Text Simplification for Comprehension-based Question-Answering

Tanvi Dadu [*†]
NSIT, Delhi

Kartikey Pant [*†]
IIIT, Hyderabad

Seema Nagar
IBM Research India

Ferdous A. Barbhuiya
IIIT, Guwahati

Kuntal Dey [†]
Accenture Tech Labs

Abstract

Text simplification is the process of splitting and rephrasing a sentence to a sequence of sentences making it easier to read and understand while preserving the content and approximating the original meaning. Text simplification has been exploited in NLP applications like machine translation, summarization, semantic role labeling, and information extraction, opening a broad avenue for its exploitation in comprehension-based question-answering downstream tasks. In this work, we investigate the effect of text simplification in the task of question-answering using a comprehension context. We release *Simple-SQuAD*, a simplified version of the widely-used *SQuAD* dataset.

Firstly, we outline each step in the dataset creation pipeline, including style transfer, thresholding of sentences showing correct transfer, and offset finding for each answer. Secondly, we verify the quality of the transferred sentences through various methodologies involving both automated and human evaluation. Thirdly, we benchmark the newly created corpus and perform an ablation study for examining the effect of the simplification process in the *SQuAD*-based question answering task. Our experiments show that simplification leads to up to 2.04% and 1.74% increase in *Exact Match* and *F1*, respectively. Finally, we conclude with an analysis of the transfer process, investigating the types of edits made by the model, and the effect of sentence length on the transfer model.

1 Introduction

Text simplification is the task of modifying the structure of a text to make it easier to read and comprehend while preserving the content and approximating the original meaning. Linguistically, simple sentences are defined as having only one subject and one verb or predicate. Therefore, a complex sentence can be rewritten into multiple simpler sentences while retaining the same meaning. When simplifying texts, a myriad of rewriting transformations have been explored, ranging from replacing complex words or phrases for simpler synonyms, to changing the syntactic structure of the sentence. Moreover, modern automated text simplification approaches are data-driven, attempting to simplify sentences using parallel corpora of aligned complex-simplified sentences.

Text simplification allows humans to read texts more efficiently and faster, empowering them to take part in society and become more active in their daily actions and healthcare (Dannells, 2020). Another vital application of text simplification is reading assistance it provides, especially for people with reading disabilities (Carroll and Macdonald, 1998; Inui et al., 2003), low-literacy readers (Watanabe et al., 2009), or non-native speakers (Siddharthan, 2002).

There has been considerable work done where NLP applications exploit the benefits of text simplification. Long sentences with complex syntax or those laden with long-distance dependencies often pose difficulties for various downstream tasks. Text simplification has been used to improve the performance for these tasks including machine translation (Hasler et al., 2017), summarization (Silveira and Branco, 2012), semantic role labeling (Vickrey and Koller, 2008; Evans and Orasan, 2019), and information extraction (Evans and Orasan, 2019). This opens up the avenue for exploring simplification for question-answering. To the best of our knowledge, text simplification for comprehension-based question-answering has not been explored yet.

In this work, we make the following contributions:-

- We propose a transformers-based text-simplification pipeline that splits and

---

[*] The first two authors contributed equally to the work.

[†] This work was done when Kartikey Pant was affiliated to IIIT Hyderabad, Tanvi Dadu was affiliated to NSIT Delhi, and Kuntal Dey was affiliated to IBM.

Original SQuAD Sentence	Transferred Simple-SQuAD Sentence
Clark also claimed that Abdul gave him preferential treatment on the show <u>due</u> to their affair.	Clark also claimed that Abdul gave him preferential treatment on the show. <u>This was due</u> to their affair.
In his Prices and Production (1931), Hayek argued that the business cycle resulted from the central bank's inflationary credit expansion and its transmission over time, <u>leading</u> to a capital misallocation caused by the artificially low interest rates.	In his Prices and Production (1931), Hayek argued that the business cycle resulted from the central bank's inflationary credit expansion and its transmission over time. <u>This led</u> to a capital misallocation caused by the artificially low interest rates.

Table 1: Examples of Simple and Complex sentences, from our proposed Simple-SQuAD dataset and the SQuAD dataset. Underlined parts denotes the splitting point.

rephrases a complex sentence into its simpler constituent sentences. We outline each step in the dataset creation pipeline, including data preprocessing, performing text simplification, thresholding the simplified sentence based on the quality of the transfer, and offset finding of answers from each question-answer pair present in the dataset.

