Neural CRF Sentence Alignment Model for 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. We also 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 method outperforms all the previous approaches on monolingual sentence alignment task by more than 5 points in F1. We apply our aligner to construct NEWSELA-AUTO and WIKI-AUTO text simplification datasets, which are larger and of better quality compared to the existing datasets. A Transformer model trained on our datasets establishes a new state-of-the-art for sentence simplification in both automatic and human evaluation.

1 Introduction
Text simplification aims to rewrite complex text into simpler language while retaining its original meaning (Saggion, 2017). Text simplification provides 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 improves 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 containing complex-simple sentence pairs. Two widely used corpora, NEWSELA (Xu et al., 2015) and WIKI-LARGE (Zhang and Lapata, 2017), were created by automatically aligning sentences between comparable articles. However, due to the lack of reliable annotated data, sentence pairs are often aligned using surface-level similarity metrics, such as Jaccard coefficient (Xu et al., 2015) and cosine distance of TF-IDF vectors (Paetzold et al., 2017), which fail to capture paraphrases and also ignore 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, WIKI-LARGE is a concatenation of three early datasets (Zhu et al., 2010; Woodsend and Lapata, 2011; Coster and Kauchak, 2011) that are extracted from Wikipedia dumps and is 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 groups, and WIKI-MANUAL with 500 article pairs. We also propose a novel neural CRF alignment model, which utilizes fine-tuned BERT to measure semantic similarity and leverages the similar order of content between parallel documents, followed by an effective paragraph alignment algorithm. Exper-

---

1We will release our code and data upon the publication.

2Hwang et al. (2015) annotated 46 article pairs from Simple-Normal Wikipedia corpus; however, its annotation is noisy, and it contains many sentence splitting errors.
Figure 1: An example of sentence alignment between an original news article (right) and its simplified version (left) in Newsela. The label $a_i$ for each simple sentence $s_i$ is the index of complex sentence $c_{a_i}$ it is aligned to.

iments 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 groups in Newsela and 137,595 article pairs in Wikipedia dump, we construct two new simplification datasets, namely NEWSLA-AUTO (666,645 sentence pairs) and WIKI-AUTO (491,096 sentence pairs). Our new datasets with improved quantity and quality facilitate the training of complex seq2seq models, such as Transformers. A BERT-initialized Transformer trained on our datasets outperforms the state-of-the-art by 3.4% in terms of SARI, the main automatic metric for simplification. Our simplification model produces 25% more rephrasing when compared to its equivalent 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 aligner that employs fine-tuned BERT to capture semantic similarity and takes advantage of the sequential nature of parallel documents after combining with an effective paragraph alignment algorithm;
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 which is used for the first time for text simplification and establishes a new state-of-the-art in both automatic and human evaluation when trained on our datasets.

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 $a$ between article 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 pair (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
linear-chain conditional random field:

\[
p(a|S, C) = \frac{\exp(\Psi(a, S, C))}{\sum_{a \in A} \exp(\Psi(a, S, C))} = \frac{\exp(\sum_{i=1}^{[S]} \psi(a_i, a_{i-1}, S, C))}{\sum_{a \in A} \exp(\sum_{i=1}^{[S]} \psi(a_i, a_{i-1}, S, C))} \tag{1}
\]

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:

\[
\psi(a_i, a_{i-1}, S, C) = \text{sim}(s_i, c_{a_i}) + T(a_i, a_{i-1}) \tag{2}
\]

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). But we fine-tuned BERT (Devlin et al., 2019) with 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 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 neural scoring function to model the transition between alignment labels using the following features:

\[
\begin{align*}
g_1 &= |a_i - a_{i-1}| \\
g_2 &= 1(a_i = 0, a_{i-1} \neq 0) \\
g_3 &= 1(a_i \neq 0, a_{i-1} = 0) \\
g_4 &= 1(a_i = 0, a_{i-1} = 0)
\end{align*} \tag{3}
\]

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:

\[
T(a_i, a_{i-1}) = \text{FFNN}([g_1, g_2, g_3, g_4]) \tag{4}
\]

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

### 2.3 Inference and Learning

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

\[
\hat{a} = \arg\max_a P(a|S, C) \tag{5}
\]

