# 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.[1]

## 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,[2] 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-

[1] Code and data are available at: https://github.com/chaojiang06/wiki-auto. Newsela data need to be requested at: https://newsela.com/data/.

[2] 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.



*Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7943–7960

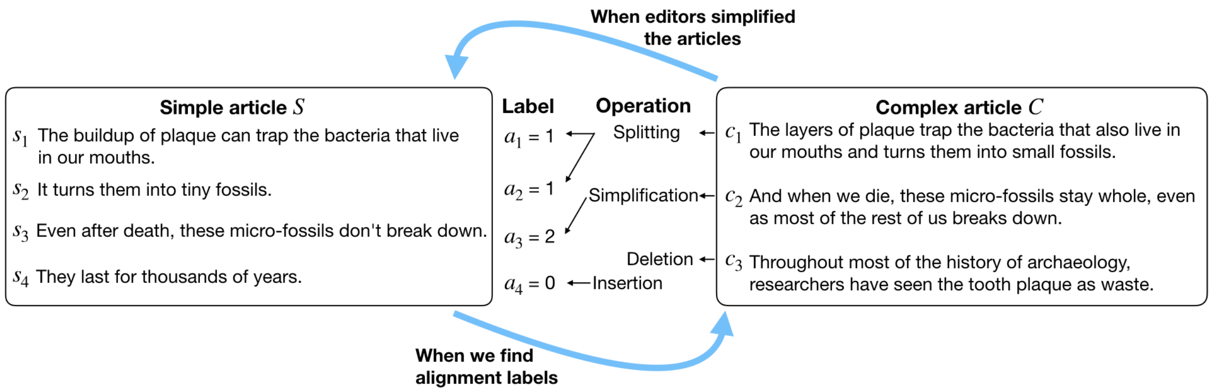


Figure 1: An example of sentence alignment between an original news article (right) and its simplified version (left) in Newsela. The label $a_i$ for each simple sentence $s_i$ is the index of complex sentence $c_{a_i}$ it aligns to.

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 Newsela and 138,095 article pairs in Wikipedia dump, we then construct two new simplification datasets, NEWSELA-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 $\mathbf{a}$ between article (or paragraph) pair $S$ and $C$ can then be represented by a sequence of alignment labels $\mathbf{a} = (a_1, a_2, \dots, a_m)$. 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 $P(\mathbf{a}|S, C)$, the conditional probability of alignment $\mathbf{a}$ given an article pair $(S, C)$, using linear-chain conditional random field:

$$P(\mathbf{a}|S,C) = \frac{\exp(\Psi(\mathbf{a},S,C))}{\sum_{\mathbf{a}\in\mathcal{A}}\exp(\Psi(\mathbf{a},S,C))} = \frac{\exp(\sum_{i=1}^{|S|}\psi(a_i,a_{i-1},S,C))}{\sum_{a\in\mathcal{A}}\exp(\sum_{i=1}^{|S|}\psi(a_i,a_{i-1},S,C)))} \tag{1}$$

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 $\mathbf{a} = (a_1, a_2, \ldots, a_m)$ between the simple article $S$ and the complex article $C$, and could be decomposed into two factors as follows:

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

where $sim(s_i, c_{a_i})$ 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:

$$\begin{aligned} g_1 &= |a_i - a_{i-1}| \\ g_2 &= \mathbb{1}(a_i = 0, a_{i-1} \neq 0) \\ g_3 &= \mathbb{1}(a_i \neq 0, a_{i-1} = 0) \\ g_4 &= \mathbb{1}(a_i = 0, a_{i-1} = 0) \end{aligned} \tag{3}$$

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 Appendix A.1.

### 2.3 Inference and Learning

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

$$\hat{\mathbf{a}} = \underset{\mathbf{a}}{\operatorname{argmax}}\, P(\mathbf{a}|S,C) \tag{5}$$

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

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

The second term sums the scores of all possible alignments and can be computed using forward algorithm in $\mathcal{O}(mn^2)$ 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 $simP$ between paragraphs by averaging or maximizing over the sentence-level similarities (§2.2). Then, we use Algorithm 2 to generate the paragraph alignment matrix $alignP$. 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**
    $simP[1,i,j] = \underset{s_p\in S_i}{\text{avg}}\left(\max_{c_q\in C_j} simSent(s_p,c_q)\right)$
    $simP[2,i,j] = \max_{s_p\in S_i, c_q\in C_j} simSent(s_p,c_q)$
  **end**
**end**
**return** $simP$