- We propose a new dataset *Simple-SQuAD* by converting each context in the *SQuAD* dataset using the proposed text-simplification pipeline. [1]

- We perform automated and human evaluation to determine the quality of the text simplification model.

- We perform event based analysis and sentence-length based transfer analysis to give deeper insights into the transfer process.

- We then benchmark both *Simple-SQuAD* and *SQuAD* for predictive performance in the *Simple-SQuAD* question-answering task.

2 Related Works

2.1 Text Simplification

Text simplification has attracted a great deal of attention due to its potential impact on society. Prior explorations on text simplification contain a myriad of approaches, which include from the use of hand-crafted syntactic rules (Carroll and Macdonald, 1998; Vickrey and Koller, 2008), statistical simplification model (Zhu et al., 2010), quasi-synchronous grammar (Woodsend and Lapata, 2011) and the semantic hierarchy of simplified sentences for recursively splitting and rephrasing complex sentences (Niklaus et al., 2019). Recently, Machine Translation, both statistical and neural, has also been

used for the task of text simplification (Narayan and Gardent, 2014; Nisioi et al., 2017).

The process of text simplification has been observed to help in the performance of multiple downstream tasks. Silveira et al. (Silveira and Branco, 2012) explored the use of a sentence simplification module in summarization systems, concluding that the simplification module removes expendable information that helps in accommodating relevant data in a summary. Hasler et al. (Hasler et al., 2017) showed that the performance of source simplification could improve translation quality in machine translation systems. Evans et al. (Evans and Orasan, 2019) proved the efficacy of integrating a text simplification step for improving the predictive performance of semantic role labeling and information extraction methodologies.

Narayan et al. (Narayan et al., 2017) introduced a new text simplification task, named Split-and-Rephrase on their proposed WebSplit dataset, containing $1,066,115$ parallel instances of complex and sequence of simple sentences having a similar meaning. The goal of the task is to split a complex input sentence into shorter sentences while preserving the meaning. In this task, the emphasis is on sentence splitting and rephrasing, with no deletion and no lexical or phrasal simplification. They further proposed five models, ranging from vanilla sequence-to-sequence to semantically-motivated models to benchmark the proposed task. Aharoni and Goldberg (Aharoni and Goldberg, 2018) and Botha et al. (Botha et al., 2018) extended the work by introducing new datasets, with a more extensive vocabulary and split examples to improve the efficacy of the prior benchmarks. We use WikiSplit, introduced by Botha et al. (Botha et al., 2018), for training our sentence simplification module.

2.2 Style Transfer

In recent works, textual style transfer has been shown to produce grammatically fluent, and

---

[1]Made available at the following Github repository: https://github.com/kartikeypant/text-simplification-qa-www2021.

information-preserved texts with fairly accurate target attributes. For the task in a semi-supervised setting, various methodologies are exploited, including back-translation (Prabhumoye et al., 2018), back-translation with attribute conditioning (Pant et al., 2020), specialized transfer methodologies like Delete, Retrieve, Generate (Li et al., 2018). However, in the presence of a large parallel corpus, sequence-to-sequence models perform competitively. Aharoni and Goldberg (Aharoni and Goldberg, 2018) exploited a copy-mechanism based sequence-to-sequence model with attention (Bahdanau et al., 2015) for the text simplification transfer.

Transformers (Vaswani et al., 2017) have been shown to perform robust language modeling, given enough data, helping in various downstream tasks. Due to the parallelized nature of the architecture, it is possible to train using much larger datasets than recurrent neural networks. However, it becomes necessary to optimize hyperparameters carefully while ensuring scalability (Popel and Bojar, 2018). Transformer-based methodologies for the task of sentence simplification have been explored by Marayuma and Yamamoto (Maruyama and Yamamoto, 2019) for Japanese and Zhao et al. (Zhao et al., 2018) for a smaller Wikipedia-based dataset. However, to the best of our knowledge, there has been no work done for supervised style transfer exploiting transformer-based models for the significantly large *WikiSplit* dataset.