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

\[
\log P(a^*|S, C) = \Psi(a^*, S, C) - \log \sum_{a \in A} \exp(\Psi(a, S, C)) \tag{6}
\]

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 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 data and reduces label space for our neural CRF model. For these reasons, 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/maximizing over the sentence-level semantic 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 \( 0^{k \times l} \)

for \( i \leftarrow 1 \) to \( k \) do
  for \( j \leftarrow 1 \) to \( l \) do
    \( simP[i, j] = \frac{1}{n} \max_{s_p \in S_i, c_q \in C_j} \text{simSent}(s_p, c_q) \)
  end
end

return \( simP \)

Algorithm 2: Paragraph Alignment Algorithm

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

for \( i \leftarrow 1 \) to \( k \) do
  \( j_{\text{max}} = \arg\max_{j} \text{simP}[i,j] \)
  if \( \text{simP}[i,j_{\text{max}}] > \tau_1 \) and \( d(i,j_{\text{max}}) < \tau_2 \) then
    \( \text{alignP}[i,j_{\text{max}}] = 1 \)
  end

for \( j \leftarrow 1 \) do
  if \( \text{simP}[2,i,j] > \tau_3 \) then
    \( \text{alignP}[i,j] = 1 \)
  end
  if \( j > 1 \) & \( \text{simP}[2,i,j] > \tau_4 \) & \( \text{simP}[2,i,j-1] > \tau_4 \) & \( d(i,j-1) < \tau_5 \) then
    \( \text{alignP}[i,j] = 1 \)
  end
  \( \text{alignP}[i,j-1] = 1 \)
end

return \( \text{alignP} \)

is defined as \( d(i, j) = \frac{1}{2} | j - \frac{i}{2} | \), and the thresholds \( \tau_1 - \tau_5 \) in Algorithm 2 are selected using the dev set. Finally, we merge the neighboring 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 B.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 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 §3), 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 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. We provide the interface in Appendix E.2.
2. For each sentence pair within the aligned paragraphs, we ask five annotators on the Figure

Table 1: Statistics of our manually and automatically created sentence alignment annotations on Newsela. \( \dagger \) 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.

<table>
<thead>
<tr>
<th>Article level</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>
<tr>
<td>Sentence level</td>
<td></td>
<td></td>
</tr>
<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(\dagger)</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>266,420</td>
</tr>
<tr>
<td>avg. length of simple sent.</td>
<td>14.8</td>
<td>14.8</td>
</tr>
<tr>
<td>avg. length of complex sent.</td>
<td>21.3</td>
<td>24.9</td>
</tr>
<tr>
<td>Labels of sentence pairs</td>
<td></td>
<td></td>
</tr>
<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>(\approx 0.99)</td>
</tr>
<tr>
<td># of not-aligned</td>
<td>(\approx 0.99)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Text simplification phenomenon</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>(\approx 0.99)</td>
</tr>
<tr>
<td># of sent. fusion (m-to-n)</td>
<td>252</td>
<td>(\approx 0.99)</td>
</tr>
<tr>
<td># of sent. deletion (1-to-0)</td>
<td>6,247</td>
<td>(\approx 0.99)</td>
</tr>
</tbody>
</table>

\(\dagger\)We consider any sentence pair not in the aligned paragraph pairs as not-aligned. This assumption leads to a small number of missing sentence alignments, which are manually corrected in Step 3.
Figure 2: Manual inspection of 100 random sentence pairs from our corpora (NEWSELA-AUTO and WIKI-AUTO) and the existing Newsela (Xu et al., 2015) and Wikipedia (Zhang and Lapata, 2017) corpora. Our corpora contain at least 44% more complex rewrites (Deletion + Paraphrase or Splitting + Paraphrase) and 27% less defective pairs (Not Aligned or Not Simpler).

Eight\(^4\) crowdsourcing platform to classify into one of the three categories: aligned, partially-aligned, or not-aligned. We provide the annotation instructions and interface in Appendix E.1. We require annotators to spend at least ten seconds per question and embed one test question in every five questions. Any worker whose accuracy drops below 85% on test questions is removed. The inter-annotator agreement is 0.807 measured by Cohen’s kappa (Artstein and Poesio, 2008).