**Algorithm 2:** Paragraph Alignment Algorithm

**Input:** $simP \in \mathbb{R}^{2\times k\times l}$
**Initialize:** $alignP \in \mathbb{I}^{k\times l}$ to $0^{k\times l}$
**for** $i \leftarrow 1$ **to** $k$ **do**
  $j_{max} = \underset{j}{\text{argmax}}\, simP[1,i,j]$
  **if** $simP[1,i,j_{max}] > \tau_1$ *and* $d(i,j_{max}) < \tau_2$ **then**
    $alignP[i,j_{max}] = 1$
  **end**
  **for** $j \leftarrow 1$ **to** $l$ **do**
    **if** $simP[2,i,j] > \tau_3$ **then**
      $alignP[i,j] = 1$
    **end**
    **if** $j > 1$ & $simP[2,i,j] > \tau_4$ & $simP[2,i,j-1] > \tau_4$ & $d(i,j) < \tau_5$ & $d(i,j-1) < \tau_5$ **then**
      $alignP[i,j] = 1$
      $alignP[i,j-1] = 1$
    **end**
  **end**
**end**
**return** $alignP$

is defined as $d(i,j) = |\frac{i}{k} - \frac{j}{l}|$, 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

| | Newsela -Manual | Newsela -Auto |
|---|---|---|
| ***Article level*** | | |
| # of original articles | 50 | 1,882 |
| # of article pairs | 500 | 18,820 |
| ***Sentence level*** | | |
| # of original sent. (level 0) | 2,190 | 59,752 |
| # of sentence pairs | 1.01M† | 666,645 |
| # of unique complex sent. | 7,001 | 195,566 |
| # of unique simple sent. | 8,008 | 246,420 |
| avg. length of simple sent. | 13.9 | 14.8 |
| avg. length of complex sent. | 21.3 | 24.9 |
| ***Labels of sentence pairs*** | | |
| # of *aligned* (not identical) | 5,182 | 666,645 |
| # of *partially-aligned* | 14,023 | |
| # of *not-aligned* | 0.99M | – |
| ***Text simplification phenomenon*** | | |
| # of sent. rephrasing (1-to-1) | 8,216 | 307,450 |
| # of sent. copying (1-to-1) | 3,842 | 147,327 |
| # of sent. splitting (1-to-n) | 4,237 | 160,300 |
| # of sent. merging (n-to-1) | 232 | – |
| # of sent. fusion (m-to-n) | 252 | – |
| # of sent. deletion (1-to-0) | 6,247 | – |

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]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.

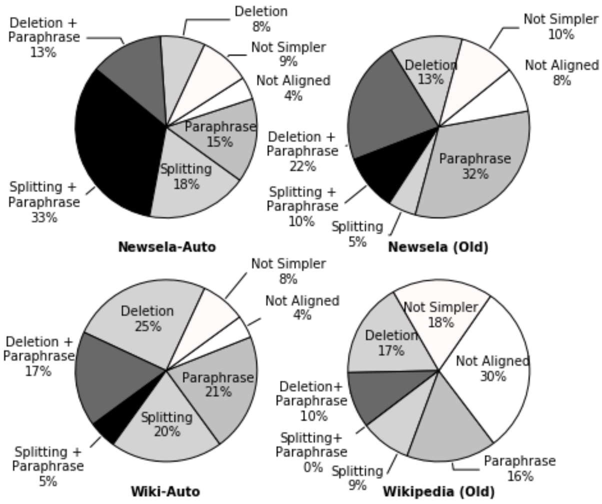


Figure 2: Manual inspection of 100 random sentence pairs from our corpora (NEWSELA-AUTO and WIKI-AUTO) and the existing Newsela (Xu et al., 2015) and Wikipedia (Zhang and Lapata, 2017) corpora. Our corpora contain at least 44% more complex rewrites (*Deletion + Paraphrase* or *Splitting + Paraphrase*) 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 to create the NEWSELA-MANUAL dataset with a 35/5/10 split for train/dev/test, respectively. We trained our aligner on this dataset (details in §4), then automatically aligned sentences in the remaining 1,882 article groups in Newsela (Table 1) to create a new sentence-aligned dataset, NEWSELA-AUTO, which consists of 666k sentence pairs predicted as *aligned* and *partially-aligned*. NEWSELA-AUTO is considerably larger than the previous NEWSELA (Xu et al., 2015) dataset of 141,582 pairs, and contains 44% more interesting rewrites (i.e., rephrasing and splitting cases) as shown in Figure 2.

