

Evaluation Metrics for Split and Rephrase

Stanford CS224N Custom Project (Grading Option: 3)

Rui Yan
Stanford University
ruiyan@stanford.edu

Mentor: Dulce Ponceleon, IBM Research

Abstract

Long and complicated sentences can often be difficult to understand for humans and machines. Split-and-rephrase task aims to make sentences easier to read by splitting a long sentence into shorter ones that are meaning preserving. However, there is no optimal evaluation metric for this task. In addition, there are only two existing datasets for model training. To address these problems, we first design a human evaluation tool that are reliable for evaluating split-and-rephrase output. We also develop a methodology for comparing the quality of datasets for training split-and-rephrase systems, which will facilitate future development for this task.

1 Introduction

Split-and-rephrase [1] is an natural language processing (NLP) task proposed by Narayan et al. in 2017. It aims to make sentences more readable through sentence splitting and rephrasing. It maps a complex sentence to a sequence of simple ones together convey the same meaning. Sentence splitting and rephrasing can benefit people with limited literacy skills, such as second-language speakers, children, and people with language impairments. It can also advance a number of NLP tasks such as Information Extraction.

Unlike some similar NLP tasks such as sentence compression or sentence simplification, there is no need of deletion or lexical simplification for split-and-rephrase. Instead, the systems must learn to split complex sentences into shorter ones and to make the syntactic transformations required by the split (e.g., turn a relative clause into a main clause). Table 1 compares four sentence rewriting tasks.

	Splitting	Deletion	Rephrasing	Meaning Preserving
Paraphrasing	✗	✗	✓	✓
Compression	✗	✓	✓	✗
Simplification	✓	✓	✓	✗
Split-and-rephrase	✓	✗	✓	✓

Table 1: Comparison of four similar sentence rewriting tasks regarding whether they have the following operations or properties: splitting, deletion, rephrasing, and meaning preserving.

Sentence split-and-rephrase is a relatively new task. There exist only two benchmarks for this task, WEBSPLIT [1] and WIKISPLIT [2]. In addition, this task does not have an optimal evaluation metric. All previous studies on split-and-rephrase task used BLEU as their evaluation metrics [1][3][2]. However, BLEU [4] is proved not suitable for the evaluation of text simplification [5], which suggests its lack of accuracy for simplicity evaluation. Hence, it is needed to introduce an optimal metric that is more reliable for split-and-rephrase output evaluation. To achieve this goal, we first design a human assessment tool on a crowdsourcing platform called Figure Eight (F8) to evaluate split-and-rephrase output based on four criteria, including grammaticality, readability, structural simplicity, and meaning preservation. Furthermore, we propose a new methodology to compare the quality of datasets for

training split-and-rephrase systems. This can be achieved by comparing the model predictions on the same test set when varying the training data, while holding model architecture fixed. LSTM-AG18, the previous best model by Aharoni and Goldberg (2018) [3], is used as our split-and-rephrase model architecture. The training data comes from the two benchmarks WEBSPLIT and WIKISPLIT, and the test set is from WEBSPLIT.

2 Approach

2.1 Task Formulation

The split-and-rephrase task can be formulated as a mathematical problem as follows. Given a complex sentence C , we want to generate a sequence of simplified sentences $S_1, \dots, S_n \in S$, where $n \geq 2$ and the meaning of C is preserved in S , such that the likelihood of S given C and model parameter θ , i.e., $P(S|C, \theta)$, is maximized.

An example for the task input and output is as shown in Table 2:

Input	John Clancy is a labor politician who leads Birmingham, where architect John Madin, who designed 103 Colmore Row, was born.
Output	Labor politician, John Clancy is the leader of Birmingham. <::::> John Madin was born in this city. <::::> He was the architect of 103 Colmore Row.

Table 2: Sample input and output for split-and-rephrase task. <::::> is sentence delimiter.

2.2 Our Model

The model implemented in our project is LSTM-AG18 model [3] proposed by Aharoni and Goldberg in 2018. LSTM-AG18 is a single-layer encoder-decoder model with bidirectional LSTM [6], with 512-dimensional embeddings. Its decoder uses the local-p attention model with feed input as in [7]. Furthermore, the model is augmented with a copy mechanism [8].

2.2.1 Copy-augmented Seq2Seq Attention Model

Previous studies [1][3][2] all tend to formulate this task into a seq2seq learning framework, which takes the complex sentence as input and sequentially generates a series of simple sentences. Seq2seq framework first uses a recurrent neural network (RNN) to convert a source sequence to a dense, fixed-length vector representation (encoder), and then uses another recurrent network (decoder) to convert that vector to a target sequence.

We train a vanilla seq2seq model with attention [9] as implemented in the OPENNMT-PY toolkit [10]. The model is augmented by a copy mechanism, which enables the model generate words by copying from the input sequence. More details about the copy mechanism can be found in [8]. The model architecture used in our work is shown Figure 1.

