Neural CRF Model for Sentence Alignment in Text Simplification

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, Wei Xu
Department of Computer Science and Engineering
The Ohio State University
{jiang.1530, maddela.4, lan.105,zhong.536, xu.1265}@osu.edu

Abstract

The success of a text simplification system heavily depends on the quality and quantity of complex-simple sentence pairs in the training corpus, which are extracted by aligning sentences between parallel articles. To evaluate and improve sentence alignment quality, we create two manually annotated sentence-aligned datasets from two commonly used text simplification corpora, Newsela and Wikipedia. We propose a novel neural CRF alignment model which not only leverages the sequential nature of sentences in parallel documents but also utilizes a neural sentence pair model to capture semantic similarity. Experiments demonstrate that our proposed approach outperforms all the previous work on monolingual sentence alignment task by more than 5 points in F1. We apply our CRF aligner to construct two new text simplification datasets, NEWSELA-AUTO and WIKI-AUTO, which are much larger and of better quality compared to the existing datasets. A Transformer-based seq2seq model trained on our datasets establishes a new state-of-the-art for text simplification in both automatic and human evaluation.\footnote{Code and data are available at: https://github.com/chaojiang06/wiki-auto. Newsela data need to be requested at: https://newsela.com/data/.

1 Introduction

Text simplification aims to rewrite complex text into simpler language while retaining its original meaning (Saggion, 2017). Text simplification can provide reading assistance for children (Kajiwara et al., 2013), non-native speakers (Petersen and Ostendorf, 2007; Pellow and Eskenazi, 2014), non-expert readers (Elhadad and Sutaria, 2007; Siddharthan and Katsos, 2010), and people with language disorders (Rello et al., 2013). As a preprocessing step, text simplification can also improve the performance of many natural language processing (NLP) tasks, such as parsing (Chandrasekar et al., 1996), semantic role labelling (Vickrey and Koller, 2008), information extraction (Miwa et al., 2010), summarization (Vanderwende et al., 2007; Xu and Grishman, 2009), and machine translation (Chen et al., 2012; Štajner and Popovic, 2016).

Automatic text simplification is primarily addressed by sequence-to-sequence (seq2seq) models whose success largely depends on the quality and quantity of the training corpus, which consists of pairs of complex-simple sentences. Two widely used corpora, NEWSELA (Xu et al., 2015) and WIKILARGE (Zhang and Lapata, 2017), were created by automatically aligning sentences between comparable articles. However, due to the lack of reliable annotated data,\footnote{Hwang et al. (2015) annotated 46 article pairs from Simple-Normal Wikipedia corpus; however, its annotation is noisy, and it contains many sentence splitting errors.} sentence pairs are often aligned using surface-level similarity metrics, such as Jaccard coefficient (Xu et al., 2015) or cosine distance of TF-IDF vectors (Paetzold et al., 2017), which fails to capture paraphrases and the context of surrounding sentences. A common drawback of text simplification models trained on such datasets is that they behave conservatively, performing mostly deletion, and rarely paraphrase (Alva-Manchego et al., 2017). Moreover, WIKILARGE is the concatenation of three early datasets (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011) that are extracted from Wikipedia dumps and are known to contain many errors (Xu et al., 2015).

To address these problems, we create the first high-quality manually annotated sentence-aligned datasets: NEWSELA-MANUAL with 50 article sets, and WIKI-MANUAL with 500 article pairs. We design a novel neural CRF alignment model, which utilizes fine-tuned BERT to measure semantic similarity and leverages the similar order of content be-
tween parallel documents, combined with an effective paragraph alignment algorithm. Experiments show that our proposed method outperforms all the previous monolingual sentence alignment approaches (Štajner et al., 2018; Paetzold et al., 2017; Xu et al., 2015) by more than 5 points in F1.

By applying our alignment model to all the 1,882 article sets in Newselea and 138,095 article pairs in Wikipedia dump, we then construct two new simplification datasets, NEWSLELA-AUTO (666,645 sentence pairs) and WIKI-AUTO (488,332 sentence pairs). Our new datasets with improved quantity and quality facilitate the training of complex seq2seq models. A BERT-initialized Transformer model trained on our datasets outperforms the state-of-the-art by 3.4% in terms of SARI, the main automatic metric for text simplification. Our simplification model produces 25% more rephrasing than those trained on the existing datasets. Our contributions include:

1. Two manually annotated datasets that enable the first systematic study for training and evaluating monolingual sentence alignment;
2. A neural CRF sentence alinger and a paragraph alignment algorithm that employ fine-tuned BERT to capture semantic similarity and take advantage of the sequential nature of parallel documents;
3. Two automatically constructed text simplification datasets which are of higher quality and 4.7 and 1.6 times larger than the existing datasets in their respective domains;
4. A BERT-initialized Transformer model for automatic text simplification, trained on our datasets, which establishes a new state-of-the-art in both automatic and human evaluation.

2 Neural CRF Sentence Aligner

We propose a neural CRF sentence alignment model, which leverages the similar order of content presented in parallel documents and captures editing operations across multiple sentences, such as splitting and elaboration (see Figure 1 for an example). To further improve the accuracy, we first align paragraphs based on semantic similarity and vicinity information, and then extract sentence pairs from these aligned paragraphs. In this section, we describe the task setup and our approach.

2.1 Problem Formulation

Given a simple article (or paragraph) $S$ of $m$ sentences and a complex article (or paragraph) $C$ of $n$ sentences, for each sentence $s_i$ ($i \in [1, m]$) in the simple article, we aim to find its corresponding sentence $c_{a_i}$ ($a_i \in [0, n]$) in the complex article. We use $a_i$ to denote the index of the aligned sentence, where $a_i = 0$ indicates that sentence $s_i$ is not aligned to any sentence in the complex article. The full alignment between article (or paragraph) pair $S$ and $C$ can then be represented by a sequence of alignment labels $a = (a_1, a_2, \ldots, a_m)$. Figure 1 shows an example of alignment labels. One specific aspect of our CRF model is that it uses a varied number of labels for each article (or paragraph) pair rather than a fixed set of labels.