3 Corpus Creation

3.1 Preliminaries

3.1.1 SQuAD

For exploring the effect of text simplification in the question-answering downstream task, we use Stanford Question Answering Dataset (SQuAD comprehension), a reading comprehension dataset released by Rajpurkar et al. (Rajpurkar et al., 2016). It consists of $100,000+$ questions from 536 articles posed by crowdworkers on a set of Wikipedia articles, where the answer to each question is a segment of text from the corresponding reading passage. Unlike other datasets, *SQuAD* does not provide a list of answer choices for each question; instead, systems must select the answer from all possible spans in the passage. Though the system must deal with a large number of candidate answers, yet span-based answers are easier to evaluate than free-form answers. Further, it is diverse in

terms of answer types, containing a significant percentage of dates, numeric data, adjective phrases, verb phrases, clauses as answers. The predictive performance of models in the *SQuAD* dataset is evaluated using *Exact-match* and *F1-score*.

3.1.2 WikiSplit

For the task of text simplification, we use the *WikiSplit* corpus (Botha et al., 2018), a parallel corpus consisting of complex sentences and its consequent sequence of simple sentences having similar meaning for the Split and Rephrase task. It contains a set of one million naturally occurring sentence rewrites mined from English Wikipedia, providing 60 times more examples and 90 times more vocabulary as compared to the *WebSplit* corpus introduced by Narayan et al. (Narayan et al., 2017). Using a larger dataset for the task of text simplification increases the efficacy of the model and the quality of the transferred data (Botha et al., 2018; Popel and Bojar, 2018).

3.2 Approach

3.2.1 Style Transfer

This subsection outlines the process of textual style transfer using transformers for converting the complex contexts in the *SQuAD* dataset into their simpler counterparts. Figure 1 illustrates the process for a sample context and question-answer pair from the *SQuAD* dataset.

We use spaCy's Sentencizer [2] to tokenize the contexts into their respective constituent sentences. This process enables us to perform the sentence-level transfer, thus ensuring higher degrees of overall transfer quality. We tokenize each sentence using a SentencePiece tokenizer, trained on 3.24 *GBs* English Wikipedia . While training the SentencePiece tokenizer, we mark the numerical tokens as custom-defined ones, ensuring no tokenization happens for such tokens. This step helps in preserving the numerical tokens in the transfer process by the transfer model.

We transfer each tokenized sentence using a transformers based machine translation model. We use the OpenNMT-py (Klein et al., 2017) toolkit for the implementation process. The model consists of 6-layered transformers architecture with 8 self-attention heads and 1028 sized hidden feed-forward layer. We trained the model for 20000 training steps with a dropout of 0.1 and a batch
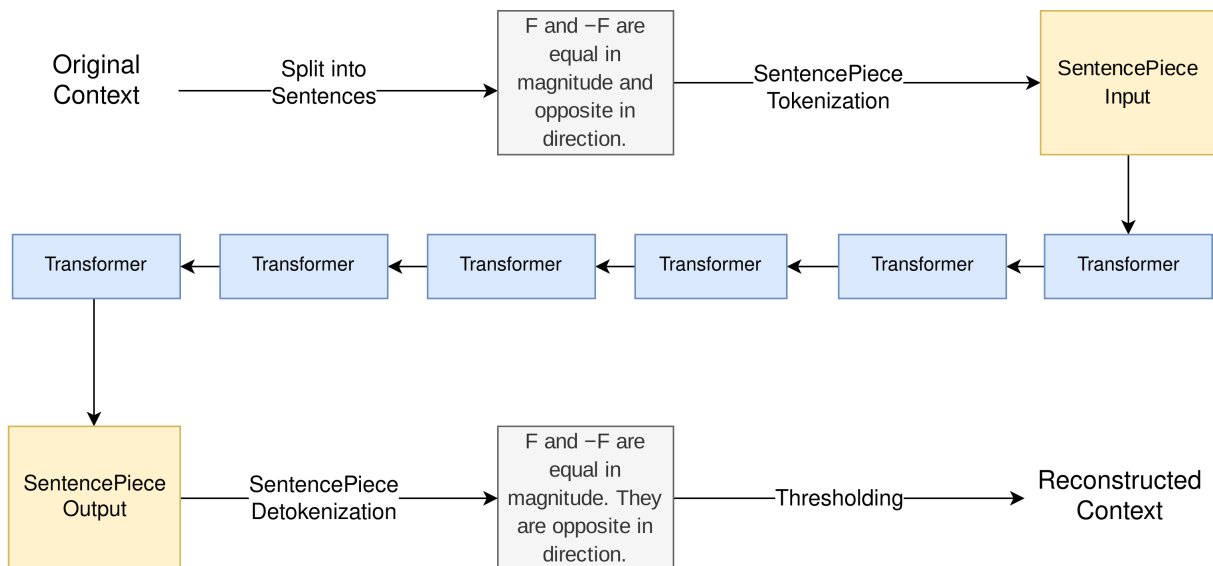
---