2.2.2 Parameters of Seq2seq Model

We choose to implement LSTM-AG18 because it is the state-of-the-art model for split-and-rephrase task as it outperforms all other existing models for this task by its BLEU score [3]. However, note that for this model, (1) the number of LSTM layers is 1; (2) LSTM layer is bidirectional; (3) the dimension of embeddings is 512. In this subsection, we study and analyze how these parameters would impact the performance of seq2seq models from the experimental results in [11].

Encoder and Decoder Depth: It is unclear how important depth (the number of LSTM layers) is, and whether shallow networks are able to produce similar results with those of deep networks. In [11], Britz et al. explored the effect of encoder and decoder depth up to 8 layers. For encoder RNNs, they found no evidence that depth beyond two layers is necessary. In fact, deeper encoder RNNs are more likely to diverge during training. For decoder RNNs, deeper models outperformed shallower ones by a small margin. Therefore, they argued that deep models are expected to perform better in

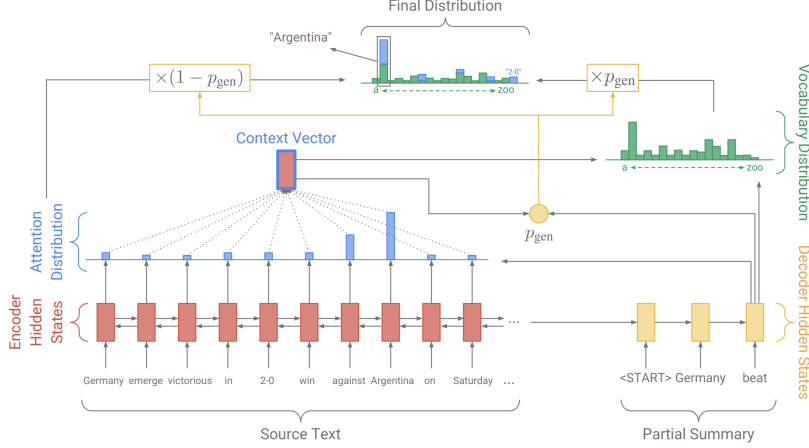


Figure 1: Sequence-to-sequence attentional model with a single-layer bidirectional LSTM augmented by copy mechanism. Source of Figure 1: [8].

general but might not differentiate the performance of shallow models too much. There is a need for more robust techniques.

Bidirectional RNNs: Bidirectional RNNs are only applicable if we have access to the entire input sequence. According to Britz et al., bidirectional RNNs generally outperform unidirectional encoders [11]. In practice, people usually use bidirectional RNNs when it is applicable.

Embedding Dimensionality: People expected better BLEU scores for larger embeddings. This was proved not always true by Britz et al.. They found that sometimes small 128-dimensional embeddings are compatible of producing results competitive with those of, say, 2048-dimensional embeddings. Moreover, small embeddings converge faster than large ones.

From the above analysis, we know that the following parameters: (1) the number of LSTM layers, (2) direction of LSTM layers, (3) embedding dimensions, would not disturb the model performance too much. However, considering that deeper model is usually expected to be more powerful, we can definitely increase the number of LSTM layers to see if we can achieve better BLEU score.

2.2.3 Model Training Details

We train the model with early stopping technique by computing its perplexity on the validation set after every epoch. We use a single layer LSTM for the encoder and decoder. We decode using beam search with a beam size of 12. All model parameters, including the embeddings are randomly initialized and learned during training. For optimization we use SGD with an initial learning rate of 1.0 and decay the learning rate by 0.5 when there is no improvement of perplexity on the validation set. GPU server is Nvidia Titan X (pascal) 12GB. OPENNMT-PY is adopted from the commit d4ab25a (<https://github.com/OpenNMT/OpenNMT-py>). The version of CUDA is 9.0.

We train this model to predict, given a complex sentence, the corresponding sequence of shorter sentences. The model is learned on pairs $\langle C, S \rangle$ of complex sentences and the corresponding text, where it directly optimizes $P(S|C; \theta)$. The code of the model is available on Github¹. *Although the code for the model implementation is available, it does not include any information about its software dependencies, which lead to a large amount of work to figure out the versions of packages they used in their code. We made many modifications on the original code to resolve the dependency conflict issues.*

2.3 Automatic Metrics

We implemented four automatic metrics for system output evaluation. The automatic metrics are (i) machine translation-inspired metric: BLEU [4], (ii) FKGL [12], (iii) #simple sentences per complex

¹<https://github.com/biu-nlp/sprp-acl2018>

sentence (#S/C) and #tokens per simple sentence (#T/S), (iv) SARI [13].