[4]https://www.figure-eight.com/

### 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.[5]

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,[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 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

[5]https://github.com/attardi/wikiextractor

[6]The average number of sentences in an article is $9.2 \pm 16.5$ for Simple English Wikipedia and $74.8 \pm 94.4$ for English Wikipedia.

| | Task 1 (*aligned&partial* vs. *others*) | | | Task 2 (*aligned* vs. *others*) | | |
|---|---|---|---|---|---|---|
| | **Precision** | **Recall** | **F1** | **Precision** | **Recall** | **F1** |
| ***Similarity-based models*** | | | | | | |
| Jaccard (Xu et al., 2015) | 94.93 | 76.69 | 84.84 | 73.43 | 75.61 | 74.51 |
| TF-IDF (Paetzold et al., 2017) | 96.24 | 83.05 | 89.16 | 66.78 | 69.69 | 68.20 |
| LR (Štajner et al., 2018) | 93.11 | 84.96 | 88.85 | 73.21 | 74.74 | 73.97 |
| ***Similarity-based models w/ alignment strategy (previous SOTA)*** | | | | | | |
| JaccardAlign (Xu et al., 2015) | 98.66 | 67.58 | 80.22† | 51.34 | 86.76 | 64.51† |
| MASSAlign (Paetzold et al., 2017) | 95.49 | 82.27 | 88.39† | 40.98 | 87.11 | 55.74† |
| CATS (Štajner et al., 2018) | 88.56 | 91.31 | 89.92† | 38.29 | 97.39 | 54.97† |
| Our CRF Aligner | **97.86** | **93.43** | **95.59** | **87.56** | **89.55** | **88.54** |

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.

| | Task 1 | | | Task 2 | | |
|---|---|---|---|---|---|---|
| | **P** | **R** | **F1** | **P** | **R** | **F1** |
| ***Neural sentence pair models*** | | | | | | |
| Infersent | 92.8 | 69.7 | 79.6 | 87.8 | 74.0 | 80.3 |
| ESIM | 91.5 | 71.2 | 80.0 | 82.5 | 73.7 | 77.8 |
| BERTScore | 90.6 | 76.5 | 83.0 | 83.2 | 74.3 | 78.5 |
| $\text{BERT}_{embedding}$ | 84.7 | 53.0 | 65.2 | 77.0 | 74.7 | 75.8 |
| $\text{BERT}_{finetune}$ | 93.3 | 84.3 | 88.6 | 90.2 | 80.0 | 84.8 |
| + ParaAlign | 98.4 | 84.2 | 90.7 | 91.9 | 79.0 | 85.0 |
| ***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 |

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

annotated dataset to automatically align sentences for all the 138,095 document pairs (details in Appendix B). In total, we yielded 604k non-identical *aligned* and *partially-aligned* sentence pairs to create the WIKI-AUTO dataset. Figure 2 illustrates that WIKI-AUTO contains 75% less defective sentence pairs than the old WIKILARGE (Zhang and Lapata, 2017) dataset.

## 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.

| | Newsela | | Wikipedia | |
|---|---|---|---|---|
| | Auto | Old | Auto | Old |
| # of article pairs | 13k | 7.9k | 138k | 65k |
| # of sent. pairs (train) | 394k | 94k | 488k | 298k |
| # of sent. pairs (dev) | 43k | 1.1k | 2k | 2k |
| # of sent. pairs (test) | 44k | 1k | 359 | 359 |
| avg. sent. len (complex) | 25.4 | 25.8 | 26.6 | 25.2 |
| avg. sent. len (simple) | 13.8 | 15.7 | 18.7 | 18.5 |

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).

**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, NEWSELA-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, NEWSELA (Xu et al., 2015) and WIKILARGE (Zhang and Lapata, 2017), were aligned using lexical similarity metrics. NEWSELA 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 NEWSELA-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.

## 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.
3. A **BiLSTM-based encoder-decoder** model used in Zhang and Lapata (2017).
4. **EditNTS** (Dong et al., 2019),[7] 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.[8] We provide the model and training details in Appendix D.1.