2.2 Neural CRF Sentence Alignment Model

We learn $P(a|S, C)$, the conditional probability of alignment $a$ given an article pair $(S, C)$, using
\begin{align*}
P(a|S, C) &= \frac{\exp(\Psi(a, S, C))}{\sum_{a \in A} \exp(\Psi(a, S, C))} \\
&= \frac{\exp(\sum_{i=1}^{S} \psi(a_i, a_{i-1}, S, C))}{\sum_{a \in A} \exp(\sum_{i=1}^{S} \psi(a_i, a_{i-1}, S, C))} 
\end{align*}

where $|S| = m$ denotes the number of sentences in article $S$. The score $\sum_{i=1}^{S} \psi(a_i, a_{i-1}, S, C)$ sums over the sequence of alignment labels $a = (a_1, a_2, \ldots, a_m)$ between the simple article $S$ and the complex article $C$, and could be decomposed into two factors as follows:

\[\psi(a_i, a_{i-1}, S, C) = \text{sim}(s_i, c_{a_i}) + T(a_i, a_{i-1})\]

where $\text{sim}(s_i, c_{a_i})$ is the semantic similarity score between the two sentences, and $T(a_i, a_{i-1})$ is a pairwise score for alignment label transition that $a_i$ follows $a_{i-1}$.

### Semantic Similarity

A fundamental problem in sentence alignment is to measure the semantic similarity between two sentences $s_i$ and $c_j$. Prior work used lexical similarity measures, such as Jaccard similarity (Xu et al., 2015), TF-IDF (Paetzold et al., 2017), and continuous n-gram features (Štajner et al., 2018). In this paper, we fine-tune BERT (Devlin et al., 2019) on our manually labeled dataset (details in §3) to capture semantic similarity.

### Alignment Label Transition

In parallel documents, the contents of the articles are often presented in a similar order. The complex sentence $c_{a_i}$ that is aligned to $s_i$ is often related to the complex sentences $c_{a_{i-1}}$ and $c_{a_{i+1}}$, which are aligned to $s_{i-1}$ and $s_{i+1}$, respectively. To incorporate this intuition, we propose a scoring function to model the transition between alignment labels using the following features:

\[
\begin{align*}
g_1 &= |a_i - a_{i-1}| \\
g_2 &= \mathbb{1}(a_i = 0, a_{i-1} \neq 0) \\
g_3 &= \mathbb{1}(a_i \neq 0, a_{i-1} = 0) \\
g_4 &= \mathbb{1}(a_i = 0, a_{i-1} = 0)
\end{align*}
\]

where $g_1$ is the absolute distance between $a_i$ and $a_{i-1}$, $g_2$ and $g_3$ denote if the current or prior sentence is not aligned to any sentence, and $g_4$ indicates whether both $s_i$ and $s_{i-1}$ are not aligned to any sentences. The score is computed as follows:

\[T(a_i, a_{i-1}) = \text{FFNN}([g_1, g_2, g_3, g_4])\]

where $[,]$ represents concatenation operation and FFNN is a 2-layer feedforward neural network. We provide more implementation details of the model in Appendix A.1.

### 2.3 Inference and Learning

During inference, we find the optimal alignment $\hat{a}$:

\[\hat{a} = \arg\max_a P(a|S, C)\]

using Viterbi algorithm in $O(mn^2)$ time. During training, we maximize the conditional probability of the gold alignment label $a^*$:

\[\log P(a^*|S, C) = \Psi(a^*, S, C) - \log \sum_{a \in A} \exp(\Psi(a, S, C))\]

The second term sums the scores of all possible alignments and can be computed using forward algorithm in $O(mn^2)$ time as well.

### 2.4 Paragraph Alignment

Both accuracy and computing efficiency can be improved if we align paragraphs before aligning sentences. In fact, our empirical analysis revealed that sentence-level alignments mostly reside within the corresponding aligned paragraphs (details in §4.4 and Table 3). Moreover, aligning paragraphs first provides more training instances and reduces the label space for our neural CRF model.

We propose Algorithm 1 and 2 for paragraph alignment. Given a simple article $S$ with $k$ paragraphs $S = (S_1, S_2, \ldots, S_k)$ and a complex article $C$ with $l$ paragraphs $C = (C_1, C_2, \ldots, C_l)$, we first apply Algorithm 1 to calculate the semantic similarity matrix $\text{sim}P$ between paragraphs by averaging or maximizing over the sentence-level similarities (§2.2). Then, we use Algorithm 2 to generate the paragraph alignment matrix $\text{align}P$.

We align paragraph pairs if they satisfy one of the two conditions: (a) having high semantic similarity and appearing in similar positions in the article pair (e.g., both at the beginning), or (b) two continuous paragraphs in the complex article having relatively high semantic similarity with one paragraph in the simple side, (e.g., paragraph splitting or fusion). The difference of relative position in documents
Algorithm 1: Pairwise Paragraph Similarity

Initialize: \( simP \in \mathbb{R}^{2 \times k \times l} \) to \( 0^{2 \times k \times l} \)

for \( i \leftarrow 1 \) to \( k \) do

for \( j \leftarrow 1 \) to \( l \) do

\[
simP[1, i, j] = \frac{\max_{s_p \in S_i, c_q \in C_j} \text{simSent}(s_p, c_q)}{\sum_{c_q \in C_j} \text{simSent}(s_p, c_q)}
\]

\[
simP[2, i, j] = \frac{\max_{s_p \in S_i, c_q \in C_j} \text{simSent}(s_p, c_q)}{\sum_{c_q \in C_j} \text{simSent}(s_p, c_q)}
\]

end

end

return \( simP \)

Algorithm 2: Paragraph Alignment Algorithm

Input: \( simP \in \mathbb{R}^{2 \times k \times l} \) to \( 0^{k \times l} \)

for \( i \leftarrow 1 \) to \( k \) do

\[
j_{max} = \arg \max_{j} \simP[1, i, j]
\]

if \( \simP[1, i, j_{max}] > \tau_1 \) and \( d(i, j_{max}) < \tau_2 \) then

\[
\text{alignP}[i, j_{max}] = 1
\]

end

for \( j \leftarrow 1 \) to \( l \) do

if \( \simP[2, i, j] > \tau_3 \) then

\[
\text{alignP}[i, j] = 1
\]

end

if \( j > 1 \) & \( \simP[2, i, j] > \tau_4 \) & \( \simP[2, i, j - 1] > \tau_4 \) & \( d(i, j - 1) < \tau_5 \) & \( d(i, j) < \tau_5 \) then

\[
\text{alignP}[i, j] = 1
\]

end

\[
\text{alignP}[i, j - 1] = 1
\]

end

end

return \( \text{alignP} \)

is defined as \( d(i, j) = \frac{|x_i - \frac{1}{x_i}|}{2} \), and the thresholds \( \tau_1 - \tau_5 \) in Algorithm 2 are selected using the dev set. Finally, we merge the neighbouring paragraphs which are aligned to the same paragraph in the simple article before feeding them into our neural CRF aligner. We provide more details in Appendix A.1.

