Improving Human Text Simplification with Sentence Fusion

Max Schwarzer
Mila
University of Montreal
max.schwarzer@umontreal.ca

Teerapaun Tanprasert and David Kauchak
Computer Science Department
Pomona College
teerapaun.tanprasert@pomona.edu
david.kauchak@pomona.edu

Abstract

The quality of fully automated text simplification systems is not good enough for use in real-world settings; instead, human simplifications are used. In this paper, we examine how to improve the cost and quality of human simplifications by leveraging crowdsourcing. We introduce a graph-based sentence fusion approach to augment human simplifications and a reranking approach to both select high quality simplifications and to allow for targeting simplifications with varying levels of simplicity. Using the Newsela dataset (Xu et al., 2015) we show consistent improvements over experts at varying simplification levels and find that the additional sentence fusion simplifications allow for simpler output than the human simplifications alone.

1 Introduction

Research on text simplification has largely focused on fully automated systems, including lexical systems that change words or phrases and sentence-level systems that make broader changes (Shardlow, 2014; Narayan and Gardent, 2016; Zhang and Lapata, 2017; Kriz et al., 2019). While the performance of such systems is steadily improving, for most real-world applications, the quality of these systems is still not good enough, particularly in domains where correctness is critical such as health and medical (Siddharthan, 2014; Shardlow and Nawaz, 2019). In such domains, human experts are still the main creators of simplified text (Zarcadoolas, 2010). The challenge is that these experts are costly to employ and the number of people equipped with the appropriate training and skills is limited.

In this paper, we examine a crowdsourcing approach to produce simplifications more efficiently and of higher quality using non-experts. Crowdsourcing has been suggested previously as a possible source of text simplifications (Amancio and Specia, 2014; Lasecki et al., 2015), however, no work has addressed quality control or how to deal with varying simplicity targets. The top part of Table 1 shows an example sentence to be simplified with two non-expert simplifications obtained through a crowdsourcing platform. While both of the human simplifications roughly convey the main idea in the original sentence, the quality is questionable. However, there are good portions of the simplifications, e.g., using "worried about" instead of "chief concerns". Our goal is to leverage these lower quality simplifications to generate high-quality simplifications that are as good as or better than those produced by an expert.

We make three main contributions. First, we describe a new sentence fusion technique for generating additional alternative simplifications based on the original input and the non-expert human simplifications. This allows for many additional simplifications to be generated by combining different portions of the original human simplifications. Second, we provide a supervised approach for ranking candidate simplifications, both human generated and sentence fusion generated. This allows the system to pick high quality simplifications from the candidates generated. Similar approaches have been used in translation for ranking and selecting both human and system translations (Callison-Burch, 2009; Zaidan and Callison-Burch, 2011). Third, we parameterize the ranking approach to optimize for different levels of simplicity allowing for different simplifications to be chosen depending on the simplicity target. This is particularly useful when combined with the sentence fusion technique which allows for a much broader range of possible candidates than just the human simplifications. We evaluate the proposed system against human expert simplifications and show consistently better results at varying simplicity levels for both simplicity and adequacy.

Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-15), pages 106–114

Original	Bird damage is often overshadowed by weather and water as a farmer's chief concerns.
Crowdsourced 1	Farmers problems with birds is over shadowed by weather and water.
Crowdsourced 2	A farmer is mostly worried about weather and water. But a farmer might also worry about birds causing damage.
Generated 1	Bird damage is often overshadowed by weather and water.
Generated 2	A farmer is mostly worried about weather and water as a farmer's chief concerns.
Generated 3	Farmers problems with birds is over shadowed by weather and water as a farmer's chief concerns.

Table 1: A sentence to be simplified (*Original*) with two crowdsourced simplifications. *Generated 1-3* are example sentences produced from the fusion graph of the original and crowdsourced sentences (the fusion graph is shown in Figure 2).