[7] https://github.com/yuedongP/EditNTS
[8] https://github.com/pytorch/fairseq

| | Evaluation on our new test set | | | | | | Evaluation on old test set | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | **SARI** | **add** | **keep** | **del** | **FK** | **Len** | **SARI** | **add** | **keep** | **del** | **FK** | **Len** |
| Complex (input) | 11.9 | 0.0 | 35.5 | 0.0 | 12 | 24.3 | 12.5 | 0.0 | 37.7 | 0.0 | 11 | 22.9 |
| ***Models trained on old dataset*** (original NEWSELA corpus released in (Xu et al., 2015)) | | | | | | | | | | | | |
| Transformer$_{rand}$ | 33.1 | 1.8 | 22.1 | 75.4 | 6.8 | 14.2 | 34.1 | 2.0 | 25.5 | **74.8** | 6.7 | 14.2 |
| LSTM | 35.6 | 2.8 | **32.1** | 72.0 | 8.2 | 16.9 | 36.2 | 2.5 | **34.9** | 71.3 | 7.7 | 16.3 |
| EditNTS | 35.5 | 1.8 | 30.0 | 75.4 | 7.1 | 14.1 | 36.1 | 1.7 | 32.8 | 73.8 | 7.0 | 14.1 |
| Transformer$_{bert}$ | 34.4 | 2.4 | 25.2 | **75.8** | 7.0 | 14.5 | 35.1 | 2.7 | 27.8 | **74.8** | 6.8 | 14.3 |
| ***Models trained on our new dataset*** (NEWSELA-AUTO) | | | | | | | | | | | | |
| Transformer$_{rand}$ | 35.6 | 3.2 | 28.4 | 75.0 | 7.1 | 14.4 | 35.2 | 2.5 | 29.7 | 73.5 | 7.0 | 14.2 |
| LSTM | 35.8 | 3.9 | 30.5 | 73.1 | 7.0 | 14.3 | 36.4 | 3.3 | 33.0 | 72.9 | 6.6 | 14.0 |
| EditNTS | 35.8 | 2.4 | 29.4 | 75.6 | 6.3 | 11.6 | 35.7 | 1.8 | 31.1 | 74.2 | **6.1** | 11.5 |
| Transformer$_{bert}$ | **36.6** | **4.5** | 31.0 | 74.3 | **6.8** | **13.3** | **36.8** | **3.8** | 33.1 | 73.4 | 6.8 | **13.5** |
| Simple (reference) | – | – | – | – | 6.6 | 13.2 | – | – | – | – | 6.2 | 12.6 |

Table 5: Automatic evaluation results on NEWSELA test sets comparing models trained on our 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. 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 Flesch-Kincaid (FK) grade level and average sentence length (Len), we consider the values closest to reference as the best.

| **Model** | **F** | **A** | **S** | **Avg.** |
|---|---|---|---|---|
| LSTM | 3.44 | 2.86 | 3.31 | 3.20 |
| EditNTS (Dong et al., 2019)† | 3.32 | 2.79 | **3.48** | 3.20 |
| Rerank (Kriz et al., 2019)† | 3.50 | 2.80 | 3.46 | 3.25 |
| Transformer$_{bert}$ (this work) | **3.64** | **3.12** | 3.45 | **3.40** |
| Simple (reference) | 3.98 | 3.23 | 3.70 | 3.64 |

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.

| **Model** | **Train** | **F** | **A** | **S** | **Avg.** |
|---|---|---|---|---|---|
| LSTM | old | 3.57 | **3.27** | 3.11 | 3.31 |
| LSTM | new | 3.55 | 2.98 | 3.12 | 3.22 |
| Transformer$_{bert}$ | old | 2.91 | 2.56 | 2.67 | 2.70 |
| Transformer$_{bert}$ | new | **3.76** | 3.21 | **3.18** | **3.39** |
| Simple (reference) | — | 4.34 | 3.34 | 3.37 | 3.69 |

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

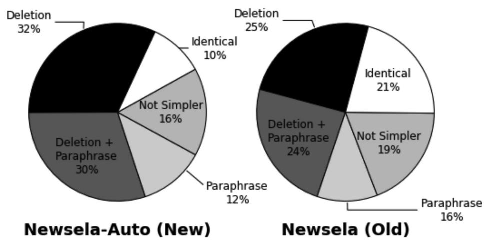


Figure 3: Manual inspection of 100 random sentences generated by Transformer$_{bert}$ trained on NEWSELA-AUTO and existing NEWSELA datasets, respectively.

### 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

| | SARI | add | keep | del | FK | Len |
|---|---|---|---|---|---|---|
| Complex (input) | 25.9 | 0.0 | 77.8 | 0.0 | 13.6 | 22.4 |
| ***Models trained on old dataset*** (WIKILARGE) | | | | | | |
| LSTM | 33.8 | 2.5 | 65.6 | 33.4 | 11.6 | 20.6 |
| Transformer$_{rand}$ | 33.5 | 3.2 | 64.1 | 33.2 | 11.1 | 17.7 |
| EditNTS | 35.3 | 3.0 | 63.9 | 38.9 | 11.1 | 18.5 |
| Transformer$_{bert}$ | 35.3 | 4.4 | 66.0 | 35.6 | 10.9 | 17.9 |
| ***Models trained on our new dataset*** (WIKI-AUTO) | | | | | | |
| LSTM | 34.0 | 2.8 | 64.0 | 35.2 | 11.0 | 19.3 |
| Transformer$_{rand}$ | 34.7 | 3.3 | **68.8** | 31.9 | **11.7** | 18.7 |
| EditNTS | 36.4 | 3.6 | 66.1 | **39.5** | 11.6 | **20.2** |
| Transformer$_{bert}$ | **36.6** | **5.0** | 67.6 | 37.2 | 11.4 | 18.7 |
| Simple (reference) | – | – | – | – | 11.7 | 20.2 |

Table 8: Automatic evaluation results on Wikipedia TURK corpus comparing models trained on WIKI-AUTO and WIKILARGE (Zhang and Lapata, 2017).

