representations and has greatly advanced the state-of-the-art for many natural language processing tasks. The model is pre-trained on the English Wikipedia as well as the Google Book Corpus. Due to computational constraints, we use the 12 layer BERT base pre-trained model and fine-tune it on our three datasets. We select the best hyperparameters based on each validation set.

XLNeT (Yang et al., 2019) overcomes the limitations of BERT (mainly the use of masks) with a permutation-based objective which considers bidirectional contextual information from all positions without data corruption. We use the 12 layer XLNeT base pre-trained model on the English Wikipedia, the Books corpus (similar to BERT), Giga5, ClueWeb 2012-B, and Common Crawl.

#### 4.1.2 Evaluation Metric

We evaluate the performance of complexity prediction models using *classification accuracy* on balanced training, validation, and testing datasets.

### 4.2 Complexity Explanation

#### 4.2.1 Candidate Models

We use *LIME* in combination with LR and LSTM classifiers, *SHAP* on top of LR, and the *extractive adversarial networks* which jointly conducts complexity prediction and explanation. We feed each test complex sentence as input to these explanatory models and compare their performance at identifying tokens (words and punctuation) that need to be removed or replaced from the input sentence.

We compare these explanatory models with three baseline methods: 1) *Random highlighting*: randomly draw the size and the positions of tokens to highlight; 2) *Lexicon based highlighting*: highlight words that appear in the Age-of-Acquisition (AoA) lexicon (Kuperman et al., 2012), which contains ratings for 30,121 English content words (nouns, verbs, and adjectives) indicating the age at which a word is acquired; and 3) *Feature highlighting*: highlight the most important features of the best performing LR models for complexity prediction.

#### 4.2.2 Evaluation Metrics

Evaluation of explanatory machine learning is an open problem. In the context of complexity explanation, when the ground truth of highlighted tokens ($y_c(d) = c_1 c_2 ... c_n$, $c_i \in \{0, 1\}$) in each complex sentence $d$ is available, we can compare the output of complexity explanation $h(d)$ with $y_c(d)$. Such

per-token annotations are usually not available in scale. To overcome this, given a complex sentence $d$ and its simplified version $d'$, we assume that all tokens $w_i$ in $d$ which are absent in $d'$ are candidate words for deletion or substitution during the text simplification process and should therefore be highlighted in complexity explanation (i.e., $c_i = 1$).

In particular, we use the following evaluation metrics for complexity explanation: 1) *Tokenwise Precision (P)*, which measures the proportion of highlighted tokens in $d$ that are truly removed in $d'$; 2) *Tokenwise Recall (R)*, which measures the proportion of tokens removed in $d'$ that are actually highlighted in $d$; 3) *Tokenwise F1*, the harmonic mean of $P$ and $R$; 4) word-level *Edit distance (ED)* (Levenshtein, 1966): between the **unhighlighted** part of $d$ and the simplified document $d'$. Intuitively, a more successful complexity explanation would highlight most of the tokens that need to be simplified, thus the remaining parts in the complex sentences will be closer to the simplified version, achieving a lower edit distance (we also explore ED with a higher penalty cost for the substitution operation, namely values of 1, 1.5 and 2); and 5) *Translation Edit Rate (TER)* (Snover et al., 2006), which measures the minimum number of edits needed to change a hypothesis (the unhighlighted part of $d$) so that it exactly matches the closest references (the simplified document $d'$). Note these metrics are all proxies of the real editing process from $d$ to $d'$. When token-level edit history is available (e.g., through track changes), it is better to compare the highlighted evaluation with these true changes made. We compute all the metrics at sentence level and macro-average them.