2 Improving Human Simplification

Crowdsourcing platforms allow for data to be generated quickly with reasonable quality for a modest price (Buhrmester et al., 2011). For text simplification, given a sentence to be simplified, we can solicit human simplifications from the crowdsourcing platform. However, the quality of the resulting simplifications is often of widely varying quality (Amancio and Specia, 2014); the workers are not experts and it can be difficult to give the workers the appropriate context, e.g., the target audience, etc.

We leverage these initial human simplifications to create higher quality simplifications. Specifically, given the original sentence, x, and non-expert human simplifications, $s_1, s_2, ..., s_n$, the goal is to produce a high-quality simplification of x. Previous work in translation (Zaidan and Callison-Burch, 2011) has shown that reasonable results can be obtained by automatically selecting the highest quality non-expert translation from those solicited, however, you are limited to those options available and additional iterations of human improvements were needed to get reasonable results.

To address these limitations, we extend the candidate simplifications by generating additional alternative candidate simplifications, $s'_1, s'_2, ..., s'_m$, using a graph-based fusion of $s_1, ..., s_n$. We then rank all of the candidate simplifications, i.e., $[s_1, ..., s_n, s'_1, ..., s'_m, x]$, which includes the human simplifications, the simplifications generated by sentence fusions, and the original unsimplified sentence (to allow for no simplification), and pick the top ranked option as the final simplification. To rank the sentences, we learn a model that optimizes a scoring function that combines simplicity and adequacy, though any scoring function could be used. We give details on each of these steps below.

2.1 Sentence Fusion

We use a graph-based sentence fusion approach where nodes represent words and directed edges denote candidate next words. The graph is created by adding each sentence to the graph a word at a time, connecting adjacent words in the sentence with a directed edge. New nodes are created for words that do not correspond to existing nodes in the graph.

We follow a similar approach to Filippova (2010), extended in two ways to adapt it to the text simplification domain. First, we create the initial graph using the words in the original sentence. This provides an initial node ordering where the information flow is correct and avoids a bias towards any of the human simplifications. Second, we restrict which words are considered equivalent and merged into a node. The original algorithm merged words that are lexically identical. For text simplification, structural reorderings are common and can create inappropriate transitions connecting content at the end of one simplification to content at the beginning of another and vice versa. These inappropriate transitions resulted in many low quality simplifications that were not always handled well with filtering and reranking. To avoid this and reduce the burden on the reranker, we word-align each human simplification, s_i, with the original sentence, x, using the Berkeley Aligner (Liang et al., 2006) and consider words as equivalent if they are lexically identical *and* aligned in the word alignment. The result is a less dense graph with less inappropriate paths.

Figure 1 shows the fusion graph over the example in Table 1 after building the graph first with the original sentence and then adding only the first crowdsourced sentence. Each path from START to END represents one candidate simplification. The graph is initially created with just the original sentence, which can be seen as START → bird →

damage → ... → END. The human simplification is then added to this graph, in this case adding an alternative way to start the sentence (START → farmers → problems) and the option to end the sentence early after "water".

Figure 2 shows the fusion graph after the second crowdsourced example is added. Several new nodes have been added representing alternative phrasings in this second sentence and many additional paths through the graph have also been added. As each additional crowdsourced sentence is added to the fusion graph, additional paths through the graph are created, resulting in more candidate sentences produced by the system. The density of the graph is dependent on the number of sentences fused, the lexical overlap between the sentences, and the diversity of phrasing. For readability, we have only shown the example with two crowdsourced sentences added. Table 1 shows three sentences generated from this graph.

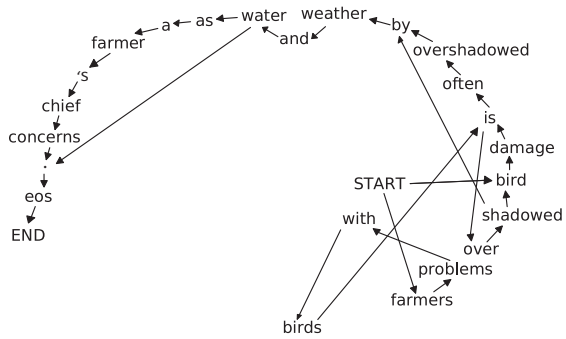

Figure 1: Fusion graph generated from only the *Original* and *Crowdsourced 1* sentences in Table 1. A directed edge (s, t) indicates that word t could follow word s in a candidate simplification.