[2]https://spacy.io/api/sentencizer

3

Figure 1: Style Transfer Architecture

size of 2048. For optimization, we used Adam optimizer with a *beta2* value of 0.998, accumulating the gradient twice. We use an initial learning rate of 2 with *noam* decay method and 800 (4% of total) warmup steps.

3.2.2 Thresholding

In this section, we outline the thresholding techniques used in the generated sequence of simple sentences to maintain their quality. We execute the following post-editing steps to reduce noise from the dataset:

1. **Perplexity** : We use open *GPT-2* (Radford et al., 2018) as the language model to assign perplexity to the generated sentences. We threshold the transferred sequence of sentences that have the perplexity between 50 and 600 and discard all the remaining sequences. This thresholding ensures that the sequence of sentences is fluent, as measured by the language model. Table 2 shows that 84951 of 91757 sentences are within the fluency thresholding, comprising of 93% of the total sentences.

2. **Length of the original sentence**: We weed out all sentences whose original length was shorter than five tokens. This heuristic is based on the condition that sentences lesser than five tokens are highly likely to be simple, reducing the number of false positives to a significant extent efficiently. Table 2 shows that the thresholding process removes 15.16%

Thresholding Step	Percentage of Total Sentences	Thresholded Sentences
Step 1: Perplexity	92.58%	84,951
Step 2: Original Length Thresholding	78.5%	72,075
Step 3: Redundancy Thresholding	77.31%	70,944

Table 2: Sentences left after each thresholding process step

of the already thresholded sentences leaving 78.5% of the total sentences.

3. **Redundancy in sequence of the transferred sentences**: We remove all transferred sentences that display a redundancy behavior, which is defined to be one in which two or more sentences in the sequence are the same. We introduce this step after a careful analysis of the transferred sentences. Table 2 illustrates that the thresholding removes 1.57% of the already thresholded sentences leaving 77.31% of the total sentences.

3.2.3 Automated Evaluation

We perform an automated sentence-level analysis of the style transfer model. We determine the extent of content preservation, lexical simplicity, and reading ease of the generated sentences using the following three metrics:

1. **BLEU** (Papineni et al. (Papineni et al., 2002)): We use the self-BLEU score taking the original sentence as the reference to measure the

4

extent of content preservation. Table 3 shows that our model achieves a mean high self-BLEU of 75.66, implying that the transferred sentences display a high level of content-based similarity with the original sentence

2. **SARI** (Xu et al. (Xu et al., 2016)): We use the SARI metric to measure the quality of lexical simplicity in the transferred sentences. It analyzes the words added, deleted, and retained by a simplification model. Our version of SARI compares the model's output to the original sentence. There is a high correlation with human judgments of simplicity gain and the SARI metric (Xu et al., 2016). In Table 3, we observe a mean SARI value of 30.08 with a low standard deviation of 2.21, denoting most sentences to have a high level of lexical simplicity.

3. **Flesch–Kincaid Grade level** (Kincaid et al. (Kincaid et al., 1975)): Flesch– Kincaid Grade Level (FKGL) is a widely-used metric for text readability. It represents the corresponding U.S. grade level, whose education is appropriate for understanding the text. As illustrated in Table 3, the Flesch-Kincaid Grade Level for the sentences decreases by 5.43 grade levels on average. Thus, the transferred sentences have much higher readability than the original sentences.