## 4.3 Experiment Setup

### 4.3.1 Datasets

We use three different datasets (Table 1) which cover different domains and application scenarios of text simplification. Our first dataset is *Newsela* (Xu et al., 2015), a corpus of news articles simplified by professional news editors. In our experiments we use the parallel Newsela corpus with the training, validation, and test splits made available in (Zhang and Lapata, 2017). Second, we use the *WikiLarge* corpus introduced in (Zhang and Lapata, 2017). The training subset of WikiLarge is created by assembling datasets of parallel aligned Wikipedia - Simple Wikipedia sentence pairs available in the literature (Kauchak, 2013). While this training set is obtained through automatic alignment procedures which can be noisy, the validation and test subsets of WikiLarge contain complex sentences with simplifications provided by Amazon Mechanical Turk workers (Xu et al., 2016); we increase the size of validation and test on top of the splits made available in (Zhang and Lapata, 2017). Third, we use the dataset released by the *Biendata* competition[2], which asks participants to match research papers from various scientific disciplines with press releases that describe them. Arguably, rewriting scientific papers into press releases has mixed objectives that are not simply text simplification. We include this task to test the generalizability of our explainable pipeline (over various definitions of simplification). We use alignments at title level. On average, a complex sentence in Newsela, WikiLarge, Biendata contains 23.07, 25.14, 13.43 tokens, and the corresponding simplified version is shorter, with 12.75, 18.56, 10.10 tokens.

Table 1: Aligned complex-simple sentence pairs.

| Dataset | Training | Validation | Test |
|---|---|---|---|
| *Newsela* | 94,208 pairs | 1,129 pairs | 1,077 pairs |
| *WikiLarge* | 208,384 pairs | 29,760 pairs | 59,546 pairs |
| *Biendata* | 29,700 pairs | 4,242 pairs | 8,486 pairs |

### 4.3.2 Ground Truth Labels

The original datasets contain aligned complex-simple sentence pairs instead of classification labels for complexity prediction. We infer ground-truth complexity labels for each sentence such that: *label 1* is assigned to every sentence for which there is an aligned simpler version not identical to itself (the sentence is complex and needs to be simplified); *label 0* is assigned to all simple counterparts of complex sentences, as well as to those sentences that have corresponding "simple" versions identical to themselves (i.e., these sentences do not need to be simplified). For complex sentences that have label 1, we further identify which tokens are not present in corresponding simple versions.

### 4.3.3 Model Training

For all shallow and deep classifiers we find the best hyperparameters using random search on validation, with early stopping. We use grid search on validation to fine-tune hyperparameters of the pre-trained models, such as maximum sequence

[2] https://www.biendata.com/competition/hackathon, retrieved on 5/31/2021.

length, batch size, learning rate, and number of epochs. For ULMFit on Newsela, we set batch size to 128 and learning rate to 1e-3. For BERT on WikiLarge, batch size is 32, learning rate is 2e-5, and maximum sequence length is 128. For XLNeT on Biendata, batch size is 32, learning rate is 2e-5, and maximum sequence length is 32.

We use grid search on validation to fine-tune the complexity explanation models, including the extractive adversarial network. For LR and LIME we determine the maximum number of words to highlight based on TER score on validation (please see Table 2); for SHAP we highlight all features with positive assigned weights, all based on TER.

Table 2: Maximum numbers of most important LR features and features highlighted by LIME.

| Model | Newsela | WikiLarge | Biendata |
|---|---|---|---|
| *LR* | 200 features | 20,000 features | 200 features |
| *LIME & LR* | 10 features | 50 features | 10 features |
| *LIME & LSTM* | 60 features | 20 features | 40 features |

For extractive adversarial networks batch size is set to 256, learning rate is 1e-4, and adversarial weight loss equals 1; in addition, sparsity weight is 1 for Newsela and Biendata, and 0.6 for WikiLarge; lastly, coherence weight is 0.05 for Newsela, 0.012 for WikiLarge, and 0.0001 for Biendata.

## 5 Results

### 5.1 Complexity Prediction

In Table 3, we evaluate how well the representative shallow, deep, and pre-trained classification models can determine whether a sentence needs to be simplified at all. We test for statistical significance of the best classification results compared to all other models using a two-tailed z-test.