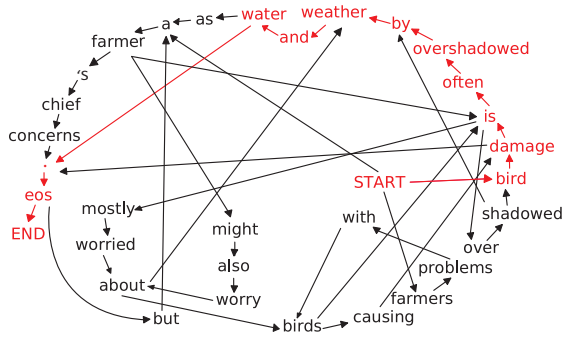

Figure 2: Fusion graph generated from the original and crowdsourced input sentences in Table 1 (the extension of Figure 1 after adding *Crowdsourced 2*). The path highlighted in red generates *Generated 1*.

2.2 Candidate Filtering

Any traversal of the graph from START to END represents a candidate simplification. In practice, the number of candidate simplifications encoded by the graph for actual examples can be huge and it is infeasible to generate all of the candidate options for ranking. To help identify higher quality candidate simplifications for the reranking stage we employ two techniques. First, we leverage characteristics of the words in the graph and the graph structure to impose an initial ordering of the candidate simplifications. We can then enumerate the candidate options from the graph based on this initial scoring, stopping after enough candidates have been generated. Second, we apply two additional filtering criteria to attempt to remove low quality candidates.

2.2.1 Graph ordering

To provide an initial ordering, we follow the heuristic from Filippova (2010) which weights edges in the graph based both on word frequency and graph path characteristics. Specifically, the weight of each edge $e_{i,j}$, representing the relationship between word i and word j, is computed as:

$$w(e_{i,j}) = \frac{f(i) + f(j)}{\sum_{s \in S} diff(s, i, j)^{-1}}$$

where $f(k)$ is defined as the frequency of word k in the sentences used to create the graph, S is the set of all sentences used to create the graph, $diff(s, i, j)$ is distance from the offset position of word i to word j in sentence s.

The formula prefers edges connecting a pair of words that frequently appear close to each other as well as those with lower *word* frequencies to *edge* frequency ratio (to discourage common words that have high edge frequency with many nodes). The first condition is enforced by the denominator, which prefers nodes with many paths between them, as well as nodes with short paths between them. The second condition is enforced by the numerator; if the sum of each word's frequencies is large, $w(e_{i,j})$ is subsequently large and thus not preferred.

The quality of a path through the graph is then the sum of the edge weights along that path. Given the weighted graph, we enumerate the candidate simplifications using lowest weight path traversals since lower edge weight denotes higher quality transitions. As an example, in Figure 2, "is" is one

of the nodes with several options of a successor. The edge between "is" and "often" has a weight of 1.33, while the other two outward edges, "is mostly" and "is over", both have weight 2.0. Therefore, all things being equal, the path including "is often" would be preferred over the other two.

2.2.2 Filtering

We also applied two filtering criteria to try and eliminate options that were obviously bad before ranking. To avoid simplifications that were too long or short, we filtered out candidates where the compression ratio (number of words in the original sentence divided by the number in the simplification) was more than one-standard deviation from the training set. To avoid simplifications that were too dissimilar from the original sentence, we filtered out candidates where their Siamese LSTM similarity score (see Section 2.3.2) was less than the average similarity score of the human simplifications and the original sentence.

We selected the first 1,000 sentences ordered by the lowest path graph traversal that passed these two filtering criteria to move on to the ranking stage (or less if generating all possible sentences from the graph yielded less). The 1,000 candidates is generated with `shortest_simple_paths` in the NetworkX library (Python), an implementation of the shortest path algorithm without repeated nodes (Yen, 1971).