Metric	Mean	Standard Deviation
BLEU	75.66	13.68
SARI	30.08	2.21
FKGL(Original)	13.29	5.74
FKGL(Transferred)	7.86	3.94

Table 3: Automated analysis of transferred sentences

3.2.4 Human Evaluation

Although we use commonly used metrics for the evaluation, automated evaluation of generative models of text is still an open research problem (Hu et al., 2017). We perform a human evaluation to analyze the quality of the transferred sentences accurately.

In the human evaluation, we randomly sampled 50 sentence pairs from the thresholded dataset consisting of both transferred sentences and their original variants. The human evaluators were asked to rate each sentence pair on a 1-5 Likert scale on the following metrics:

1. **Fluency**: A high *Fluency* score, 4 or 5, denotes that the transferred sentence is well constructed. A medium score of 3 denotes that the sentence contains lexical errors, while a low score of 2 or 1 denotes major errors and extremely poor constructions respectively.

2. **Relative Simplicity**: A sentence pair is given a high *Relative Simplicity* score, 4 or 5, if the transferred sentence is significantly simplified as compared to the original sentence. If the simplicity remains the same, it is given a medium score of 3 score. Moreover, a low score represents that the transferred sentence is more complex than the original sentence.

3. **Content Preservation**: A high *Content Preservation* score of 4 and 5 indicates that the content of the original sentence was well-preserved in the transferred sentence. A medium score of 3 denotes that there were minor differences between the transferred sentences and the original sentences. A low score of 2 and 1 denotes major violation of content preservation.

To verify the inter-rater agreement, we perform a Krippendorff's alpha analysis across all three metrics. Our analysis shows that we obtain an averaged Krippendorff's alpha inter-rater agreement of 0.63 over all metrics, denoting reasonable agreement between the raters.

Metric	Average Value
Fluency	4.26
Relative Simplicity	3.91
Content Preservation	4.10

Table 4: Human Evaluation Results

Table 4 illustrates the results of the performed human evaluation. We observe that the transferred sentences were judged to be fluent, significantly simplified, and content-preserved compared to their original counterparts.

3.2.5 Offset Finding and Dataset Finalization

In this subsection, we outline the process of finding the offsets for each question-answer pair and reconstructing our proposed simplified version of *SQuAD*, *Simple-SQuAD*, from the thresholded sentences.

Firstly, we preprocess the sentences, which involves removing the split sentence indicator and

reconstructing context texts from both thresholded and original sentences. If a sentence is not simplified, we use the original sentence itself, ensuring minimal loss of information due to the transfer process.

Secondly, for every question-answer pair, we calculate the answer's character offset by using exact matching in the reconstructed context. If we do not find an exact match, we use case-insensitive pattern matching to calculate the offset. We found the offsets for $88,690$ questions in total.

Finally, we create the following two datasets: *Simple-SQuAD* and *Original*. The *Simple-SQuAD* dataset contains the reconstructed context with the calculated character offsets for each answer. The *Original* dataset contains the original context itself but with only those questions whose answers are present in the *Simple-SQuAD* dataset. We create the new *Original* dataset to ensure an equal number of question-answer pairs, leading to a fair comparison. This *Original* dataset represents the *SQuAD* candidate for the benchmarking process.

4 Benchmarking Experiments

This section highlights the models and experiments performed for benchmarking the *Simple-SQuAD*. We compare the results obtained for *Simple-SQuAD* against the *SQuAD* dataset.

4.1 Model Used

For benchmarking *Simple-SQuAD*, we use two different variations of RoBERTa as introduced by Liu et al. (Liu et al., 2019). RoBERTa is a replication study of BERT pretraining, which is trained on more extensive training data with bigger batches, longer sequences, and dynamically changing masking patterns. Consequently, RoBERTa achieves better results over BERT and attains state-of-the-art results on *GLUE*, *RACE*, and *SQuAD*.

4.2 Experimental Settings

We perform a dataset-based ablation study, experimenting with multiple variants of input datasets for each model. Firstly, we finetune the model on the *SQuAD* and the *Simple-SQuAD* dataset separately for 2 epochs. We then finetune the *Simple-SQuAD* trained model on the *SQuAD* dataset and the *SQuAD*-trained model on the *Simple-SQuAD* dataset for 2 epochs each. We benchmark the results for each of these combinations of the dataset

input to better infer the effect of simplifying sentences in the original dataset.