In general, the best performing models can achieve around 80% accuracy on two datasets (Newsela and WikiLarge) and a very high performance on the Biendata ($> 95\%$). This difference presents the difficulty of complexity prediction in different domains, and distinguishing highly specialized scientific content from public facing press releases is relatively easy (Biendata).

Deep classification models in general outperform shallow ones, however with carefully designed handcrafted features and proper hyperparameter optimization shallow models tend to approach to the results of the deep classifiers. Overall models pre-trained on large datasets and fine-tuned for text simplification yield superior classification performance. For Newsela the best performing classification model is ULMFiT (accuracy = 80.83%, recall = 76.87%), which significantly ($p < 0.01$) surpasses all other classifiers except for XLNeT and CNN (char-level). On WikiLarge, BERT presents the highest accuracy ($81.45\%, p < 0.01$), and recall = 83.30%. On Biendata, XLNeT yields the highest accuracy ($95.48\%, p < 0.01$) with recall = 94.93%, although the numerical difference to other pre-trained language models is small. This is consistent with recent findings in other natural language processing tasks (Cohan et al., 2019).

Table 3: Accuracy of representative shallow*, deep, and pre-trained models for complexity prediction. **BOLD**: best performing models.

| Classifier | Newsela | WikiLarge | Biendata |
|---|---|---|---|
| *NB n-grams* | 73.10 % | 62.70 % | 84.30 % |
| *NB enriched features* | 73.10 % | 63.10 % | 86.00 % |
| *LR n-grams* | 75.30 % | 71.90 % | 89.60 % |
| *LR enriched features* | 76.30 % | 72.60 % | 91.70 % |
| *SVM n-grams* | 75.20 % | 71.90 % | 89.50 % |
| *SVM enriched features* | 77.39 % | 70.16 % | 88.60 % |
| *RF n-grams* | 71.50 % | 71.50 % | 84.60 % |
| *RF enriched features* | 74.40 % | 73.40 % | 87.00 % |
| *LSTM (word-level)* | 73.31 % | 71.62 % | 89.87 % |
| *CNN (word-level)* | 70.71 % | 69.27 % | 89.05 % |
| *CNN (char-level)* | 78.83%† | 74.88 % | 88.00 % |
| *CNN (word & char-level)* | 75.90 | 74.00 % | 92.30 % |
| *Extractive Adversarial Networks* | 72.76 % | 71.50 % | 88.64 % |
| *ULMFiT* | **80.83%**** | 74.80 % | 94.17 % |
| *BERT* | 77.15 % | **81.45%**** | 94.43 % |
| *XLNeT* | 78.83%† | 73.49 % | **95.48%**** |

* Shallow models perform similarly and some are omitted for space; Difference between the best performing model and other models is statistically significant: $p < 0.05$ (*), $p < 0.01$ (**), except for †: difference between this model and the best performing model is not statistically significant.

### 5.2 Complexity Explanation

We evaluate how well complexity classification can be explained, or how accurately the complex parts of a sentence can be highlighted.

Results (Table 4) show that highlighting words in the AoA lexicon or LR features are rather strong baselines, indicating that most complexity of a sentence still comes from word usage. Highlighting more LR features leads to a slight drop in precision and a better recall. Although LSTM and LR perform comparably on complexity classification, using LIME to explain LSTM presents better recall, F1, and TER (at similar precision) compared to using LIME to explain LR. The LIME & LSTM combination is reasonably strong on all datasets, as is SHAP & LR. TER is a reliable indicator of the difficulty of the remainder (unhighlighted part) of the complex sentence. ED with a substitution penalty of 1.5 efficiently captures the variations among the explanations. On Newsela and Bien-

Table 4: Results for complexity explanation. P, R and F1 - the higher the better; TER and ED 1.5 - the lower the better. **BOLD** & Underlined: best & second best.