2.3 Ranking

To choose the final simplification we combined and ranked the original sentence (to allow for no simplification), the human simplifications, and the sentence fusion candidates. We employed a supervised, feature-based, pairwise ranking approach using a linear SVM (Lee and Lin, 2014) with the implementation from Pedregosa et al. (2012).

2.3.1 Ranking Metric

Supervised ranking algorithms require training data of ranked examples. For our problem, a training example is a list of candidate simplifications, which we ranked with a quality score. Text simplification quality has been evaluated using both automated metrics, such as BLEU and SARI, and human evaluation metrics, including fluency, adequacy, and simplicity (Xu et al., 2016). Automated metrics require high-quality (i.e. expert) reference simplifications. Expert references are not available in many domains and, since our candidate outputs include crowdsourced sentences, it is unclear how a gold standard reference should be defined and obtained. Therefore, we utilize human metrics, which can be generated using non-experts.

Among the three human metrics, previous work has shown that fluency correlates with simplicity, and there is an intuitive tradeoff between simplicity and adequacy (Schwarzer and Kauchak, 2018): as sentences get simpler more content tends to be removed and the adequacy suffers. Therefore, we focus on simplicity and adequacy. The tradeoff between them can also be observed in the example shown in Table 2. For instance, the fourth sentence (*Generated 1*) is very simple, but the crucial contextual information about farmers is missing. On the other hand, the second sentence (*Crowdsourced 1*) retains most of the information in the original sentence, but also some redundant information. The tradeoff is reflected in their simplicity and adequacy scores.

To capture this tradeoff, we use a composite of simplicity and adequacy as our ranking metric during training. We define the score of a candidate simplification, s, as the weighted geometric mean of its normalized (0-1) adequacy, A_s, and simplicity, S_s,

$$\text{score}_\alpha(s) = \sqrt{A_s^\alpha \cdot S_s}.$$

Varying α biases the ranking towards simplicity (with lower α) or adequacy (with higher α). We only allow positive alpha. In the extremes, $\alpha = 0$ corresponds to optimizing only for simplicity and $\alpha = \infty$ only for adequacy.

2.3.2 Features

We used seven features for the ranking approach including two language model features and two features that quantify the similarity between the original sentence and the candidate simplification.

N-gram Language Model Log-prob normalized by the number of words in the sentence of a trigram language model using Kneser-Ney smoothing trained on the billion-word language model corpus (Chelba et al., 2013) using SRILM (Stolcke, 2002).

Neural Language Model Log-prob normalized by the number of non-stop words in the sentence of a recurrent-convolutional character-based neural language model (Kim et al., 2016).

Candidate Sentence	Simplicity	Adequacy	Ngram Logprob	Logprob	TF-IDF	Siamese	Comp. Ratio
Original	0.000	5.000	6.226	10.986	0.000	1.000	1.000
Crowdsourced 1	1.333	4.000	7.082	13.478	4.605	0.379	0.667
Crowdsourced 2	-0.667	3.333	7.151	10.376	4.493	0.287	1.556
Generated 1	1.667	2.333	6.161	9.548	1.620	0.385	0.667
Generated 2	N/A	N/A	6.408	12.322	2.650	0.497	0.889
Generated 3	N/A	N/A	6.833	13.686	3.038	0.520	1.000

Table 2: Sentences from Table 1 (in the same order) along with the features used to rerank them. Simplicity and adequacy scores are not available for the last two candidates because they did not get picked by the decile ranker for annotation in the experiments (see **Training** in 3.1 for more details).

TF-IDF Cosine Similarity TF-IDF cosine similarity between the original sentence and the candidate simplification, using sentence-level IDF values calculated from the Newsela corpus.