For benchmarking, we use 442 training articles containing $78,810$ questions and 48 development articles containing 9880 questions. Thus, we have a 90:10 and an approximate 89:11 train-test split based on the number of articles and the number of questions, respectively. We used 10% of the training examples as a validation set for both our models.

For both $RoBERTa_{Base}$ and $RoBERTa_{Large}$, we use a maximum sequence length of 380, a stride of 128, a maximum query length of 64, and a maximum answer length of 30. We use a learning rate of $1*10^{-5}$ with a weight decay of 0.01.

4.3 Results

This section outlines the result of the ablation study to determine the effect of text simplification on the question answering downstream task in the *SQuAD* dataset. We observe that text simplification improves the predictive performance of both $RoBERTa_{base}$ and $RoBERTa_{large}$.

Model	Input	Exact	F1
$RoBERTa_{Base}$	SQuAD	0.787	0.863
$RoBERTa_{Base}$	Simple-SQuAD	0.786	0.866
$RoBERTa_{Base}$	Simple-SQUAD → SQuAD	0.799	0.876
$RoBERTa_{Base}$	SQuAD → Simple-SQuAD	**0.803**	**0.878**

Table 5: Benchmarking Results for $RoBERTa_{Base}$.

Table 5 illustrates the results for the $RoBERTa_{Base}$ model. We observe an increase of 2.03% in *Exact Match* and 1.74% in *F1* when fine-tuning it with *SQuAD* followed by *Simple-SQuAD*, in contrast with the model when trained on only SQuAD.

Model	Input	Exact	F1
$RoBERTa_{Large}$	SQuAD	0.835	0.905
$RoBERTa_{Large}$	Simple-SQuAD	**0.852**	**0.917**
$RoBERTa_{Large}$	Simple-SQUAD → SQuAD	0.838	0.907
$RoBERTa_{Large}$	SQuAD → Simple-SQuAD	0.836	0.908

Table 6: Benchmarking Results for $RoBERTa_{Large}$.

Table 6 illustrates the results for the $RoBERTa_{Large}$ model. Similar to $RoBERTa_{Base}$, we observe an increase of 2.04% in *Exact Match* and 1.33% in *F1* when fine-tuning it with *Simple-SQuAD* when compared to model when trained on *SQuAD*.

Category	Original	Transferred
Inter-Event Splitting	Although his administrative abilities <u>had been noticed</u>, on the eve of the U.S. entry into World War II he had never held an active command above a battalion and was far from <u>being considered</u> by many as a potential commander of major operations.	Although his administrative abilities <u>had been noticed</u> on the eve of the U.S. entry into World War II he had never held an active command above a battalion. He was far from <u>being considered</u> by many as a potential commander of major operations.
Intra-Event Splitting	Clark also <u>claimed</u> that Abdul gave him preferential treatment on the show due to their affair.	Clark also <u>claimed</u> that Abdul gave him preferential treatment on the show. This was due to their affair.