3 Constructing Alignment Datasets

To address the lack of reliable sentence alignment for Newsela (Xu et al., 2015) and Wikipedia (Zhu et al., 2010; Woodsend and Lapata, 2011), we designed an efficient annotation methodology to first manually align sentences between a few complex and simple article pairs. Then, we automatically aligned the rest using our alignment model trained on the human annotated data. We created two sentence-aligned parallel corpora (details in §5), which are the largest to date for text simplification.

3.1 Sentence Aligned Newsela Corpus

Newsela corpus (Xu et al., 2015) consists of 1,932 English news articles where each article (level 0) is re-written by professional editors into four simpler versions at different readability levels (level 1-4). We annotate sentence alignments for article pairs at adjacent readability levels (e.g., 0-1, 1-2) as the alignments between non-adjacent levels (e.g., 0-2) can be then derived automatically. To ensure efficiency and quality, we designed the following three-step annotation procedure:

1. Align paragraphs using CATS toolkit (Štajner et al., 2018), and then correct the automatic paragraph alignment errors by two in-house annotators.3 Performing paragraph alignment as the first step significantly reduces the number of sentence pairs to be annotated from every possible sentence pair to the ones within the aligned paragraphs. We design an efficient visualization toolkit for this step, for which a screenshot can be found in Appendix E.2.

2. For each sentence pair within the aligned paragraphs, we ask five annotators on the Figure

\[\text{Article level}\]

<table>
<thead>
<tr>
<th></th>
<th>Newsela-Manual</th>
<th>Newsela-Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td># of original articles</td>
<td>50</td>
<td>1,882</td>
</tr>
<tr>
<td># of article pairs</td>
<td>500</td>
<td>18,820</td>
</tr>
</tbody>
</table>

\[\text{Sentence level}\]

<table>
<thead>
<tr>
<th></th>
<th>Newsela-Manual</th>
<th>Newsela-Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td># of original sent. (level 0)</td>
<td>2,190</td>
<td>59,752</td>
</tr>
<tr>
<td># of sentence pairs</td>
<td>1.01M†</td>
<td>666,645</td>
</tr>
<tr>
<td># of unique complex sent.</td>
<td>7,001</td>
<td>195,566</td>
</tr>
<tr>
<td># of unique simple sent.</td>
<td>8,008</td>
<td>246,420</td>
</tr>
<tr>
<td>avg. length of simple sent.</td>
<td>13.9</td>
<td>14.8</td>
</tr>
<tr>
<td>avg. length of complex sent.</td>
<td>21.3</td>
<td>24.9</td>
</tr>
</tbody>
</table>

\[\text{Labels of sentence pairs}\]

<table>
<thead>
<tr>
<th></th>
<th>Newsela-Manual</th>
<th>Newsela-Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td># of aligned (not identical)</td>
<td>5,182</td>
<td>666,645</td>
</tr>
<tr>
<td># of partially-aligned</td>
<td>14,023</td>
<td></td>
</tr>
<tr>
<td># of not-aligned</td>
<td>0.99M</td>
<td></td>
</tr>
</tbody>
</table>

\[\text{Text simplification phenomenon}\]

<table>
<thead>
<tr>
<th></th>
<th>Newsela-Manual</th>
<th>Newsela-Auto</th>
</tr>
</thead>
<tbody>
<tr>
<td># of sent. rephrasing (1-to-1)</td>
<td>8,216</td>
<td>307,450</td>
</tr>
<tr>
<td># of sent. copying (1-to-1)</td>
<td>3,842</td>
<td>147,327</td>
</tr>
<tr>
<td># of sent. splitting (1-to-n)</td>
<td>4,237</td>
<td>160,300</td>
</tr>
<tr>
<td># of sent. merging (n-to-1)</td>
<td>232</td>
<td></td>
</tr>
<tr>
<td># of sent. fusion (m-to-n)</td>
<td>252</td>
<td></td>
</tr>
<tr>
<td># of sent. deletion (1-to-0)</td>
<td>6,247</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Statistics of our manually and automatically created sentence alignment annotations on Newsela. † This number includes all complex-simple sentence pairs (including aligned, partially-aligned, or not-aligned) across all 10 combinations of 5 readability levels (level 0-4), of which 20,343 sentence pairs between adjacent readability levels were manually annotated and the rest of labels were derived.
3.2 Sentence Aligned Wikipedia Corpus

We also create a new version of Wikipedia corpus by aligning sentences between English Wikipedia and Simple English Wikipedia. Previous work (Xu et al., 2015) has shown that Wikipedia is much noisier than the Newsela corpus. We provide this dataset in addition to facilitate future research.

We first extract article pairs from English and Simple English Wikipedia by leveraging Wikidata, a well-maintained database that indexes named entities (and events etc.) and their Wikipedia pages in different languages. We found this method to be more reliable than using page titles (Coster and Kauchak, 2011) or cross-lingual links (Zhu et al., 2010; Woodsend and Lapata, 2011), as titles can be ambiguous and cross-lingual links may direct to a disambiguation or mismatched page (more details in Appendix B). In total, we extracted 138,095 article pairs from the 2019/09 Wikipedia dump, which is two times larger than the previous datasets (Coster and Kauchak, 2011; Zhu et al., 2010) of only 60∼65k article pairs, using an improved version of the WikiExtractor library.

Then, we crowdsourced the sentence alignment annotations for 500 randomly sampled document pairs (10,123 sentence pairs total). As document length in English and Simple English Wikipedia articles vary greatly, we designed the following annotation strategy that is slightly different from Newsela. For each sentence in the simple article, we select the sentences with the highest similarity scores from the complex article for manual annotation, based on four similarity measures: lexical similarity from CATS (ˇStajner et al., 2018), cosine similarity using TF-IDF (Paetzold et al., 2017), cosine similarity between BERT sentence embeddings, and alignment probability by a BERT model fine-tuned on our NEWSELA-MANUAL data (§3.1).

As these four metrics may rank the same sentence at the top, on an average, we collected 2.13 complex sentences for every simple sentence and annotated the alignment label for each sentence pair. Our pilot study showed that this method captured 93.6% of the aligned sentence pairs. We named this manually labeled dataset WIKI-MANUAL with a train/dev/test split of 350/50/100 article pairs.