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 Transfomer$_{bert}$ 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; Tufiș 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-initalized 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 SocialSim 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


Ahmad Aghaebrahimian. 2018. Deep neural networks at the service of multilingual parallel sentence extraction. In *Proceedings of the 27th International Conference on Computational Linguistics*.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing*.

Ron Artstein and Massimo Poesio. 2008. Survey article: Inter-coder agreement for computational linguistics. *Computational Linguistics*.

Regina Barzilay and Noemie Elhadad. 2003. Sentence alignment for monolingual comparable corpora. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*.

Ohio Supercomputer Center. 2012. Oakley supercomputer. http://osc.edu/ark:/19495/hpc0cvqn.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996. Motivations and methods for text simplification. In *The 16th International Conference on Computational Linguistics*.

Han-Bin Chen, Hen-Hsen Huang, Hsin-Hsi Chen, and Ching-Ting Tan. 2012. A simplification-translation-restoration framework for cross-domain SMT applications. In *Proceedings of the 24th International Conference on Computational Linguistics*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

William Coster and David Kauchak. 2011. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Biological, translational, and clinical language processing*.

Simon Gottschalk and Elena Demidova. 2017. Multiwiki: interlingual text passage alignment in wikipedia. *ACM Transactions on the Web*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning sentences from standard Wikipedia to simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics*.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for WMT'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*.

Tomoyuki Kajiwara, Hiroshi Matsumoto, and Kazuhide Yamamoto. 2013. Selecting proper lexical paraphrase for children. In *Proceedings of the 25th Conference on Computational Linguistics and Speech Processing*.

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

Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-weighted loss and diverse reranking for sentence simplification. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

Shashi Narayan and Claire Gardent. 2014. Hybrid simplification using deep semantics and machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. Transforming complex sentences into a semantic hierarchy. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. MASSAlign: Alignment and annotation of comparable documents. In *Proceedings of the IJCNLP 2017, System Demonstrations*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*.

David Pellow and Maxine Eskenazi. 2014. An open corpus of everyday documents for simplification tasks. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations*.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing*.

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: A corpus analysis. In *Proceedings of Workshop on Speech and Language Technology for Education*.

Scott E. Reed and Nando de Freitas. 2016. Neural programmer-interpreters. In *4th International Conference on Learning Representations*.

Luz Rello, Ricardo Baeza-Yates, and Horacio Saggion. 2013. The impact of lexical simplification by verbal paraphrases for people with and without dyslexia. In *Proceedings of the 14th International Conference on Computational Linguistics and Intelligent Text Processing*.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*.

Horacio Saggion. 2017. Automatic text simplification. *Synthesis Lectures on Human Language Technologies*.

Advaith Siddharthan. 2006. Syntactic simplification and text cohesion. *Research on Language and Computation*.

Advaith Siddharthan and Napoleon Katsos. 2010. Reformulating discourse connectives for non-expert readers. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*.

Sanja Štajner, Hannah Béchara, and Horacio Saggion. 2015. A deeper exploration of the standard PB-SMT approach to text simplification and its evaluation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. CATS: A tool for customized alignment of text simplification corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*.

Brian Thompson and Philipp Koehn. 2019. Vecalign: Improved sentence alignment in linear time and space. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*.

Chen-Tse Tsai and Dan Roth. 2016. Cross-lingual wikification using multilingual embeddings. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*.

Dan Tufiș, Radu Ion, Ștefan Dumitrescu, and Dan Ștefănescu. 2013. Wikipedia as an SMT training corpus. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*.

Lucy Vanderwende, Hisami Suzuki, Chris Brockett, and Ani Nenkova. 2007. Beyond sumbasic: Task-focused summarization with sentence simplification and lexical expansion. *Inf. Process. Manage.*

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*.

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

Wei Xu and Ralph Grishman. 2009. A parse-and-trim approach with information significance for Chinese sentence compression. In *Proceedings of the 2009 Workshop on Language Generation and Summarisation*.

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

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

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

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics*.