| Dataset | Explanation Model | P | R | F1 | TER | ED 1.5 |
|---|---|---|---|---|---|---|
| Newsela | Random | 0.515 | 0.487 | 0.439 | 0.985 | 13.825 |
| | AoA lexicon | **0.556** | 0.550 | 0.520 | 0.867 | 12.899 |
| | LR Features | 0.522 | 0.250 | 0.321 | 0.871 | 12.103 |
| | LIME & LR | 0.535 | 0.285 | 0.343 | 0.924 | 12.459 |
| | LIME & LSTM | 0.543 | **0.818** | **0.621** | 0.852 | 11.991 |
| | SHAP & LR | 0.553 | 0.604 | 0.546 | 0.848 | 12.656 |
| | Extractive Networks | 0.530 | 0.567 | 0.518 | **0.781** | **11.406** |
| WikiLarge | Random | 0.412 | 0.439 | 0.341 | 1.546 | 17.028 |
| | AoA lexicon | 0.427 | 0.409 | 0.357 | 1.516 | 16.731 |
| | LR Features | 0.442 | 0.525 | 0.413 | 0.993 | 17.933 |
| | LIME & LR | 0.461 | 0.509 | 0.415 | **0.988** | 18.162 |
| | LIME & LSTM | **0.880** | 0.470 | 0.595 | 1.961 | 25.051 |
| | SHAP & LR | 0.842 | **0.531** | **0.633** | 1.693 | 22.811 |
| | Extractive Networks | 0.452 | 0.429 | 0.359 | 1.434 | **16.407** |
| Biendata | Random | 0.743 | 0.436 | 0.504 | 1.065 | 12.921 |
| | AoA lexicon | 0.763 | 0.383 | 0.475 | 1.064 | 13.247 |
| | LR Features | 0.796 | 0.257 | 0.374 | 0.979 | 10.851 |
| | LIME & LR | **0.837** | 0.466 | 0.577 | 0.982 | **10.397** |
| | LIME & LSTM | 0.828 | 0.657 | 0.713 | **0.952** | 16.568 |
| | SHAP & LR | 0.825 | 0.561 | 0.647 | 0.979 | 11.908 |
| | Extractive Networks | 0.784 | **0.773** | **0.758** | 0.972 | 10.678 |

data, the extractive adversarial networks yield solid performances (especially TER and ED 1.5), indicating that jointly making predictions and generating explanations reinforces each other. Table 5 provides examples of highlighted complex sentences by each explanatory model.

### 5.3 Benefit of Complexity Prediction

One may question whether explainable prediction of text complexity is still a necessary preliminary step in the pipeline if a strong, end-to-end simplification generator is used. We show that it is. We consider the scenario where a pre-trained, end-to-end text simplification model is blindly applied to texts regardless of their complexity level, compared to only simplifying those considered complex by the best performing complexity predictor in Table 3. Such a comparison demonstrates whether adding complexity prediction as a preliminary step is beneficial to a text simplification process when a state-of-the-art, end-to-end simplifier is already in place. From literature we select the current best text simplification models on WikiLarge and Newsela which have released pre-trained models:

- ACCESS (Martin et al., 2020), a controllable sequence-to-sequence simplification model that reported the highest performance (41.87 SARI) on WikiLarge.
- Dynamic Multi-Level Multi-Task Learning for Sentence Simplification (DMLMTL) (Guo et al., 2018), which reported the highest performance (33.22 SARI) on Newsela.

We apply the author-released, pre-trained ACCESS and DMLMTL on all sentences from the validation and testing sets of all three datasets. We do not use the training examples as the pre-trained models may have already seen them. Presumably, a smart model should **not** further simplify an input sentence if it is already simple enough. However, to our surprise, a majority of the *out-of-sample simple* sentences are still changed by both models (above 90% by DMLMTL and above 70% by ACCESS, please see Table 6).