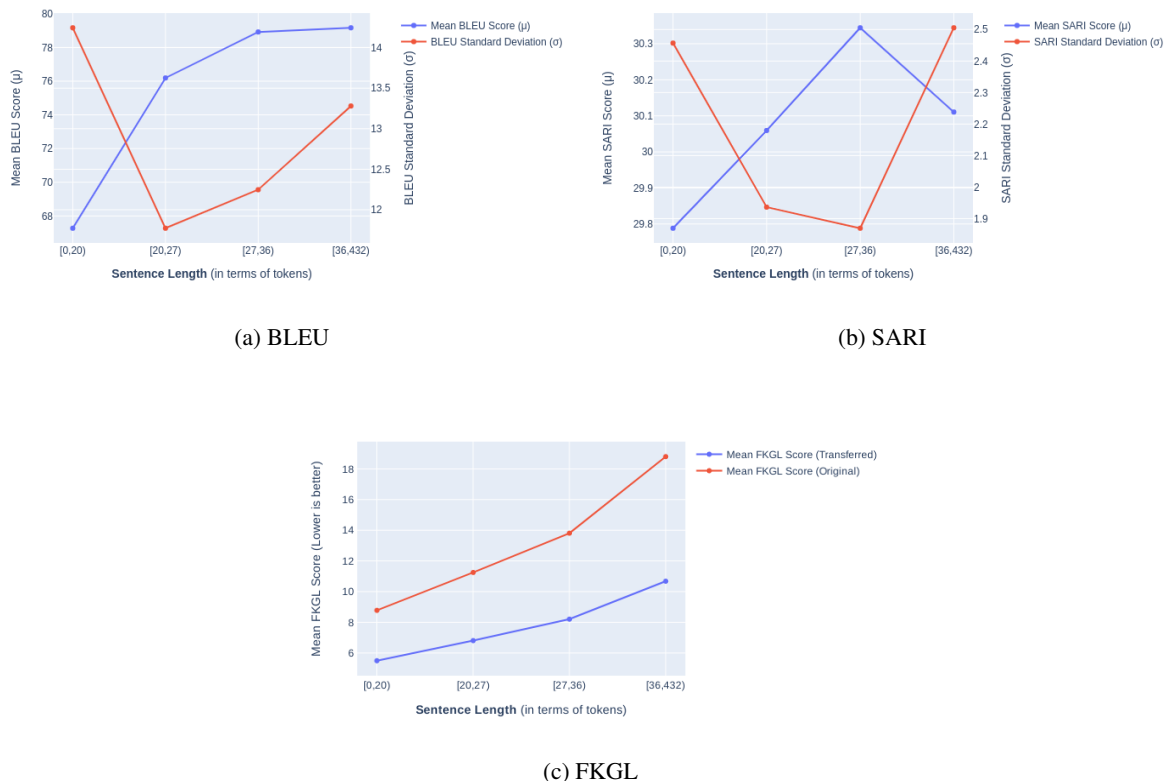Table 7: Examples of event-based splitting.

(a) BLEU

(b) SARI

(c) FKGL

Figure 2: Plots illustrating sentence-length based transfer analysis

5 Discussion

5.1 Edit Analysis

We conduct an event-based analysis of the edits performed by the transfer model to convert the originally complex sentences into simpler forms. We use the definition of a linguistic event as defined in Pustejovsky (Pustejovsky, 1991). We perform the analysis on 50 parallel sentences in their pre and post simplification forms, classifying each edit into the following two classes: *Inter-Event Splitting* and *Intra-Event Splitting*.

Inter-Event Splitting denotes the type of edit in which the model splits two different events. As illustrated in Table 7, the events of *notice* and *consider* are split into two different sentences, thus

simplifying the complex sentence containing the two events. On the other hand, *Intra-Event Splitting* is the type of edit in which all the simplified sentences contain the same event as the original sentence, as is illustrated for the event *claim* in Table 7.

In our analysis, we found 32% of the instances to show *Inter-Event Splitting*, showing that our model can capture event boundaries. On the other hand, 60% of the total instances show successful *Intra-Event Splitting*, illustrating that the model can capture intra-event detailing boundaries. Interestingly, 8% of the total instances displayed unsuccessful attempts of *Intra-Event Splitting*, which can be improved in future work.

7

5.2 Transfer Analysis

For transfer analysis, we divide all the sentences in *Simple-SQuAD* into four buckets based on the original sentence length split equally in terms of word-level tokenization ($0-20$ tokens, $20-27$ tokens, $27-36$ tokens and $36-432$ tokens). We then compute the following three metrics on a sentence level: BLEU score, SARI scores, and FKGL scores. In Figure 2, we observe that the performance of the model varies with the sentence length.

1. **BLEU**: Sentence-preservation, measured through average BLEU score, is directly proportional to the sentence length. However, the standard deviation of the BLEU scores first decreases then increases as we increase the sentence length.

2. **SARI**: Lexical simplicity, measured through average SARI score, first increases then decreases as we increase the sentence length. However, the standard deviation of the SARI scores first decreases then increases as sentence length increases.

3. **FKGL**: Text readability, measured via sentence-level FKGL score, was computed for both transferred and original sentences. We observe that sentence-level FKGL score of transferred sentences is directly proportional to the sentence length. Whereas sentence-level FKGL scores for original sentences first decreases then increases with sentence length. Moreover, the mean sentence-level FKGL score for the transferred sentences was always lower than that for the original sentences regardless of the sentence length.