3. We have four in-house annotators (not authors) verify the crowdsourced labels.

We manually aligned 50 article groups (named as NEWSELA-MANUAL) and split them into train/dev/test sets containing 35/5/10 article groups, respectively. We then trained our aligner on this annotated dataset (details in §4) to automatically align sentences in the remaining 1,882 article groups in Newsela (Table 1). We constructed a new sentence-aligned dataset (NEWSELA-AUTO) with 666k sentence pairs classified as aligned and partially-aligned (i.e., rephrasing and splitting cases), which is considerably larger than the previous dataset with 141,582 pairs. Figure 2 shows that our NEWSLE-AUTO contains 44% more complex rewrites than the existing NEWSELA (Xu et al., 2015) corpus.

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 noisier than the Newsela corpus. We provide this dataset primarily to facilitate future research.

First, we 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 (details in Appendix §A). 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.\(^5\)

Then, we crowdsourced the sentence alignment annotations for 500 randomly sampled document pairs (7,959 sentence pairs). As document length in English and Simple English Wikipedia articles vary greatly,\(^6\) 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 (Štajner 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 NEWSLEA-MANUAL data (§3.1). As these four metrics may rank the same sentence at the top, on an average, we collected 2.06 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 5002/889/2068 sentence pairs.

Finally, we trained alignment model by fine-

\(^4\)https://www.figure-eight.com/
\(^5\)https://github.com/attardi/wikiextractor
\(^6\)The average number of sentences in an article is 9.2 ± 16.5 for Simple English Wikipedia and 74.8 ± 94.4 for English Wikipedia.
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 ngram overlap 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 while reporting performance as they are trivial to classify.

4.3 Results

Table 2 shows the results on **NEWSLELA-MANUAL** test set. For similarity-based methods, we choose a threshold for 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 fine-tuned BERT model (Devlin et al., 2019) achieves 74.2 F1 for Task 1 (aligned + partially-aligned vs. not-aligned) on the **WIKI-MANUAL** test set. It outperforms all the previous SOTA approaches by more than 4 points in F1. We provide additional details in Appendix C.
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).

4.4 Ablation Study

We analyze the design choices crucial for the high performance of our alignment model, namely CRF component, paragraph alignment and the BERT-based semantic similarity measure. Table 3 shows the importance of each choice with a series of ablation experiments on the dev set.

**CRF Model** Our aligner achieves 93.2 F1 and 88.1 F1 on Task 1 and 2, 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 better handle sentence splitting, 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 the other neural sentence pair models, including InferSent (Conneau et al., 2017), ESIM (Chen et al., 2017) and the pre-trained BERT embeddings (Devlin et al., 2019).

5 Experiments on Automatic Sentence Simplification

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

5.1 Comparison with existing datasets

Existing datasets of complex-simple sentences, NEWSLELA (Xu et al., 2015) and WIKILARGE (Zhang and Lapata, 2017), were aligned using lexical similarity metrics. NEWSLELA 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 Wikipedia and editing history were aligned by TF-IDF cosine similarity.

For our new dataset NEWSLELA-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 of WIKILARGE. Table 4 shows the statistics of the existing and our new datasets.

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., 2019).

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 out-

---

https://github.com/yuedongP/EditNTS
We report SARI, the main automatic metric for simplification, precision for deletion and F1 scores for adding and keeping operations. We also show Flesch-Kincaid (FK) grade level readability, and average sentence length (Len). Add scores are low partially because we are using one reference. Bold typeface and underline denote the best and the second best performances respectively. For FK and Len, we consider the values closest to reference as the best.

Table 5: Automatic evaluation results on NEWSELA test sets comparing models trained on our new dataset NEWSELA-AUTO against the existing dataset (Xu et al., 2015). We report SARI, the main automatic metric for simplification, precision for deletion and F1 scores for adding and keeping operations. We also show Flesch-Kincaid (FK) grade level readability, and average sentence length (Len). Add scores are low partially because we are using one reference. Bold typeface and underline denote the best and the second best performances respectively. For FK and Len, we consider the values closest to reference as the best.