We further quantify the difference with vs. without complexity prediction as a preliminary step. Intuitively, without complexity prediction, an already simple sentence is likely to be overly simplified and result in a loss in text simplification metrics. In contrast, an imperfect complexity predictor may mistaken a complex sentence as simple, which misses the opportunity of simplification and results in a loss as well. The empirical question is which loss is higher. From Table 7, we see that after directly adding a complexity prediction step before either of the state-of-the-art simplification models, there is a considerable drop of errors in three text simplification metrics: Edit Distance (ED), TER, and Fréchet Embedding Distance (FED) that measures the difference of a simplified text and the ground-truth in a semantic space (de Masson d'Autume et al., 2019). For ED alone, the improvements are between 30% to 50%. This result is very encouraging: considering that the complexity predictors are only 80% accurate and the complexity predictor and the simplification models don't depend on each other, there is considerable room to optimize this gain. Indeed, the benefit is higher on Biendata where the complexity predictor is more accurate.

Qualitatively, one could frequently observe syntactic, semantic, and logical mistakes in the model-simplified version of *simple* sentences. We give a few examples below.

- In Ethiopia, HIV disclosure is low → In Ethiopia , HIV is low (ACCESS)
- Mustafa Shahbaz , 26 , was shopping for books about science . → Mustafa Shahbaz , 26 years old , was a group of books about science . (ACCESS)
- New biomarkers for the diagnosis of Alzheimer's → New biomarkers are diagnosed with Alzheimer (ACCESS)

Table 5: Explanations of complexity predictions (in red). Extractive network obtains a higher recall.

| Explanatory Model | Complexity Explanation |
|---|---|
| LIME & LR | Their fatigue changes their voices , but they 're still on the freedom highway . |
| LIME & LSTM | Their fatigue changes their voices , but they 're still on the freedom highway . |
| SHAP & LR | Their fatigue changes their voices , but they 're still on the freedom highway . |
| Extractive Networks | Their fatigue changes their voices , but they 're still on the freedom highway . |
| Simple sentence | Still , they are fighting for their rights . |
| LIME & LR | Digitizing physically preserves these fragile papers and allows people to see them , he said . |
| LIME & LSTM | Digitizing physically preserves these fragile papers and allows people to see them , he said . |
| SHAP & LR | Digitizing physically preserves these fragile papers and allows people to see them , he said . |
| Extractive Networks | Digitizing physically preserves these fragile papers and allows people to see them , he said . |
| Simple sentence | The papers are old and fragile , he said . |

Table 6: Percentage of out-of-sample simple sentences changed by pre-trained, end-to-end simplification models. Ideal value is 0%.

| Dataset | Pre-trained Model | Validation | Testing |
|---|---|---|---|
| Newsela | ACCESS | 72.73 % | 75.50 % |
| | DMLMTL | 90.48 % | 91.69 % |
| WikiLarge | ACCESS | 70.83 % | 71.12 % |
| | DMLMTL | 95.20 % | 95.61 % |
| Biendata | ACCESS | 94.25 % | 93.66 % |
| | DMLMTL | 98.88 % | 98.73 % |

Table 7: Out-of-sample performance of simplification models. ED, TER, FED metrics: the lower the better. Adding complexity prediction as preliminary step reduces simplification error by a wide margin.

| Dataset | Sentence Pairs | Metric | ACCESS | DMLMTL |
|---|---|---|---|---|
| Newsela | No complexity prediction (simplify everything) | ED | 4.044 | 12.212 |
| | | TER | 0.175 | 1.611 |
| | | FED | 0.016 | 0.170 |
| | With complexity prediction (predicted simple: no change) | ED | **2.631** (-35%) | **8.677** (-29%) |
| | | TER | 0.089 (-49%) | 1.149 (-29%) |
| | | FED | 0.006 (-63%) | 0.066 (-61%) |
| WikiLarge | No Complexity Prediction (simplify everything) | ED | 5.857 | 16.920 |
| | | TER | 0.208 | 2.328 |
| | | FED | 0.004 | 0.143 |
| | With Complexity Prediction (predicted simple: no change) | ED | **4.021** (-31%) | **10.566** (-38%) |
| | | TER | 0.132 (-37%) | 1.452 (-38%) |
| | | FED | 0.002 (-50%) | 0.049 (-66%) |
| Biendata | No Complexity Prediction (simplify everything) | ED | 3.796 | 9.030 |
| | | TER | 0.254 | 1.348 |
| | | FED | 0.033 | 0.131 |
| | With Complexity Prediction (predicted simple: no change) | ED | **1.887** (-50%) | **5.249** (-42%) |
| | | TER | 0.114 (-55%) | 0.819 (-39%) |
| | | FED | 0.009 (-73%) | 0.051 (-61%) |