BLEU (BiLingual Evaluation Understudy) [4] is one of the most commonly-used metrics for language model. It was proposed by Papineni et al. in 2002 for evaluating the predictions made by machine translation systems. Here we use it as a metric to evaluate split-and-rephrase output because split-and-rephrase can be considered as translating a text from complex to simple. BLEU scores $\in [0, 100]$. *The higher the BLEU score, the better the predicted split-and-rephrase output consistent with gold standard references.*

The BLEU score is computed as in Equation (1). Traditionally, $N = 4$ and $w_n = 1/N$.

$$BLEU = BP \cdot \exp \left(\sum_{n=1}^N w_n \log(p_n) \right) \quad (1)$$

The modified precision score, p_n , for the entire test corpus is computed as in Equation (2). We first compute the n -gram matches sentence by sentence. We then add the clipped n -gram counts for all the candidate sentences and divide by the number of candidate n -grams in the test corpus.

$$p_n = \frac{\sum_{K \in Candidates} \sum_{n\text{-gram} \in K} Count_{clip}(n\text{-gram})}{\sum_{K' \in Candidates} \sum_{n\text{-gram}' \in K'} Count(n\text{-gram}')} \quad (2)$$

The BP (Brevity Penalty) factor used in BLEU computation is calculated as in Equation (3), where c is the length of the candidate split-and-rephrase corpus, r is the reference corpus length, and r/c is used in a decaying exponential.

$$BP = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - r/c) & \text{if } c \leq r \end{cases} \quad (3)$$

In research of text simplification, several studies argue that BLEU has high correlation with human assessments of grammaticality and meaning preservation, but not simplicity [13][14][15]. However, it is proved to have a low correlation with human assessment of simplicity when sentence splitting has been performed [5]. Therefore, BLEU is not supposed to be the only metric used for evaluation of split-and-rephrase output.

FKGL (Flesch-Kincaid Grade Level) [12] is designed to indicate how difficult a passage in English is to understand, in terms of structural simplicity. FKGL values correspond to grade levels in the United States. *Lower values indicate a higher level of structural simplicity.*

$$FKGL = 0.39 \left(\frac{\#words}{\#sentences} \right) + 11.8 \left(\frac{\#syllables}{\#words} \right) - 15.59 \quad (4)$$

FKGL measures structural simplicity assuming that the text is well-formed, and thus is insufficient alone as a metric for evaluating automatically generated sentences.

#S/C & #T/S (Average Sentence Length & Average Token Length) measures structural simplicity. #S/C measures the ability of a system to split a complex sentence into multiple simple sentences, and #T/S measures the ability of producing smaller simple sentences. *Higher measurements of these two quantities indicates higher level of structural simplicity.*

SARI (System Output Against References and Input Sentence) [13] measures how good the words added, deleted, and kept by a split-and-rephrase model are. SARI score is the arithmetic mean of n -gram precisions and recalls for add, keep, and delete. *Higher SARI score indicates better simplicity or readability.*

Considering a system output O , the input I and the references R . For addition, we award addition operations where system output O was not in the input I but occurred in any of the references R , i.e. $O \cap \bar{I} \cap R$. The n -gram precision $p(n)$ and recall $r(n)$ for addition operations are defined as follows:

$$p_{add}(n) = \frac{\sum_{g \in O} \min(\#_g(O \cap \bar{I}), \#_g(R))}{\sum_{g \in O} \#_g(O \cap \bar{I})} \quad (5)$$

$$r_{add}(n) = \frac{\sum_{g \in O} \min(\#_g(O \cap \bar{I}), \#_g(R))}{\sum_{g \in O} \#_g(R \cap \bar{I})} \quad (6)$$

where $\#_g(\cdot)$ is a binary indicator of occurrence of n -gram g in a given set. Here,

$$\#_g(O \cap \bar{I}) = \max(\#_g(O) - \#_g(I), 0) \quad (7)$$

$$\#_g(R \cap \bar{I}) = \max(\#_g(R) - \#_g(I), 0) \quad (8)$$

Similarly, we compute the n -gram precision $p_{keep}(n)$ and $p_{del}(n)$, and recall $r_{keep}(n)$ and $r_{del}(n)$ for word retaining and word deleting. Please refer to the corresponding paper for more details. Finally the SARI is computed as follows:

$$SARI = d_1 F_{add} + d_2 F_{keep} + d_3 F_{del} \quad (9)$$

where $d_1 = d_2 = d_3 = \frac{1}{3}$ and

$$P_{operation} = \frac{1}{k} \sum_{n=[1, \dots, k]} p_{operation}(n), R_{operation} = \frac{1}{k} \sum_{n=[1, \dots, k]} r_{operation}(n) \quad (10)$$

$$F_{operation} = \frac{2 \times P_{operation} \times R_{operation}}{P_{operation} + R_{operation}} \quad (11)$$

where $operation \in [add, del, keep]$

Unlike BLEU which only takes into account the similarity of the output with the references, SARI compares the output against the input sentence and references simultaneously. It rewards output that modify the input in ways that are expressed by the references. However, it also has many limitations, for example, the number of simplification operations considered is limited, i.e. only add, delete, and keep; also, SARI score works only if there are multiple reference sentences.