Siamese LSTM Two LSTM recurrent neural networks with shared weights trained on the SemEval2014 SICK dataset (Marelli et al., 2014) using fixed, pre-trained Google News word embeddings (Mikolov et al., 2013). The similarity is calculated by comparing the hidden states of two input sentences (Mueller and Thyagarajan, 2016).

Compression Ratio The ratio of the number of words of the original sentence versus the candidate simplification.

Source Label Two binary features, one indicating if the candidate is human-generated and one indicating if it is the original sentence.

3 Experiments

To evaluate our approach, we collected training and testing sets consisting of an original sentence and four human simplifications. To help better train the ranker, we also collected additional training data by scoring some simplifications generated by the sentence fusion approach. To understand the effect of alpha on the output, we trained rankers over 80 values of α, chosen to be densest near $\alpha = 1$, resulting in 80 different rankers that prioritize different levels of simplicity. We tested each of these models on the test set and compared them to four levels of expert human evaluations based on adequacy, simplicity, and fluency.

3.1 Data

We used the Newsela corpus (Xu et al., 2015) as the data set for evaluation. Newsela is a sentence-aligned corpus generated from articles manually simplified by experts at four simplicity levels (referred to as V1-V4, in order of increasing simplicity). We chose this dataset because it provides a strong baseline with expert simplifications and has multiple simplicity levels, which is suitable for testing our target-simplicity-specified rerankers.

Training We randomly selected 119 original sentences and collected 4 human simplifications and scored them for simplicity and adequacy. This data lacked examples of sentence fusion-generated simplifications that had been scored, and the initial ranker trained on it did not perform well.

To include sentence fusion examples in the data, we selected and scored some sentence fusion outputs. For each original sentence, we split the sentence fusion candidates into deciles based on the ranker (with $\alpha = 1$) and annotated the first sentence from each decile with simplicity and adequacy scores. This resulted in 10 sentence fusion simplifications per original sentence, in addition to the 4 human. We repeated this process: starting with the original sentences, annotating, and training on the freshly created dataset in each iteration. After two iterations, we observed approximate convergence in adequacy and simplicity scores on the training data and stopped iterating.

This new dataset consists of 119 original sentences, each with 4 human and 10 sentence fusion simplifications (15 candidate sentences per example, for a total of 1785 sentences) each annotated with simplicity and adequacy, and is used as the *training* data. Note that once the ranker has been trained, the only data required to apply the model to rank new sentences is the original sentence and the four crowdsourced simplifications. The generation and annotation procedure described above is only required to train the model.

Testing For the test set we chose an additional 200 random sentences where each original sentence was aligned with a sentence at each of the four simplicity levels (V1-V4). This allowed for a comparison of our approach against all four expert simplification levels.

Data Collection We used Amazon Mechanical Turk both to generate the four candidate human simplifications and to score simplifications (Callison-Burch and Dredze, 2010). The instruction for sentence simplification is to "make the sentence easier to understand" such that it "means the same thing as the original sentence". For adequacy and simplicity scores, the annotators are given the original and the simplified sentences and asked to judge to which degree the latter retains the meaning of or is simpler than the former, respectively, while fluency is annotated independently of the original sentence. We averaged judgments from three workers for each sentence. For simplicity, we asked the annotators to compare the simplification to the original on a five-point scale ranging from -2 (much less simple) to 2 (much simpler). Adequacy and fluency were assessed using on a five-point Likert scale with higher numbers representing better values.

Workers were required to be in the United States and have a historical 97% acceptance rate, but we placed no other restrictions on education, English proficiency, or previous simplification experience: the workers generating the simplifications were *not* experts. The full dataset (training and testing) with human evaluation scores is available online[1].

3.2 Results

Simplicity and Adequacy Figure 3 shows mean adequacy (1 to 5) and simplicity (-2 to 2) on the test set for Newsela V1-V4 and our approach (Reranked Joint) for a range of α. Higher is better denoting simpler output for simplicity and better content retention for adequacy. One of the main benefits of our approach is that different levels of simplicity can be targeted by varying α: the simplicity varies in the output ranging from points in the bottom right where no simplification occurs to points in the top center where significant simplification has happened. In general, the system output is both simpler and retains more information than the human expert baseline of Newsela. In particular, for all levels of Newsela (V1-V4) there is a setting of α where the system produces simplifications that have significantly better simplicity *and* adequacy. Table 3 gives examples along with statistical comparison based on a paired t-test.