Finally, we trained our alignment model on this dataset.
Table 2: Performance of different sentence alignment methods on the NEWSELA-MANUAL test set. † Previous work was designed only for Task 1 and used alignment strategy (greedy algorithm or dynamic programming) to improve either precision or recall.

<table>
<thead>
<tr>
<th>Similarity-based models</th>
<th>Task 1 (aligned &amp; partial vs. others)</th>
<th>Task 2 (aligned vs. others)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>Jaccard (Xu et al., 2015)</td>
<td>94.93</td>
<td>76.69</td>
</tr>
<tr>
<td>TF-IDF (Paetzold et al., 2017)</td>
<td>96.24</td>
<td>83.05</td>
</tr>
<tr>
<td>LR (Štajner et al., 2018)</td>
<td>93.11</td>
<td>84.96</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Similarity-based models w/ alignment strategy (previous SOTA)</th>
<th>Task 1 (aligned &amp; partial vs. others)</th>
<th>Task 2 (aligned vs. others)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>JaccardAlign (Xu et al., 2015)</td>
<td>98.66</td>
<td>67.58</td>
</tr>
<tr>
<td>MASSAlign (Paetzold et al., 2017)</td>
<td>95.49</td>
<td>82.27</td>
</tr>
<tr>
<td>CATS (Štajner et al., 2018)</td>
<td>88.56</td>
<td>91.31</td>
</tr>
<tr>
<td>Our CRF Aligner</td>
<td>97.86</td>
<td>93.43</td>
</tr>
</tbody>
</table>

Table 3: Ablation study of our aligner on dev set.

Table: | Task 1 P R F1 | Task 2 P R F1 |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Neural sentence pair models</td>
<td></td>
</tr>
<tr>
<td>Interest</td>
<td>92.8</td>
</tr>
<tr>
<td>ESIM</td>
<td>91.5</td>
</tr>
<tr>
<td>BERTScore</td>
<td>90.6</td>
</tr>
<tr>
<td>BERTembedding</td>
<td>84.7</td>
</tr>
<tr>
<td>BERTsim</td>
<td>93.3</td>
</tr>
<tr>
<td>+ ParaAlign</td>
<td>98.4</td>
</tr>
</tbody>
</table>

Neural CRF aligner

| Our CRF Aligner | 96.5 | 90.1 | 93.2 | 88.6 | 87.7 | 88.1 |
| + gold ParaAlign | 97.3 | 91.1 | 94.1 | 88.9 | 88.0 | 88.4 |

4 Evaluation of Sentence Alignment

In this section, we present experiments that compare our neural sentence alignment against the state-of-the-art approaches on NEWSELA-MANUAL (§3.1) and WIKI-MANUAL (§3.2) datasets.

4.1 Existing Methods

We compare our neural CRF aligner with the following baselines and state-of-the-art approaches:

1. Three similarity-based methods: **Jaccard similarity** (Xu et al., 2015), **TF-IDF cosine similarity** (Paetzold et al., 2017) and a **logistic regression classifier** trained on our data with lexical features from Štajner et al. (2018).
2. **JaccardAlign** (Xu et al., 2015), which uses Jaccard coefficient for sentence similarity and a greedy approach for alignment.
3. **MASSAlign** (Paetzold et al., 2017), which combines TF-IDF cosine similarity with a vicinity-driven dynamic programming algorithm for alignment.

4. **CATS** toolkit (Štajner et al., 2018), which uses character n-gram features for sentence similarity and a greedy alignment algorithm.

4.2 Evaluation Metrics

We report **Precision**, **Recall** and **F1** on two binary classification tasks: **aligned + partially-aligned vs. not-aligned** (Task 1) and **aligned vs. partially-aligned + not-aligned** (Task 2). It should be noted that we excluded identical sentence pairs in the evaluation as they are trivial to classify.

4.3 Results

Table 2 shows the results on NEWSELA-MANUAL test set. For similarity-based methods, we choose a threshold based on the maximum F1 on the dev set. Our neural CRF aligner outperforms the state-of-the-art approaches by more than 5 points in F1. In particular, our method performs better than the previous work on partial alignments, which contain many interesting simplification operations, such as sentence splitting and paraphrasing with deletion.

Similarly, our CRF alignment model achieves 85.1 F1 for Task 1 (aligned + partially-aligned vs. not-aligned) on the WIKI-MANUAL test set. It outperforms one of the previous SOTA approaches CATS (Štajner et al., 2018) by 15.1 points in F1. We provide more details in Appendix C.

4.4 Ablation Study

We analyze the design choices crucial for the good performance of our alignment model, namely CRF component, the paragraph alignment and the BERT-based semantic similarity measure. Table 3 shows the importance of each component with a series of ablation experiments on the dev set.
We establish a new state-of-the-art for sentence simplification compared to the old datasets (Xu et al., 2015; Zhang and Lapata, 2017).

CRF Model Our aligner achieves 93.2 F1 and 88.1 F1 on Task 1 and 2, respectively, which is around 3 points higher than its variant without the CRF component (BERT finetune + ParaAlign). Modeling alignment label transitions and sequential predictions helps our neural CRF aligner to handle sentence splitting cases better, especially when sentences undergo dramatic rewriting.

Paragraph Alignment Adding paragraph alignment (BERT finetune + ParaAlign) improves the precision on Task 1 from 93.3 to 98.4 with a negligible decrease in recall when compared to not aligning paragraphs (BERT finetune). Moreover, paragraph alignments generated by our algorithm (Our Aligner) perform close to the gold alignments (Our Aligner + gold ParaAlign) with only 0.9 and 0.3 difference in F1 on Task 1 and 2, respectively.

Semantic Similarity BERT finetune performs better than other neural models, including InferSent (Conneau et al., 2017), ESIM (Chen et al., 2017), BERTScore (Zhang et al., 2020) and pre-trained BERT embedding (Devlin et al., 2019). For BERTScore, we use idf weighting, and treat simple sentence as reference.

5 Experiments on Automatic Sentence Simplification

In this section, we compare different automatic text simplification models trained on our new parallel corpora, NEWSLEA-AUTO and WIKI-AUTO, with their counterparts trained on the existing datasets. We establish a new state-of-the-art for sentence simplification by training a Transformer model with initialization from pre-trained BERT checkpoints.