# A Neural CRF Alignment Model

## A.1 Implementation Details

We used PyTorch[9] to implement our neural CRF alignment model. For the sentence encoder, we used Huggingface implementation(Wolf et al., 2019) of BERT$_{base}$ [10] 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.

| Parameter | Value | Parameter | Value |
|---|---|---|---|
| hidden units | 768 | # of layers | 12 |
| learning rate | 0.00002 | # of heads | 12 |
| max sequence length | 128 | batch size | 8 |

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

| Threshold | Value |
|---|---|
| $\tau_1$ | 0.1 |
| $\tau_2$ | 0.34 |
| $\tau_3$ | 0.9998861788416304 |
| $\tau_4$ | 0.998915818299745 |
| $\tau_5$ | 0.5 |

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.

| Threshold | Value |
|---|---|
| $\tau_1$ | 0.991775706637882 |
| $\tau_2$ | 0.8 |
| $\tau_3$ | 0.5 |
| $\tau_4$ | 5 |
| $\tau_5$ | 0.9958 |

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

**Algorithm 3:** Pairwise Paragraph Similarity

**Initialize:** $simP \in \mathbb{R}^{1\times k\times l}$ to $0^{1\times k\times l}$
**for** $i \leftarrow 1$ **to** $k$ **do**
  **for** $j \leftarrow 1$ **to** $l$ **do**
    $simP[1,i,j] = \max_{s_p\in S_i, c_q\in C_j} simSent(s_p, c_q)$
  **end**
**end**
**return** $simP$

**Algorithm 4:** Paragraph Alignment Algorithm

**Input :** $simP \in \mathbb{R}^{1\times k\times l}$
**Initialize:** $alignP \in \mathbb{I}^{k\times l}$ to $0^{k\times l}$
**for** $i \leftarrow 1$ **to** $k$ **do**
  $cand$ = []
  **for** $j \leftarrow 1$ **to** $l$ **do**
    **if** $simP[1,i,j] > \tau_1$ & $d(i,j) < \tau_2$ **then**
      $cand.append(j)$
    **end**
  **end**
  $range = max(cand) - min(cand)$
  **if** $len(cand) > 1$ & $range/l > \tau_3$ & $range > \tau_4$ **then**
    $dist = []$
    **for** $m \in cand$ **do**
      $dist.append(abs(m - i))$
    **end**
    $j_{cloest} = cand[\operatorname*{argmin}_n dist[n]]$
    **for** $m \in cand$ **do**
      **if** $m \neq j_{cloest}$&$simP[1,i,m] \leq \tau_5$ **then**
        $cand.remove(m)$
      **end**
    **end**
  **end**
  **for** $m \in cand$ **do**
    $alignP[i,m] = 1$
  **end**
**end**
**return** $alignP$

# B Sentence Aligned Wikipedia Corpus

We present more details about our pre-processing steps for creating the WIKI-MANUAL and WIKI-AUTO 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

[9]https://pytorch.org/
[10]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*.

| | Dev set | | |
|---|---|---|---|
| | **P** | **R** | **F** |
| MASSAlign (Paetzold et al., 2017) | 72.9 | 79.5 | 76.1 |
| CATS (Štajner et al., 2018) | 65.6 | 82.7 | 73.2 |
| BERT$_{finetune}$ (NEWSELA-MANUAL) | 82.6 | 83.9 | 83.2 |
| BERT$_{finetune}$ (WIKI-MANUAL) | 87.9 | 85.4 | 86.6 |
| + ParaAlign | 88.6 | 85.4 | 87.0 |
| Our CRF Aligner (WIKI-MANUAL) | 92.4 | 85.8 | 89.0 |

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

| | Test set | | |
|---|---|---|---|
| | **P** | **R** | **F** |
| MASSAlign (Paetzold et al., 2017) | 68.6 | 72.5 | 70.5 |
| CATS (Štajner et al., 2018) | 68.4 | 74.4 | 71.3 |
| BERT$_{finetune}$ (NEWSELA-MANUAL) | 80.6 | 78.8 | 79.6 |
| BERT$_{finetune}$ (WIKI-MANUAL) | 86.3 | 82.4 | 84.3 |
| + ParaAlign | 86.6 | 82.4 | 84.5 |
| Our CRF Aligner (WIKI-MANUAL) | 89.3 | 81.6 | 85.3 |

Table 13: Performance of different sentence alignment methods on the WIKI-MANUAL test set for Task 1.

## D Sentence Simplification

### D.1 Implementation Details

We used Fairseq[11] toolkit to implement our Transformer (Vaswani et al., 2017) and LSTM (Hochreiter and Schmidhuber, 1997) baselines. For the Transformer baseline, we followed BERT$_{base}$ [12] 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.

| **Parameter** | **Value** | **Parameter** | **Value** |
|---|---|---|---|
| hidden units | 768 | batch size | 32 |
| filter size | 3072 | max len | 100 |
| # of layers | 12 | activation | GELU |
| attention heads | 12 | dropout | 0.1 |
| loss | CE | seed | 13 |

Table 14: Parameters of our Transformer model.

| **Parameter** | **Value** | **Parameter** | **Value** |
|---|---|---|---|
| hidden units | 256 | batch size | 64 |
| embedding dim | 300 | max len | 100 |
| # of layers | 2 | dropout | 0.2 |
| lr | 0.001 | optimizer | Adam |
| clipping | 5 | epochs | 30 |
| min vocab freq | 3 | seed | 13 |

Table 15: Parameters of our LSTM model.