- Healthy diet linked to lower risk of chronic lung disease → Healthy diet linked to lung disease (DMLMTL)
- Dramatic changes needed in farming practices to keep pace with climate change → changes needed to cause climate change (DMLMTL)
- Social workers can help patients recover from mild traumatic brain injuries → Social workers can cause better problems . (DMLMTL)

All these qualitative and quantitative results suggest that the state-of-the-art black-box models tend to oversimplify and distort the meanings of out-of-sample input that is already simple. Evidently, the lack of transparency and explainability has limited the application of these end-to-end black-box models in reality, especially to out-of-sample data, context, and domains. The pitfall can be avoided with the proposed pipeline and simply with explainable complexity prediction as a preliminary step. Even though this explainable preliminary does not necessarily reflect how a black-box simplification model "thinks", adding it to the model is able to yield better out-of-sample performance.

## 6 Conclusions

We formally decompose the ambiguous notion of text simplification into a compact, transparent, and logically dependent pipeline of sub-tasks, where explainable prediction of text complexity is identified as the preliminary step. We conduct a systematic analysis of its two sub-tasks, namely complexity prediction and complexity explanation, and show that they can be either solved separately or jointly through an extractive adversarial network. While pre-trained neural language models achieve significantly better performance on complexity prediction, an extractive adversarial network that solves the two tasks jointly presents promising advantage in complexity explanation. Using complexity prediction as a preliminary step reduces the error of the state-of-the-art text simplification models by a large margin. Future work should integrate rationale extractor into the pre-trained neural language models and extend it for simplification generation.

## Acknowledgement

This work is in part supported by the National Science Foundation under grant numbers 1633370 and 1620319 and by the National Library of Medicine under grant number 2R01LM010681-05.

## References


Emil Abrahamsson, Timothy Forni, Maria Skeppstedt, and Maria Kvist. 2014. Medical text simplification using synonym replacement: Adapting assessment of word difficulty to a compounding language. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, pages 57–65.

Sandra Aluisio, Lucia Specia, Caroline Gasperin, and Carolina Scarton. 2010. Readability assessment for text simplification. In *Proceedings of the NAACL HLT 2010 Fifth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 1–9.

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

Leila Arras, Franziska Horn, Grégoire Montavon, Klaus-Robert Müller, and Wojciech Samek. 2017. " what is relevant in a text document?": An interpretable machine learning approach. *PloS one*, 12(8):e0181142.

Laurens Van den Bercken, Robert-Jan Sips, and Christoph Lofi. 2019. Evaluating neural text simplification in the medical domain. In *The World Wide Web Conference*, pages 3286–3292. ACM.

Samuel Carton, Qiaozhu Mei, and Paul Resnick. 2018. Extractive adversarial networks: High-recall explanations for identifying personal attacks in social media posts. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3497–3507.

Arman Cohan, Iz Beltagy, Daniel King, Bhavana Dalvi, and Daniel S Weld. 2019. Pretrained language models for sequential sentence classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3684–3690.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. 2020. Underspecification presents challenges for credibility in modern machine learning. *arXiv preprint arXiv:2011.03395*.

Louise Deléger and Pierre Zweigenbaum. 2009. Extracting lay paraphrases of specialized expressions from monolingual comparable medical corpora. In *Proceedings of the 2nd Workshop on Building and Using Comparable Corpora: from Parallel to Non-parallel Corpora*, pages 2–10. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Siobhan Lucy Devlin. 1999. *Simplifying natural language for aphasic readers.* Ph.D. thesis, University of Sunderland.