Table 6: Human evaluation of fluency (F), adequacy (A) and simplicity (S) on the old NEWSELA test set. †We used the system outputs shared by the authors.

Table 7: Human evaluation of fluency (F), adequacy (A) and simplicity (S) on NEWSELA-AUTO test set.

Figure 3: Manual inspection of 100 random sentences generated by Transformer\textsubscript{bert} trained on NEWSELA-AUTO and existing NEWSELA datasets respectively.
Table 8: Automatic evaluation results on Wikipedia TURK corpus comparing models trained on WIKI-AUTO and WIKILARGE (Zhang and Lapata, 2017).

<table>
<thead>
<tr>
<th>Complex (input)</th>
<th>SARI</th>
<th>add</th>
<th>keep</th>
<th>del</th>
<th>FK</th>
<th>Len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Models trained on old dataset (WIKILARGE)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM</td>
<td>34.2</td>
<td>2.8</td>
<td>68.6</td>
<td>31.9</td>
<td>11.6</td>
<td>20.7</td>
</tr>
<tr>
<td>Transformer$_{rand}$</td>
<td>31.3</td>
<td>2.9</td>
<td>58.3</td>
<td>32.7</td>
<td>10.8</td>
<td>16.8</td>
</tr>
<tr>
<td>Transformer$_{bert}$</td>
<td>32.6</td>
<td>3.7</td>
<td>60.3</td>
<td>33.6</td>
<td>10.8</td>
<td>16.9</td>
</tr>
<tr>
<td>EditNTS</td>
<td>33.7</td>
<td>2.6</td>
<td>66.6</td>
<td>32.5</td>
<td>11.2</td>
<td>18.3</td>
</tr>
</tbody>
</table>

| Models trained on our new dataset (WIKI-AUTO) | | | | | | |
| LSTM            | 34.4 | 2.9 | 69.1 | 32.1 | 11.3 | 20.6 |
| Transformer$_{rand}$ | 32.7 | 3.2 | 65.6 | 29.5 | 11.4 | 18.5 |
| Transformer$_{bert}$ | 34.1 | 4.3 | 64.8 | 33.1 | 11.0 | 17.7 |
| EditNTS         | 33.9 | 3.1 | 65.7 | 32.8 | 11.1 | 17.9 |

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 NEWSEA-AUTO and the existing dataset. Each worker evaluated these aspects on a 5-point Likert scale. We averaged the 5 ratings for the final value. Table 7 demonstrates that Transformer$_{bert}$ trained on NEWSEA-AUTO greatly outperforms the one trained on the old dataset. Even with 12.8% 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 NEWSEA-AUTO performs 25% more paraphrasing and deletions than its variant trained on the previous NEWSEA (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 preference for deletion-based simplification.

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 a dynamic programming (Barzilay and Elhadad, 2003; Paetzold et al., 2017) algorithm was used to search for the optimal alignment. Another 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 designed an efficient method to create two manually annotated sentence alignment datasets: NEWSEA-MANUAL and WIKI-MANUAL. We proposed a novel neural CRF sentence alignment model, which substantially outperformed the existing approaches on our new annotated datasets. We constructed two largest text simplification datasets to date, namely NEWSEA-AUTO and WIKI-AUTO. We showed that a BERT-initialized Transformer trained on our new datasets establishes a new state-of-the-art for automatic sentence simplification.

Acknowledgments

We thank 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 their help with data annotation. This material is based in part on research sponsored by the NSF under grants IIS-1822754 and IIS-1755898, DARPA through the ARO under agreement number W911NF-17-C-0095, IARPA through the BETTER program, through Figure-Eight AI for Everyone Award and a Criteo Faculty Research Award to Wei Xu. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of the U.S. Government.
References


A Sentence Aligned Wikipedia Corpus

We present more details about the our preprocessing step for the Wikipedia corpus here. In Wikipedia, Simple English is considered as a language by itself. When extracting articles from the 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.