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 spaCy[13] 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.

[11] https://github.com/pytorch/fairseq
[12] https://github.com/google-research/bert
[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:

| **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. | | | |
| **Simpifications** | **Meaning** | **Grammar** | **Simplicity** |
| 1. Financial markets had expected Portugal's need for help because costs had become unsustainable and investors dismissed the news on Thursday. | 5 | 5 | 5 |
| 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. | 5 | 5 | 2 |
| 3. Financial markets the need need for assistance had anticipated, costs of financing unsustainable shrugged of the news Thursday. | 1 | 1 | 1 |
| 4. Financial markets had anticipated Portugal's need for assistance. | 2 | 5 | 5 |
| 5. Financial markets dismissed the news on Thursday. | 1 | 5 | 4 |

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 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 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 spitting 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

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

Table 16: Examples of simplified sentences generated by LSTM and Transformer$_{bert}$ models trained on our new NEWSELA-AUTO (this work) and old existing NEWSELA (Xu et al., 2015) 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.

| | Examples |
|---|---|
| Complex (input) | *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.* |
| Simple (reference) | *He reads like a high school student.* |
| LSTM | *now* ***he is*** *age 9.* |
| EditNTS (Dong et al., 2019) | *he say his classmates* ***are using a special job.*** |
| Rerank (Kriz et al., 2019) | ***but it*** *is well above those of his classmates.* |
| Transfomer$_{bert}$ (this work) | *now at age 9 , his teachers say* ***that richie reads high schoolwork.*** |
| Complex (input) | *He can recall the special feeling when, at age 7, he built his first kite and saw it waft into the air.* |
| Simple (reference) | *He can remember the special feeling when he built his first kite .* |
| LSTM | *he can* ***remember the people*** *when he was age 7.* |
| EditNTS (Dong et al., 2019) | *, at age 7, he built his first kite.* |
| Rerank (Kriz et al., 2019) | *he could* ***remember*** *the special feeling when.* |
| Transfomer$_{bert}$ (this work) | *he can* ***remember*** *the special feeling when he was 7 years old.* |
| Complex (input) | *Following the action , two middle-aged brothers slid down a hill holding signs.* |
| Simple (reference) | *For example , two grownup brothers slid down a hill holding signs .* |
| LSTM | *<unk> middle - aged brothers slid down a hill holding signs.* |
| EditNTS (Dong et al., 2019) | *two middle-aged brothers* ***, 14 ,*** *heard down a hill signs.* |
| Rerank (Kriz et al., 2019) | ***he made a*** *hill holding signs.* |
| Transfomer$_{bert}$ (this work) | *two middle-aged brothers slid down a hill holding signs.* |

Table 17: 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. Both LSTM and Transformer$_{bert}$ are trained on NEWSELA-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