5.1 Comparison with existing datasets

Existing datasets of complex-simple sentences, NEWSLEA (Xu et al., 2015) and WikiLARGE (Zhang and Lapata, 2017), were aligned using lexical similarity metrics. NEWSLEA dataset (Xu et al., 2015) was aligned using JaccardAlign (§4.1). WikiLARGE is a concatenation of three early datasets (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011) where sentences in Simple/Normal English Wikipedia and editing history were aligned by TF-IDF cosine similarity.

For our new NEWSLEA-AUTO, we partitioned the article sets such that there is no overlap between the new train set and the old test set, and vice-versa. Following Zhang and Lapata (2017), we also excluded sentence pairs corresponding to the levels 0–1, 1–2 and 2–3. For our WIKI-AUTO dataset, we eliminated sentence pairs with high (>0.9) or low (<0.1) lexical overlap based on BLEU scores (Papineni et al., 2002), following Štajner et al. (2015). We observed that sentence pairs with low BLEU are often inaccurate paraphrases with only shared named entities and the pairs with high BLEU are dominated by sentences merely copied without simplification. We used the benchmark TURK corpus (Xu et al., 2016) for evaluation on Wikipedia, which consists of 8 human-written references for sentences in the validation and test sets. We discarded sentences in TURK corpus from WIKI-AUTO. Table 4 shows the statistics of the existing and our new datasets.

<table>
<thead>
<tr>
<th></th>
<th>NEWSLEA</th>
<th>WIKI-AUTO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Auto</td>
<td>Old</td>
</tr>
<tr>
<td># of article pairs</td>
<td>13k</td>
<td>7.9k</td>
</tr>
<tr>
<td># of sent. pairs (train)</td>
<td>394k</td>
<td>94k</td>
</tr>
<tr>
<td># of sent. pairs (dev)</td>
<td>43k</td>
<td>1.1k</td>
</tr>
<tr>
<td># of sent. pairs (test)</td>
<td>44k</td>
<td>1k</td>
</tr>
<tr>
<td>avg. sent. len (complex)</td>
<td>25.4</td>
<td>25.8</td>
</tr>
<tr>
<td>avg. sent. len (simple)</td>
<td>13.8</td>
<td>15.7</td>
</tr>
</tbody>
</table>

Table 4: Statistics of our newly constructed parallel corpora for sentence simplification compared to the old datasets (Xu et al., 2015; Zhang and Lapata, 2017).

5.2 Baselines and Simplification Models

We compare the following seq2seq models trained using our new datasets versus the existing datasets:

1. A BERT-initialized Transformer, where the encoder and decoder follow the BERT base architecture. The encoder is initialized with the same checkpoint and the decoder is randomly initialized (Rothe et al., 2020).
2. A randomly initialized Transformer with the same BERT base architecture as above.
4. EditNTS (Dong et al., 2019), a state-of-the-art neural programmer-interpreter (Reed and de Freitas, 2016) approach that predicts explicit edit operations sequentially.

In addition, we compared our BERT-initialized Transformer model with the released system outputs from Kriz et al. (2019) and EditNTS (Dong et al., 2019). We implemented our LSTM and Transformer models using Fairseq. We provide the model and training details in Appendix D.1.
5.3 Results

In this section, we evaluate different simplification models trained on our new datasets versus on the old existing datasets using both automatic and human evaluation.

5.3.1 Automatic Evaluation

We report SARI (Xu et al., 2016), Flesch-Kincaid (FK) grade level readability (Kincaid and Chissom, 1975), and average sentence length (Len). While SARI compares the generated sentence to a set of reference sentences in terms of correctly inserted, kept and deleted n-grams ($n \in \{1, 2, 3, 4\}$), FK measures the readability of the generated sentence. We also report the three rewrite operation scores used in SARI: the precision of delete (del), the F1-scores of add (add), and keep (keep) operations.

Tables 5 and 8 show the results on Newsela and Wikipedia datasets respectively. Systems trained on our datasets outperform their equivalents trained on the existing datasets according to SARI. The difference is notable for Transformer$_{bert}$ with a 6.4% and 3.7% increase in SARI on NEWSELA-AUTO test set and TURK corpus, respectively. Larger size and improved quality of our datasets enable the training of complex Transformer models. In fact, Transformer$_{bert}$ trained on our new datasets outperforms the existing state-of-the-art systems for automatic text simplification. Although improvement in SARI is modest for LSTM-based models (LSTM and EditNTS), the increase in F1 scores for addition and deletion operations indicate that the models trained on our datasets make more meaningful changes to the input sentence.

5.3.2 Human Evaluation

We also performed human evaluation by asking five Amazon Mechanical Turk workers to rate fluency, adequacy and simplicity (detailed instructions in Appendix D.2) of 100 random sentences generated by different simplification models trained on NEWSELA-AUTO and the existing dataset. Each
A worker evaluated these aspects on a 5-point Likert scale. We averaged the ratings from five workers. Table 7 demonstrates that Transformer_{bert} trained on NEWSELA-AUTO greatly outperforms the one trained on the old dataset. Even with shorter sentence outputs, our Transformer_{bert} retained similar adequacy as the LSTM-based models. Our Transformer_{bert} model also achieves better fluency, adequacy, and overall ratings compared to the SOTA systems (Table 6). We provide examples of system outputs in Appendix D.3. Our manual inspection (Figure 3) also shows that Transformer_{bert} trained on NEWSELA-AUTO performs 25% more paraphrasing and deletions than its variant trained on the previous NEWSELA (Xu et al., 2015) dataset.

6 Related Work

Text simplification is considered as a text-to-text generation task where the system learns how to simplify from complex-simple sentence pairs. There is a long line of research using methods based on hand-crafted rules (Siddharthan, 2006; Niklaus et al., 2019), statistical machine translation (Narayan and Gardent, 2014; Xu et al., 2016; Wubben et al., 2012), or neural seq2seq models (Zhang and Lapata, 2017; Zhao et al., 2018; Nisioi et al., 2017). As the existing datasets were built using lexical similarity metrics, they frequently omit paraphrases and sentence splits. While training on such datasets creates conservative systems that rarely paraphrase, evaluation on these datasets exhibits an unfair preference for deletion-based simplification over paraphrasing.