B Neural CRF Alignment Model

B.1 Implementation Details

We used PyTorch\(^9\) to implement our neural CRF alignment model. For the sentence encoder, we used BERT\(^{base}\)\(^10\) architecture. Table 9 summarizes the hyperparameters of our model. Table 10 provides the thresholds for our paragraph alignment algorithm 2

<table>
<thead>
<tr>
<th>Thresholds</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>batch size</td>
<td>8</td>
</tr>
<tr>
<td>max sequence length</td>
<td>8</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Thresholds</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>0.998913818299745</td>
</tr>
<tr>
<td>(\tau_5)</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Table 10: Paragraph alignment algorithm thresholds.

C Sentence Alignment on Wikipedia

In this section, we discuss the performance of different sentence alignment approaches on the WIKI-MANUAL dataset. For Wikipedia, we fine-tuned BERT without fine-tuning (BERT\(_{embedding}\)) performs worse than the lexical metrics. BERT model fine-tuned on NEWSELA-MANUAL dataset performs comparably to the one fine-tuned on WIKI-MANUAL dataset.

<table>
<thead>
<tr>
<th>Model</th>
<th>Test set</th>
<th>Dev set</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>Jaccard (Xu et al., 2015)</td>
<td>68.6</td>
<td>68.2</td>
</tr>
<tr>
<td>TF-IDF (Paetzold et al., 2017)</td>
<td>57.7</td>
<td>77.3</td>
</tr>
<tr>
<td>C3G (Štajner et al., 2018)</td>
<td>61.6</td>
<td>79.7</td>
</tr>
<tr>
<td>BERT(_{embedding})</td>
<td>44.8</td>
<td>55.7</td>
</tr>
<tr>
<td>BERT(_{Newsela})</td>
<td>64.4</td>
<td>85.6</td>
</tr>
<tr>
<td>BERT(_{Wiki}) (this work)</td>
<td>68.9</td>
<td>80.3</td>
</tr>
<tr>
<td>Jaccard (Xu et al., 2015)</td>
<td>77.0</td>
<td>70.0</td>
</tr>
<tr>
<td>TF-IDF (Paetzold et al., 2017)</td>
<td>74.1</td>
<td>72.5</td>
</tr>
<tr>
<td>C3G (Štajner et al., 2018)</td>
<td>67.4</td>
<td>77.5</td>
</tr>
<tr>
<td>BERT(_{embedding})</td>
<td>52.4</td>
<td>56.9</td>
</tr>
<tr>
<td>BERT(_{Newsela})</td>
<td>79.1</td>
<td>82.4</td>
</tr>
<tr>
<td>BERT(_{Wiki}) (this work)</td>
<td>83.3</td>
<td>78.3</td>
</tr>
</tbody>
</table>

Table 11: Performance of different models on the WIKI-MANUAL test set for Task 1. C3G refers to the continuous 3-gram feature in the CATS toolkit (Štajner et al., 2018).

Table 12: Performance of different models on the WIKI-MANUAL dev set for Task 1.

D Sentence Simplification

D.1 Implementation Details

We used Fairseq\(^{11}\) toolkit to implement our transformer and LSTM baselines. For transformer (Vaswani et al., 2017) baselines, we followed BERT\(_{base}\)\(^{12}\) architecture for both encoder and decoder. We initialized the encoder using BERT-base-uncased checkpoint. Rothe et al. (2019) used a similar model for sentence fusion and summarization. We trained each model using Adam opti-
mizer with a learning rate of 0.0001, linear learning rate warmup of 40k steps and 200k training steps. We tokenized the data with BERT word-piece tokenizer. Table 13 provides the rest of the parameters. For LSTM baseline, we replicated the LSTM encoder-decoder model used in Zhang and Lapata (2017). We preprocessed the data by replacing the named entities in a sentence using spaCy\(^\text{13}\) toolkit. We also replaced all the words with frequency less than three with \texttt{<UNK>>}. If our model predicts \texttt{<UNK>}, we replaced it with the aligned source word (Jean et al., 2015). Table 14 summarizes LSTM model parameters. We used 300-dimensional GloVe word embeddings (Pennington et al., 2014) to initialize the embedding layer.

<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 13: 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 14: Parameters of our LSTM model.