6 Conclusion

In this work, we study the effect of text simplification in the comprehension based question-answering downstream task using the *SQuAD* dataset. For *Simple-SQuAD* corpus creation, we use a transformers based style transfer model to transfer complex sentences to sequences of simple sentences while retaining the original meaning. We further use post-editing techniques to reduce noise from the dataset, followed by the use of heuristics to find required offsets for an answer in each question-answer pair. We prove the efficacy of our model using automated evaluation as well as human evaluation. We then benchmark *Simple-SQuAD* using two different variants of RoBERTa

and perform an ablation study to investigate the effects of text simplification using four different variations of input. We prove that text simplification in the question-answering downstream task increases the predictive performance of the models. We further conduct edit-type analysis and sentence-length analysis to give insights about the transfer process. Future work may include improving style transfer performance using a more extensive corpus for text simplification and exploring effects of text simplification for other downstream tasks like text summarization, sentiment analysis.

References

Roee Aharoni and Yoav Goldberg. 2018. Split and rephrase: Better evaluation and stronger baselines. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 719–724, Melbourne, Australia. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Jan A. Botha, Manaal Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. 2018. Learning to split and rephrase from Wikipedia edit history. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 732–737, Brussels, Belgium. Association for Computational Linguistics.

John A. Carroll and Richard G. Macdonald. 1998. Practical simplification of english newspaper text to assist aphasic readers.

Dana Dannells. 2020. Automatic generation and simplification of written documents.

Richard Evans and Constantin Orasan. 2019. Sentence simplification for semantic role labelling and information extraction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 285–294, Varna, Bulgaria. INCOMA Ltd.

Eva Hasler, Adri de Gispert, Felix Stahlberg, Aurelien Waite, and Bill Byrne. 2017. Source sentence simplification for statistical machine translation. *Comput. Speech Lang.*, 45(C):221–235.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P. Xing. 2017. Toward controlled generation of text. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1587–1596. JMLR.org.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing - Volume 16*, PARAPHRASE '03, page 9–16, USA. Association for Computational Linguistics.

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

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Opensource toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1865–1874, New Orleans, Louisiana. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

T. Maruyama and K. Yamamoto. 2019. Extremely low resource text simplification with pre-trained transformer language model. In *2019 International Conference on Asian Language Processing (IALP)*, pages 53–58.

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 (Volume 1: Long Papers)*, pages 435–445, Baltimore, Maryland. Association for Computational Linguistics.

Shashi Narayan, Claire Gardent, Shay B. Cohen, and Anastasia Shimorina. 2017. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 606–616, Copenhagen, Denmark. Association for Computational Linguistics.

Christina Niklaus, Matthias Cetto, André Freitas, and Siegfried Handschuh. 2019. Transforming complex sentences into a semantic hierarchy. *ArXiv*, abs/1906.01038.

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 (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Kartikey Pant, Yash Verma, and Radhika Mamidi. 2020. Sentiinc: Incorporating sentiment information into sentiment transfer without parallel data. In *Advances in Information Retrieval*, pages 312–319, Cham. Springer International Publishing.

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 of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Martin Popel and Ondřej Bojar. 2018. Training tips for the transformer model. *The Prague Bulletin of Mathematical Linguistics*, 110.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

James Pustejovsky. 1991. The syntax of event structure. *Cognition*, 41:47–81.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language models are unsupervised multitask learners.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Advaith Siddharthan. 2002. An architecture for a text simplification system. In *Proceedings of the Language Engineering Conference (LEC'02)*, LEC '02, page 64, USA. IEEE Computer Society.

Sara Silveira and António Branco. 2012. Enhancing multi-document summaries with sentence simplification.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *ArXiv*, abs/1706.03762.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio. Association for Computational Linguistics.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and

Sandra Maria Aluísio. 2009. Facilita: Reading assistance for low-literacy readers. In *Proceedings of the 27th ACM International Conference on Design of Communication*, SIGDOC '09, page 29–36, New York, NY, USA. Association for Computing Machinery.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, page 409–420, USA. Association for 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*, 4:401–415.

Sanqiang Zhao, Rui Meng, Daqing He, Andi Saptono, and Bambang Parmanto. 2018. Integrating transformer and paraphrase rules for sentence simplification. *ArXiv*, abs/1810.11193.

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*, COLING '10, page 1353–1361, USA. Association for Computational Linguistics.