Sentence alignment has been widely used to extract complex-simple sentence pairs from parallel articles for training text simplification systems. Previous work used surface-level similarity metrics, such as TF-IDF cosine similarity (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011; Paetzold et al., 2017), Jaccard-similarity (Xu et al., 2015), and other lexical features (Hwang et al., 2015; Štajner et al., 2018). Then, a greedy (Štajner et al., 2018) or dynamic programming (Barzilay and Elhadad, 2003; Paetzold et al., 2017) algorithm was used to search for the optimal alignment. Another related line of research (Smith et al., 2010; Tufis et al., 2013; Tsai and Roth, 2016; Gottschalk and Demidova, 2017; Aghaebrahimian, 2018; Thompson and Koehn, 2019) aligns parallel sentences in bilingual corpora for machine translation.

7 Conclusion

In this paper, we proposed a novel neural CRF model for sentence alignment, which substantially outperformed the existing approaches. We created two high-quality manually annotated datasets (NEWSELA-MANUAL and WIKI-MANUAL) for training and evaluation. Using the neural CRF sentence aligner, we constructed two largest sentence-aligned datasets to date (NEWSELA-AUTO and WIKI-AUTO) for text simplification. We showed that a BERT-initialzed Transformer trained on our new datasets establishes new state-of-the-art performance for automatic sentence simplification.

Acknowledgments

We thank three anonymous reviewers for their helpful comments, Newsela for sharing the data, Ohio Supercomputer Center (Center, 2012) and NVIDIA for providing GPU computing resources. We also thank Sarah Flanagan, Bohan Zhang, Raleigh Potluri, and Alex Wing for help with data annotation. This research is supported in part by the NSF awards IIS-1755898 and IIS-1822754, ODNI and IARPA via the BETTER program contract 19051600004, ARO and DARPA via the Social-Sim program contract W911NF-17-C-0095, Figure Eight AI for Everyone Award, and Criteo Faculty Research Award to Wei Xu. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of NSF, ODNI, IARPA, ARO, DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.
References


A Neural CRF Alignment Model

A.1 Implementation Details

We used PyTorch\(^{10}\) to implement our neural CRF alignment model. For the sentence encoder, we used Huggingface implementation (Wolf et al., 2019) of BERT\(_{base}\)\(^{11}\) architecture with 12 layers of Transformers. When fine-tuning the BERT model, we use the representation of [CLS] token for classification. We use cross entropy loss and update the weights in all layers. Table 9 summarizes the hyperparameters of our model. Table 10 provides the thresholds for our paragraph alignment Algorithm 2, which were chosen based on Newsela-Manual dev data.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hidden units</td>
<td>768</td>
</tr>
<tr>
<td>learning rate</td>
<td>0.00002</td>
</tr>
<tr>
<td>max sequence length</td>
<td>128</td>
</tr>
<tr>
<td># of layers</td>
<td>12</td>
</tr>
<tr>
<td># of heads</td>
<td>12</td>
</tr>
<tr>
<td>batch size</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 9: Parameters of our neural CRF sentence alignment model.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tau_1)</td>
<td>0.1</td>
</tr>
<tr>
<td>(\tau_2)</td>
<td>0.34</td>
</tr>
<tr>
<td>(\tau_3)</td>
<td>0.9998861788416304</td>
</tr>
<tr>
<td>(\tau_4)</td>
<td>0.999891818299745</td>
</tr>
<tr>
<td>(\tau_5)</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 10: The thresholds in paragraph alignment Algorithm 2 for Newsela data.

For Wikipedia data, we tailored our paragraph alignment algorithm (Algorithm 3 and 4). Table 11 provides the thresholds for Algorithm 4, which were chosen based on WIKI-MANUAL dev data.

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\tau_1)</td>
<td>0.991775706637882</td>
</tr>
<tr>
<td>(\tau_2)</td>
<td>0.8</td>
</tr>
<tr>
<td>(\tau_3)</td>
<td>0.5</td>
</tr>
<tr>
<td>(\tau_4)</td>
<td>5</td>
</tr>
<tr>
<td>(\tau_5)</td>
<td>0.9958</td>
</tr>
</tbody>
</table>

Table 11: The thresholds in paragraph alignment Algorithm 4 for Wikipedia data.

B Sentence Aligned Wikipedia Corpus

We present more details about our pre-processing steps for creating the WIKI-MANUAL and WIKIAUTO corpora here. In Wikipedia, Simple English is considered as a language by itself. When extracting articles from Wikipedia dump, we removed the meta-page and disambiguation pages. We also removed sentences with less than 4 tokens and sentences that end with a colon.

After the pre-processing and matching steps, there are 13,036 article pairs in which the simple article contains only one sentence. In most cases, that one sentence is aligned to the first sentence in the complex article. However, we find that the patterns of these sentence pairs are very repetitive (e.g., XXX is a city in XXX. XXX is a football player in XXX.). Therefore, we use regular expressions to filter out the sentences with repetitive patterns. Then, we use a BERT model fine-tuned on the WIKI-MANUAL dataset to compute the semantic similarity of each sentence pair and keep the ones with a similarity larger than a threshold.

---

\(^{10}\)https://pytorch.org/

\(^{11}\)https://github.com/google-research/bert
tuned on the dev set. After filtering, we ended up with 970 aligned sentence pairs in total from these 13,036 article pairs.

C Sentence Alignment on Wikipedia

In this section, we compare different approaches for sentence alignment on the Wiki-Manual dataset. Tables 12 and 13 report the performance for Task 1 (aligned + partially-aligned vs. not-aligned) on dev and test set. To generate prediction for MASSAlign, CATS and two BERT_finetune methods, we first utilize the method in §3.2 to select candidate sentence pairs, as we found this step helps to improve their accuracy. Then we apply the similarity metric from each model to calculate the similarity of each candidate sentence pair. We tune a threshold for max f1 on the dev set and apply it to the test set. Candidate sentence pairs with a similarity larger than the threshold will be predicted as aligned, otherwise not-aligned. Sentence pairs that are not selected as candidates will also be predicted as not-aligned.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hidden units</td>
<td>768</td>
<td>batch size</td>
<td>32</td>
</tr>
<tr>
<td>filter size</td>
<td>3072</td>
<td>max len</td>
<td>100</td>
</tr>
<tr>
<td># of layers</td>
<td>12</td>
<td>activation</td>
<td>GELU</td>
</tr>
<tr>
<td>attention heads</td>
<td>12</td>
<td>dropout</td>
<td>0.1</td>
</tr>
<tr>
<td>loss</td>
<td>CE</td>
<td>seed</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 14: Parameters of our Transformer model.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>hidden units</td>
<td>256</td>
<td>batch size</td>
<td>64</td>
</tr>
<tr>
<td>embedding dim</td>
<td>300</td>
<td>max len</td>
<td>100</td>
</tr>
<tr>
<td># of layers</td>
<td>2</td>
<td>dropout</td>
<td>0.2</td>
</tr>
<tr>
<td>lr</td>
<td>0.001</td>
<td>optimizer</td>
<td>Adam</td>
</tr>
<tr>
<td>clipping</td>
<td>5</td>
<td>epochs</td>
<td>30</td>
</tr>
<tr>
<td>min vocab freq</td>
<td>3</td>
<td>seed</td>
<td>13</td>
</tr>
</tbody>
</table>