[1] https://cs.pomona.edu/~dkauchak/simplification/

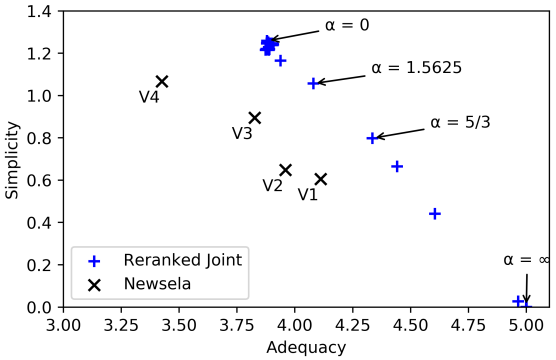

Figure 3: Average simplicity and adequacy scores for the system trained over a range of α compared to V1-V4 of Newsela on the test set.

Source	Simp.	Adequacy	Fluency
System ($\alpha = 5/3$)	**0.81**	**4.33**	4.15
Newsela V1	0.61^{**}	4.11^{***}	4.26^{*}
Newsela V2	0.65^{*}	3.96^{***}	**4.26***
System ($\alpha = 1.5625$)	**1.06**	**4.08**	4.11
Newsela V3	0.90^{**}	3.83^{***}	**4.24***
System ($\alpha = 0$)	**1.26**	3.88	4.00
Human-Only ($\alpha = 0$)	1.19^{*}	**3.94**	4.05
Newsela V4	1.06^{**}	3.43^{**}	**4.26*****

Table 3: Results for three α with statistical significance for comparable Newsela versions *, **, *** denoting $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively.

Fluency Table 3 also shows the fluency scores for three different α settings. These alphas were selected from the range explored in the experiments to highlight how different settings of alpha produced models with significantly better performance than human experts. For all approaches, the fluency is high with values ranging from 4.00 to 4.26. The system output is less fluent than the human experts, particularly at lower levels of α. To understand the cause of this difference, we compared the system fluency to the fluency of the *non-expert* (crowdsourced) humans that the system sentences were created from. For all three settings of α there is no statistically significant difference between the system output and the non-expert humans: the drop in fluency is a result of using non-expert humans.

Qualitative Table 4 shows an original sentence from the test dataset and the four crowdsourced simplifications. There is a fair amount of variabil-

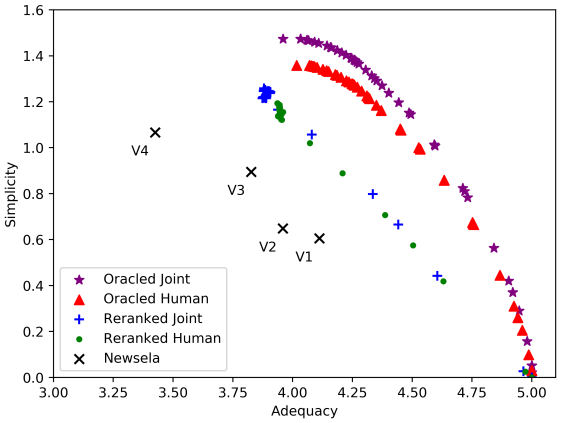

Figure 4: Simplicity and adequacy scores for reranking only the human simplifications as well as oracle output for both the full system (joint) and human only.