Finale Doshi-Velez and Been Kim. 2018. Considerations for evaluation and generalization in interpretable machine learning. In *Explainable and interpretable models in computer vision and machine learning*, pages 3–17. Springer.

Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 49–56. Association for Computational Linguistics.

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. 2018. Dynamic multi-level multi-task learning for sentence simplification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 462–476.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339.

Sasikiran Kandula, Dorothy Curtis, and Qing Zeng-Treitler. 2010. A semantic and syntactic text simplification tool for health content. In *AMIA annual symposium proceedings*, volume 2010, page 366. American Medical Informatics Association.

David Kauchak. 2013. Improving text simplification language modeling using unsimplified text data. In *Proceedings of the 51st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pages 1537–1546.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel.

Victor Kuperman, Hans Stadthagen-Gonzalez, and Marc Brysbaert. 2012. Age-of-acquisition ratings for 30,000 english words. *Behavior research methods*, 44(4):978–990.

Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Jure Leskovec. 2019. Faithful and customizable explanations of black box models. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 131–138.

Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2016. Rationalizing neural predictions. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 107–117.

Gondy Leroy, James E Endicott, David Kauchak, Obay Mouradi, and Melissa Just. 2013. User evaluation of the effects of a text simplification algorithm using term familiarity on perception, understanding, learning, and information retention. *Journal of medical Internet research*, 15(7):e144.

Benjamin Letham, Cynthia Rudin, Tyler H McCormick, David Madigan, et al. 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *The Annals of Applied Statistics*, 9(3):1350–1371.

Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710.

Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*, pages 4765–4774.

Louis Martin, Éric Villemonte de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4689–4698.

Cyprien de Masson d'Autume, Shakir Mohamed, Mihaela Rosca, and Jack Rae. 2019. Training language gans from scratch. In *Advances in Neural Information Processing Systems*, pages 4302–4313.

Elijah Mayfield, Michael Madaio, Shrimai Prabhumoye, David Gerritsen, Brittany McLaughlin, Ezekiel Dixon-Román, and Alan W Black. 2019. Equity beyond bias in language technologies for education. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 444–460.

Courtney Napoles, Benjamin Van Durme, and Chris Callison-Burch. 2011. Evaluating sentence compression: Pitfalls and suggested remedies. In *Proceedings of the Workshop on Monolingual Text-To-Text Generation*, pages 91–97.

Gustavo Paetzold. 2015. Reliable lexical simplification for non-native speakers. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 9–16.

Gustavo Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 560–569.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Ellie Pavlick and Chris Callison-Burch. 2016. Simple ppdb: A paraphrase database for simplification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–148.

Luz Rello, Ricardo Baeza-Yates, Stefan Bott, and Horacio Saggion. 2013. Simplify or help?: text simplification strategies for people with dyslexia. In *Proceedings of the 10th International Cross-Disciplinary Conference on Web Accessibility*. ACM.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "why should I trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1135–1144.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718.

Elliot Schumacher, Maxine Eskenazi, Gwen Frishkoff, and Kevyn Collins-Thompson. 2016. Predicting the relative difficulty of single sentences with and without surrounding context. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1871–1881.

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

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*, 4(1):77–109.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Sanja Štajner and Ioana Hulpuș. 2020. When shallow is good enough: Automatic assessment of conceptual text complexity using shallow semantic features. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 1414–1422.

Danny TY Wu, David A Hanauer, Qiaozhu Mei, Patricia M Clark, Lawrence C An, Joshua Proulx, Qing T Zeng, VG Vinod Vydiswaran, Kevyn Collins-Thompson, and Kai Zheng. 2016. Assessing the readability of clinicaltrials. gov. *Journal of the American Medical Informatics Association*, 23(2):269–275.

Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 1015–1024. Association for Computational Linguistics.

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

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in Neural Information Processing Systems*, 32:5753–5763.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594.