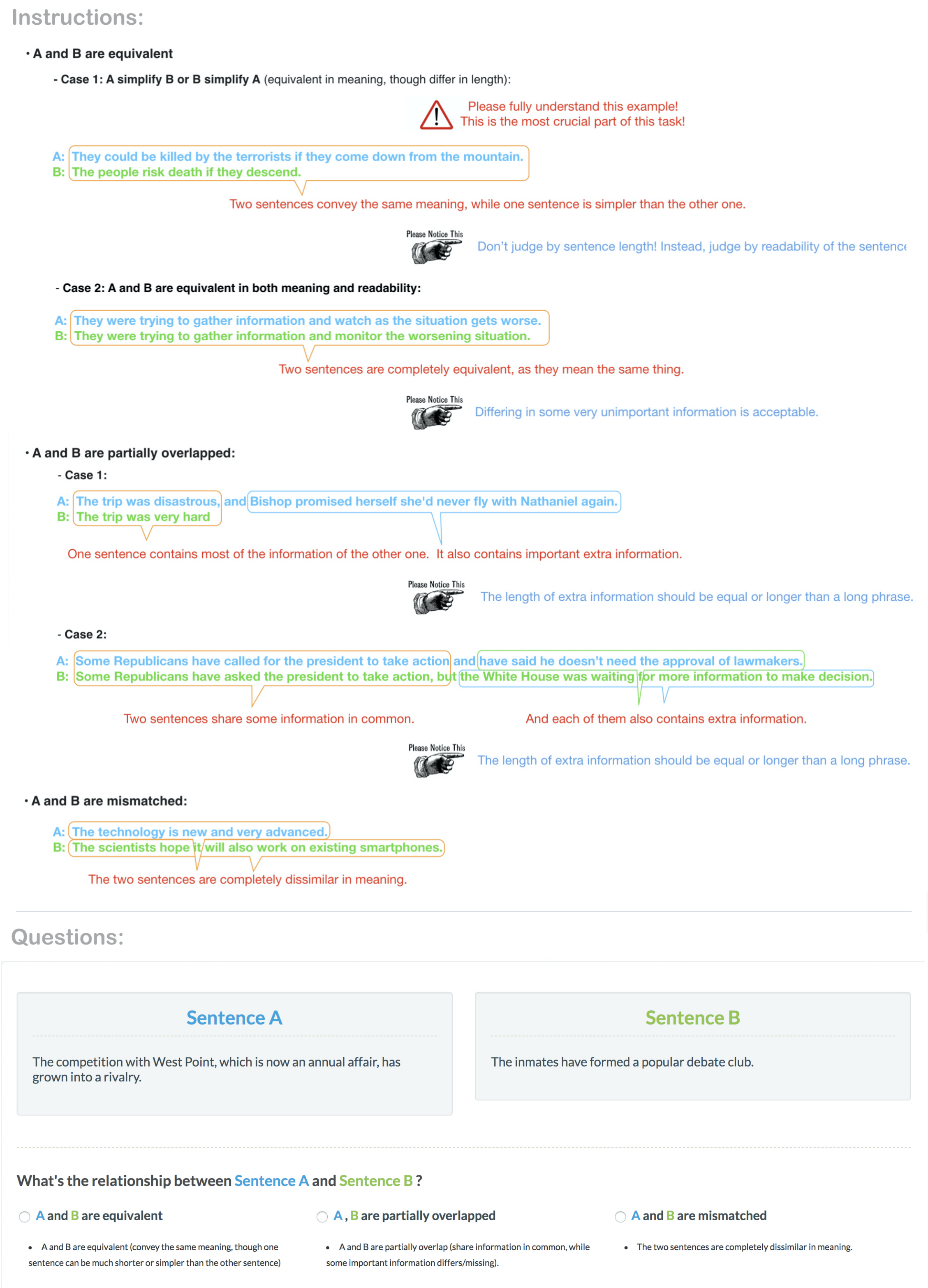


Figure 5: Instructions and an example question for our crowdsourcing annotation on the Figure Eight platform.

## E.2 In-house Annotation Interface

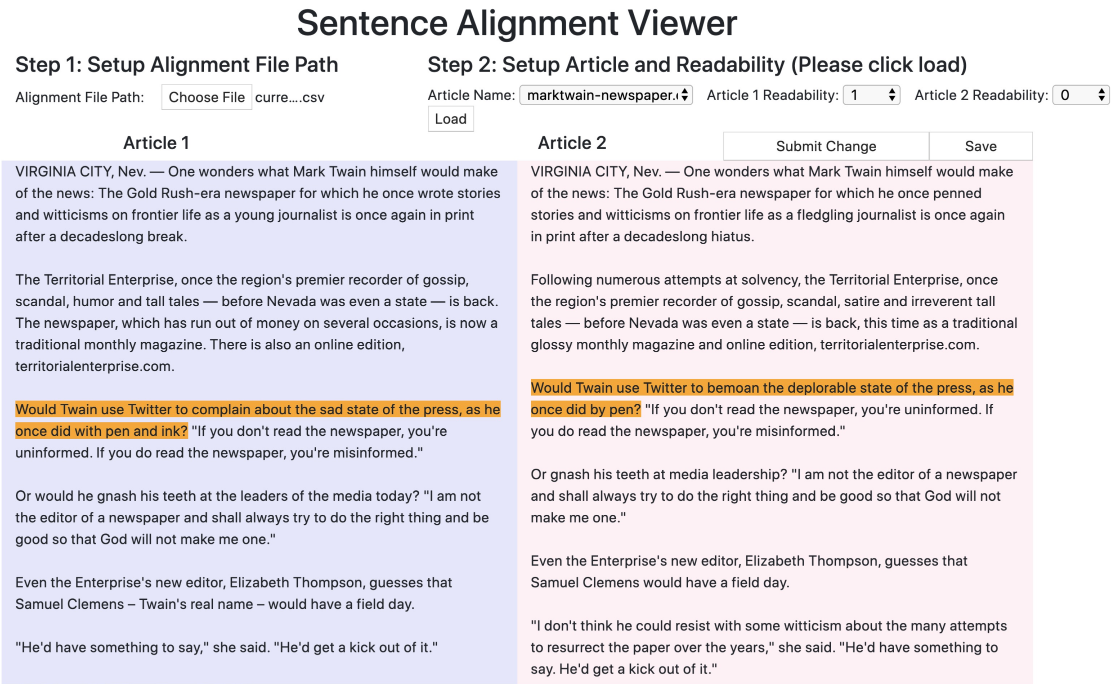


Figure 6: Annotation interface for correcting the crowdsourced alignment labels.