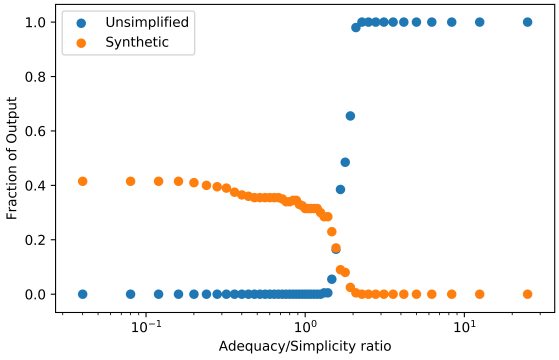

Figure 5: The fractions of output sentences coming from the sentence fusion system (synthetic) and unsimplified output sentences (the rest of the outputs are the crowdsourced simplifications), shown against the relative weightings of adequacy and simplicity.

ity in both the way that the text was simplified as well as the level of simplification. Crowdsourced 1 has only minor simplifications while the fourth is fairly aggressive. Crowdsourced 2 and 3 both split the sentence to try and make it simpler. The bottom part of the table shows the ranked output of our approach with $\alpha = 1$ (even balance between simplicity and adequacy). The top ranked choice (and therefore the one chosen) is a system fusion generated sentence; while simple, the sentence maintains the critical information in the original information. Table 4 also shows next three highest ranked options. The original sentence was ranked second (representing no simplification) followed by another system generated sentence and the first crowdsourced sentence.

3.3 Fusion and Ranking

We conducted additional experiments to understand the contributions of sentence fusion and ranking. To understand the contribution of the sentence fusion approach, we compared the general approach (Reranked Joint) to a version where only the four human simplifications were ranked (Reranked Human), i.e. without sentence fusion candidates (Figure 4). When adequacy is prioritized, the results are similar, however, as simplicity get prioritized more, the human simplifications are limited by the simplifications available. Adding sentence fusion allows for more varied simplifications, some of which are simpler. Table 3 gives a concrete example at $\alpha = 0$; the system is significantly simpler than the human only output, but there is no significant difference in adequacy or fluency.

Overall, the approach tends to select a combination of human and sentence fusion simplifications. Figure 5 shows the proportion of unsimplified and synthetic (fusion generated) sentences chosen as the best simplification by the ranker on the test data set for varying levels of α. For higher α, biasing towards adequacy, the system simply chooses not to simplify and selects the original unsimplified sentence. For the other values of α, however, the approach utilizes a combination of the human simplifications and the fusion generated (synthetic) simplifications, using the fusion generated sentence for 30-40% of the simplifications.

We also conducted an oracle study, where we picked the best simplification candidate based on the the simplicity/adequacy annotations ("Oracled" variations in Figure 4). This is similar to the approach of Zaidan and Callison-Burch (2011), and is an option when such annotations are available. We tested this for human simplifications only (Oracled Human) and the full system with human simplifications and the top sentence fusion candidate (Oracled Joint). Again, we see that the sentence fusion approach enables more simplification, providing candidates that are significantly simpler than those generated by humans when simplicity is prioritized. The performance gap between the reranked results and the oracled result suggests that there could still be room for improving the quality of the ranking.

Input Sentences		
Original		Ferguson has done dozens of studies on the subject and has consistently found that violent video games do not contribute to societal aggression.
Crowdsourced 1		Through dozens of studies on the subject, Ferguson has consistently found that violent video games do not contribute to societal aggression.
Crowdsourced 2		Ferguson found that violent video games do not contribute to societal aggression. He has done dozens of studies on the subject and has consistently come to the same conclusion.
Crowdsourced 3		Ferguson has completed many studies on the subject of violent video games. Ferguson concluded that these games do not contribute to societal aggression.
Crowdsourced 4		Ferguson did over 12 studies on it and saw that violent video games don't make people violent.
Ranked Output Sentences ($\alpha = 1.0$)		
1	System 1	Ferguson found that violent video games do not contribute to societal aggression.
2	Original	Ferguson has done dozens of studies on the subject and has consistently found that violent video games do not contribute to societal aggression.
3	System 2	Ferguson has completed many studies on the subject of violent video games do not contribute to societal aggression.
4	Crowdsourced 1	Through dozens of studies on the subject, Ferguson has consistently found that violent video games do not contribute to societal aggression.