\(^{13}\)https://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:

<table>
<thead>
<tr>
<th>Original sentence:</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
</tr>
</tbody>
</table>

<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 need help for assistance had anticipated, costs of financing unsustainable shrugged of 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 has 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 much 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 omitted 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>Complex</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformer$_{ bert }$</td>
<td>Now at age 9, his teachers say Richie reads at the level of a student in high school, and his vocabulary is well above those of his classmates.</td>
</tr>
<tr>
<td>EditNTS</td>
<td>Now at age 9, his teachers say <em>that Richie reads high schoolwork.</em></td>
</tr>
<tr>
<td>Rerank</td>
<td>he say his classmates are using a special job.</td>
</tr>
<tr>
<td>LSTM</td>
<td>but it is well above those of his classmates.</td>
</tr>
<tr>
<td>Transformer$_{ bert }$</td>
<td>now he is age 9.</td>
</tr>
<tr>
<td>EditNTS</td>
<td>he can recall the special feeling when he was 7 years old.</td>
</tr>
<tr>
<td>Rerank</td>
<td>he could remember the special feeling when he was 7.</td>
</tr>
<tr>
<td>LSTM</td>
<td>he can remember the special feeling when he was 7.</td>
</tr>
<tr>
<td>Complex</td>
<td>Following the action, two middle-aged brothers slid down a hill holding signs.</td>
</tr>
<tr>
<td>Transformer$_{ bert }$</td>
<td>two middle-aged brothers slid down a hill holding signs.</td>
</tr>
<tr>
<td>EditNTS</td>
<td>two middle-aged brothers, 14, heard down a hill signs.</td>
</tr>
<tr>
<td>Rerank</td>
<td>he made a hill holding signs.</td>
</tr>
<tr>
<td>LSTM</td>
<td><em>&lt;unk&gt;</em> middle-aged brothers slid down a hill holding signs.</td>
</tr>
</tbody>
</table>

Table 15: Examples of simplifications generated by our best model, Transformer$_{ bert }$, and other baselines namely EditNTS (Dong et al., 2019), Rerank (Kriz et al., 2019) and LSTM on the old Newsela test set. **Bold** indicates new phrases introduced by the model.

### Generated by LSTM baseline

<table>
<thead>
<tr>
<th>Complex</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>New</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>Old</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>Complex</td>
<td>he thought another suggestion would be against the police. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>New</td>
<td>he recognized that another suggestion would be controversial with police groups. (Lexical Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Old</td>
<td>the Philadelphia Museum of Art has two famous selfie spots, both from the movie &quot;Rocky.&quot; (Lexical Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Complex</td>
<td>The Philadelphia Museum of Art has two picture spots. (Lexical Paraphrase + Deletion)</td>
</tr>
<tr>
<td>New</td>
<td>some Chicago residents got angry about it. (Phrasal Paraphrase)</td>
</tr>
<tr>
<td>Old</td>
<td>some Chicago residents got angry. (Deletion)</td>
</tr>
<tr>
<td>Complex</td>
<td>Emissions standards have been tightened, and the government is investing money in solar, wind and other renewable energy. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>New</td>
<td>the government is putting aside money for new types of energy. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Old</td>
<td>the government is investing in money, wind and other equipment. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>Complex</td>
<td>On Feb. 9, 1864, he was sitting for several portraits, including the one used for the $5 bill. (Phrasal Paraphrase + Deletion)</td>
</tr>
<tr>
<td>New</td>
<td>on Feb 9, 1864, he was sitting for several portraits. (Deletion)</td>
</tr>
<tr>
<td>Old</td>
<td>on Feb 9, 1864, he was sitting for several portraits, including the $5 bill for the bill. (Deletion)</td>
</tr>
</tbody>
</table>

Table 16: Examples of simplified sentences generated by LSTM and Transformer$_{ bert }$ models trained on our Newsela-Auto and existing Newsela datasets. The source sentences are from our new Newsela-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.
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):
    
    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.

    ![](https://example.com)

    Please fully understand this example! This is the most crucial part of this task!

- **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.

- **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.

- **A** and **B** are partially overlapped:
  - 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.

- **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 partially overlapped
- A and B are mismatched

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

![Sentence Alignment Viewer](image)

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