Table 15: Parameters of our LSTM model.

architecture for both encoder and decoder. We initialized the encoder using BERT_base uncased checkpoint. Rothe et al. (2020) used a similar model for sentence fusion and summarization. We trained each model using Adam optimizer with a learning rate of 0.0001, linear learning rate warmup of 40k steps and 200k training steps. We tokenized the data with BERT WordPiece tokenizer. Table 14 shows the values of other hyperparameters.

For the LSTM baseline, we replicated the LSTM encoder-decoder model used by Zhang and Lapata (2017). We preprocessed the data by replacing the named entities in a sentence using spaCy14 toolkit. We also replaced all the words with frequency less than three with <UNK>. If our model predicted <UNK>, we replaced it with the aligned source word (Jean et al., 2015). Table 15 summarizes the hyperparameters of LSTM model. We used 300-dimensional GloVe word embeddings (Pennington et al., 2014) to initialize the embedding layer.

D Sentence Simplification

D.1 Implementation Details

We used Fairseq12 toolkit to implement our Transformer (Vaswani et al., 2017) and LSTM (Hochreiter and Schmidhuber, 1997) baselines. For the Transformer baseline, we followed BERTbase 13

12https://github.com/pytorch/fairseq
13https://github.com/google-research/bert
14https://spacy.io/
D.2 Human Evaluation

For this task you are given one source sentence and five (5) simplifications of the original sentence generated by different computer programs. The goal is to judge whether each simplified sentence

- is grammatically correct i.e. whether it is well-formed
- is simpler than the original source sentence.
- preserves meaning of the original sentence.

You will do this using a 1-5 rating scale, where 5 is best and 1 is worst. There are no “correct” answers and whatever choice is appropriate for you is a valid response. For example, if you are given the following complex sentence and simplifications:

Original sentence:
Financial markets had anticipated Portugal’s need for assistance as its costs of financing had risen to unsustainable levels, and investors generally shrugged off the news on Thursday.

<table>
<thead>
<tr>
<th>Simplification</th>
<th>Meaning</th>
<th>Grammar</th>
<th>Simplicity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Financial markets had expected Portugal’s need for help because costs had become unsustainable and investors dismissed the news on Thursday.</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>2. Financial markets had expected Portugal’s need for help as its costs of financing had risen to unsustainable levels, and investors generally shrugged off the news on Thursday.</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
<tr>
<td>3. Financial markets the need for assistance had anticipated, costs of financing unsustainable shrugged off the news Thursday.</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4. Financial markets had anticipated Portugal’s need for assistance.</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>5. Financial markets dismissed the news on Thursday.</td>
<td>1</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

Sentence (1) gets a high rating with respect to simplicity since the long and complex sentence had been simplified considerably. Few words (e.g., generally, of financing) have been dropped, whereas others have been substituted with what are more familiar ones (e.g. anticipated). It also gets high rating with respect to grammar and meaning because it is grammatically correct and preserves most of the meaning of the original. Sentence (2) also rates high in terms of grammar and meaning. However, it is not as simple as sentence (1) although some unfamiliar words have been substituted with simpler alternatives. Therefore, it gets a modest simplicity rating. Simplified sentence (3) makes little sense and is rather difficult to read. Therefore, it gets a low rating for grammar, simplicity and meaning. Simplified sentence (4) is fluent and easier to understand. So, it gets high rating in terms of grammar and simplicity. Although it is simpler than the original, it has retained a large part of the sentence content. Simplifications that drastically change the meaning of the original sentence should be rated low in terms of meaning. Simplified sentence (5) changes the meaning but is easier to understand and well-formed. So, its gets low rating for meaning and high rating for simplicity and grammar. Simplifications that are grammatically correct should be rated high in terms of grammar even though they change the meaning of the original sentence.

In some cases, the computer program will choose not to change the original sentence at all. In such cases, try to think if you could make the sentence simpler. If this is the case then you should probably rate the computer-generated sentence low in terms of simplicity. Otherwise you can give high rating.

These sentences have been preprocessed by converting all letters to lowercase, separating punctuation, and splitting conjunctions. Please ignore this in your work and do not allow it to affect your judgments.

Figure 4: Instructions provided to Amazon Mechanical Turk workers to evaluate generated simplified sentences. We used the same instructions as described in Kriz et al. (2019).
### D.3 Example System Outputs