Table 4: An example of real input from the test data set, consisting of the original sentence and four human simplifications, and top output sentences generated and ranked by an $\alpha = 1$ reranker.

4 Discussion

We introduced a new approach for leveraging crowdsourced human simplification that generates additional candidate simplifications using a sentence fusion technique and a reranking approach to pick high-quality simplifications. Our proposed approach is capable of producing simplifications that outperform expert human simplifications and the sentence fusion technique is particularly good at generating simpler variants.

We also introduced the new task of generating a high-quality text simplification based on crowdsourced simplifications. Our sentence fusion algorithm followed by reranking provides one possible approach, but there are a number of areas where it could be improved. We used a graph-based fusion approach, but recent neural approaches that have been applied in abstractive summarization may be adapted (Chopra et al., 2016; Nallapati et al., 2016). Many aspects of the reranker still need to be further explored. While the reranker did a reasonable job of selecting good candidates across different simplicity levels the oracle study (Figure 4) suggests that there is still room for improvement and additional features and alternative reranking algorithms should be investigated. The question of how well our trained reranker ports to different domains is also yet to be investigated. Future research on the relationships between α and simplicity is needed to establish a standard for choosing appropriate values of α as well. We hope this paper and the associated data provides a good starting point for future research in this area.

References

Marcelo Amancio and Lucia Specia. 2014. An analysis of crowdsourced text simplifications. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*.

Michael Buhrmester, Tracy Kwang, and Samuel D Gosling. 2011. Amazon's mechanical turk: A new source of inexpensive, yet high-quality, data? *Perspectives on Psychological Science*, 6(1):3–5.

Chris Callison-Burch. 2009. Fast, cheap, and creative: evaluating translation quality using amazon's mechanical turk. In *Proceedings of EMNLP*.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with amazon's mechanical turk. In *Proceedings of NAACL-HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, and Phillipp Koehn. 2013. One billion word benchmark for measuring progress in statistical language modeling. *arXiv:1312.3005*.

Sumit Chopra, Michael Auli, and Alexander M Rush. 2016. Abstractive sentence summarization with attentive recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 93–98.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of COLING*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *Proceedings of AAAI*.

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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3137–3147.

Walter S Lasecki, Luz Rello, and Jeffrey P Bigham. 2015. Measuring text simplification with the crowd. In *Proceedings of Web for All Conference*.

Ching-Pei Lee and Chih-Jen Lin. 2014. Large-scale linear ranksvm. *Neural computation*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of NAACL-HLT*.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. Semeval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the International Workshop on Semantic Evaluation (SemEval)*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of AAAI*.

Ramesh Nallapati, Bowen Zhou, Caglar Gulcehre, Bing Xiang, et al. 2016. Abstractive text summarization using sequence-to-sequence rnns and beyond. *arXiv preprint arXiv:1602.06023*.

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

Fabian Pedregosa, Elodie Cauvet, Gaël Varoquaux, Christophe Pallier, Bertrand Thirion, and Alexandre Gramfort. 2012. Learning to rank from medical imaging data. In *Proceedings of International Workshop on Machine Learning in Medical Imaging*.

Max Schwarzer and David Kauchak. 2018. Human evaluation for text simplification: The simplicity-adequacy tradeoff. In *Proceedings of SoCal NLP Symposium*.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*.

Matthew Shardlow and Raheel Nawaz. 2019. Neural text simplification of clinical letters with a domain specific phrase table. In *Proceedings of ACL*.

Advaith Siddharthan. 2014. A survey of research on text simplification. *International Journal of Applied Linguistics*.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*.

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

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*.

Jin Y Yen. 1971. Finding the k shortest loopless paths in a network. *management Science*, 17(11):712–716.

Omar F Zaidan and Chris Callison-Burch. 2011. Crowdsourcing translation: Professional quality from non-professionals. In *Proceedings of NAACL-HLT*.

Christina Zarcadoolas. 2010. The simplicity complex: exploring simplified health messages in a complex world. *Health Promotion International*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.