<table>
<thead>
<tr>
<th><strong>Generated by LSTM baseline</strong></th>
<th><strong>Examples</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex (input)</td>
<td>In Seattle, eight activists between ages 10 and 15 petitioned Washington state last year to adopt stricter science-based regulations to protect them against climate change.</td>
</tr>
<tr>
<td>Simple (reference)</td>
<td>In Seattle, eight youths between 10 to 15 years old petitioned the state of Washington to change the law.</td>
</tr>
<tr>
<td>New (this work)</td>
<td>in seattle, eight activists between ages 10 and 15 asked Washington state last year to keep the environment safe. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Old (Xu et al., 2015)</td>
<td>in seattle, eight activists between ages 10 and 15 asked Washington state last year to adopt stricter science-based rules to protect them against climate change. (Lexical Paraphrase)</td>
</tr>
<tr>
<td>Complex (input)</td>
<td>He recognized that another recommendation would be controversial with police groups: independent investigations after police shootings.</td>
</tr>
<tr>
<td>Simple (reference)</td>
<td>He admitted that police would not like one of the recommendations.</td>
</tr>
<tr>
<td>New (this work)</td>
<td>he thought another suggestion would be against the police. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Old (Xu et al., 2015)</td>
<td>he recognized that another suggestion would be controversial with police groups. (Lexical Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Complex (input)</td>
<td>The Philadelphia Museum of Art has two famous selfie spots, both from the movie &quot;Rocky.&quot;</td>
</tr>
<tr>
<td>Simple (reference)</td>
<td>The Philadelphia Museum of Art has two big selfie spots.</td>
</tr>
<tr>
<td>New (this work)</td>
<td>the philadelphia museum of art has two picture spots. (Lexical Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Old (Xu et al., 2015)</td>
<td>the philadelphia museum of art has two famous spots. (Deletion)</td>
</tr>
<tr>
<td><strong>Generated by Transformerbert</strong></td>
<td><strong>Examples</strong></td>
</tr>
<tr>
<td>Complex (input)</td>
<td>Some Chicago residents got angry about it.</td>
</tr>
<tr>
<td>Simple (reference)</td>
<td>The plan made some people angry.</td>
</tr>
<tr>
<td>New (this work)</td>
<td>some people in chicago were angry. (Phrasal Paraphrase)</td>
</tr>
<tr>
<td>Old (Xu et al., 2015)</td>
<td>some chicago residents got angry. (Deletion)</td>
</tr>
<tr>
<td>Complex (input)</td>
<td>Emissions standards have been tightened, and the government is investing money in solar, wind and other renewable energy.</td>
</tr>
<tr>
<td>Simple (reference)</td>
<td>China has also put a great deal of money into solar, wind and other renewable energy.</td>
</tr>
<tr>
<td>New (this work)</td>
<td>the government is putting aside money for new types of energy. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Old (Xu et al., 2015)</td>
<td>the government is investing in money, wind and other equipment. (Lexical Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Complex (input)</td>
<td>On Feb. 9, 1864, he was sitting for several portraits, including the one used for the $5 bill.</td>
</tr>
<tr>
<td>Simple (reference)</td>
<td>On Feb. 9, 1864, several artists painted pictures of him.</td>
</tr>
<tr>
<td>New (this work)</td>
<td>on feb. 9, 1864, he was sitting for several portraits. (Deletion)</td>
</tr>
<tr>
<td>Old (Xu et al., 2015)</td>
<td>on feb 9, 1864 , he was sitting for several , including the $ 5 bill for the bill. (Deletion)</td>
</tr>
</tbody>
</table>

Table 16: Examples of simplified sentences generated by LSTM and Transformerbert models trained on our new NEWSLE-AUTO (this work) and old existing NEWSLELA (Xu et al., 2015) datasets. The source sentences are from our new NEWSLE-AUTO test set. Models trained on our new data rephrase the input sentence more often than the models trained on old data. **Bold** indicates deletions or paraphrases.

<table>
<thead>
<tr>
<th><strong>Examples</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Complex (input)</td>
</tr>
<tr>
<td>Simple (reference)</td>
</tr>
<tr>
<td>LSTM</td>
</tr>
<tr>
<td>EditNTS (Dong et al., 2019)</td>
</tr>
<tr>
<td>Rerank (Kriz et al., 2019)</td>
</tr>
<tr>
<td>Transformerbert (this work)</td>
</tr>
<tr>
<td>Complex (input)</td>
</tr>
<tr>
<td>Simple (reference)</td>
</tr>
<tr>
<td>LSTM</td>
</tr>
<tr>
<td>EditNTS (Dong et al., 2019)</td>
</tr>
<tr>
<td>Rerank (Kriz et al., 2019)</td>
</tr>
<tr>
<td>Transformerbert (this work)</td>
</tr>
<tr>
<td>Complex (input)</td>
</tr>
<tr>
<td>Simple (reference)</td>
</tr>
<tr>
<td>LSTM</td>
</tr>
<tr>
<td>EditNTS (Dong et al., 2019)</td>
</tr>
<tr>
<td>Rerank (Kriz et al., 2019)</td>
</tr>
<tr>
<td>Transformerbert (this work)</td>
</tr>
</tbody>
</table>

Table 17: Examples of simplifications generated by our best model, Transformerbert, and other baselines, namely, EditNTS (Dong et al., 2019), Rerank (Kriz et al., 2019) and LSTM on the old NEWSLELA test set. Both LSTM and Transformerbert are trained on NEWSLE-AUTO. For EditNTS and Rerank, we use the system outputs shared by their original authors. **Bold** indicates new phrases introduced by the model.
E  Annotation Interface

E.1  Crowdsourcing Annotation Interface

Instructions:

- **A and B are equivalent**
  - **Case 1:** A simplify B or B simplify A (equivalent in meaning, though differ in length):
    
    Please fully understand this example!
    This is the most crucial part of this task!

    A: They could be killed by the terrorists if they come down from the mountain.
    B: The people risk death if they descend.

    Two sentences convey the same meaning, while one sentence is simpler than the other one.

    Don’t judge by sentence length! Instead, judge by readability of the sentence.

- **Case 2:** A and B are equivalent in both meaning and readability:
  
  A: They were trying to gather information and watch as the situation gets worse.
  B: They were trying to gather information and monitor the worsening situation.

  Two sentences are completely equivalent, as they mean the same thing.

  Differing in some very unimportant information is acceptable.

- **A and B are partially overlapped:**
  - **Case 1:**
    
    A: The trip was disastrous, and Bishop promised herself she’d never fly with Nathaniel again.
    B: The trip was very hard

    One sentence contains most of the information of the other one. It also contains important extra information.

    The length of extra information should be equal or longer than a long phrase.

  - **Case 2:**
    
    A: Some Republicans have called for the president to take action and have said he doesn’t need the approval of lawmakers.
    B: Some Republicans have asked the president to take action, but the White House was waiting for more information to make decision.

    Two sentences share some information in common. And each of them also contains extra information.

    The length of extra information should be equal or longer than a long phrase.

- **A and B are mismatched:**
  
  A: The technology is new and very advanced.
  B: The scientists hope it will also work on existing smartphones.

  The two sentences are completely dissimilar in meaning.

Questions:

---

**Sentence A**

The competition with West Point, which is now an annual affair, has grown into a rivalry.

**Sentence B**

The inmates have formed a popular debate club.

---

What's the relationship between **Sentence A** and **Sentence B**?

- **A and B are equivalent**
  
  A and B are equivalent (convey the same meaning, though one sentence can be much shorter or simpler than the other sentence).

- **A and B are partially overlapped**
  
  A and B are partially overlapped (share information in common, while some important information is different/misleading).

- **A and B are mismatched**
  
  The two sentences are completely dissimilar in meaning.

---

Figure 5: Instructions and an example question for our crowdsourcing annotation on the Figure Eight platform.
E.2 In-house Annotation Interface

Figure 6: Annotation interface for correcting the crowdsourced alignment labels.