Given the formulas of metrics, we wrote our own implementation code for all the above metrics, except BLEU, which is available on Github². We then preprocessed our split-and-rephrase system output and used the above automatic metrics to evaluate the output.

2.4 Human Evaluation with Figure Eight

Our main approach is to design a human assessment tool to evaluate the split-and-rephrase results, and compare the human assessment with automatic metrics to study how well each metric reflects human judgement. We will implement our human assessment tool with Figure Eight (F8), a crowdsourcing platform where it asks human judges to do some simple tasks such as images annotating. Our tool is designed to measure our split-and-rephrase system output on four most important criteria, including grammaticality (G), meaning preservation (M), simplicity (S), and structural simplicity (StS).

Human judges were presented with a pair of a pair of sentences (the original and the simplification), and are asked to answer the questions as shown in Table 3.

²https://github.com/google-research/google-research/tree/master/wiki_split_bleu_eval

Criterion	Questions	Type of Questions
G	Is the simplified text grammatical correct?	Rating with a scale of 1-5
M	(1) Does the simplified text miss some facts that are present in the original text? (2) Does the simplified text have new facts that are not present in the original text?	YES / NO (1 / 0)
S	How much more easier to understand the simplified text is when compared with the original one?	Rating with a scale of 1-5
StS	Does the simplified text have one or more sentences that should be further split?	YES / NO (1 / 0)

Table 3: Questions for human judges to evaluate system output on grammaticality (G), meaning preservation (M), simplicity (S), and structural simplicity (StS).

3 Experiments

3.1 Data

Our datasets are WEBSPLIT [1] and WIKISPLIT [2]. Both datasets contain training/test/validation sets that include the original complex sentences C and the corresponding sequence of simplified sentences $S'_1, \dots, S'_n$. Their data formats are different so data preprocessing is required.

- WEBSPLIT:
 - Data collection: preprocessed all 13,308 distinct verbalisations contained in the WEBNLG corpus using the Stanford CoreNLP and then manually corrected the incorrect segmentations.
 - Data format: Complex and simple sentences stored in different files, e.g., complex sentences of training data stored in “train.complex”, simplified sentences of training data stored in “train.simple”.
 - Data size: See Table 4.
- WIKISPLIT:
 - Data collection: identified Wikipedia edits history that involve sentences being split. A list of sentences for each snapshot is obtained by stripping HTML tags and Wikipedia markup and running a sentence break detector.
 - Data format: Complex and simple sentences stored in the same file, e.g., train set in “train.tsv”. Complex sentence and its corresponding single sentences are concatenated together as a string. Single sentence segments are separated by “<::::>”.
 - Data size: See Table 4.

	WEBSPLIT		WIKISPLIT	
	Count	Unique	Count	Unique
C	1.3m	17k	1.0m	1.0m
S'	6.1m	28k	2.0m	1.9m

Table 4: Training corpus statistics in terms of complex sentences (C) and simple sentences for reference (S'). WIKISPLIT provides much greater diversity and scale. Also note that for each C , $|S'| = 2$ for WIKISPLIT and $|S'| \geq 2$ for WEBSPLIT.

3.2 Evaluation Method

Please refer to Sections 2.3 and 2.4 for the evaluation methods we are going to implement.

3.3 Experimental details

We trained LSTM-AG18 model over two datasets WEBSPLIT and WIKISPLIT, and then had the model simplify a set of test sentences in WEBSPLIT to obtain two sets of split-and-rephrase output.

After implementing our baseline automatic metrics and human assessment tools, we used them to evaluate our split-and-rephrase output and compared their scoring results. We showed that automatic metrics cannot accurately reflect human judgement on split-and-rephrase output. Thus, either a new and more reliable automatic metric, or the human evaluation tool itself should be used for evaluating split-and-rephrase.

3.4 Results

3.4.1 Model Predictions

The examples in Table 5 demonstrate several errors that occur in the system output of the model trained by WEBSPLIT and WIKISPLIT.

Input 1	A Fortress of Grey Ice with ISBM 0-7653-0633-6 has 672 pages.
WEBSPLIT output 1	A Fortress of Grey Ice has 672 pages. <::::> A Fortress of Grey Ice was made in “Print”.
WIKISPLIT output 1	A Fortress of Grey Ice. <::::> With ISBM 0-7653-0633-6 has 672 pages.
Input 2	Alfred Worden was a backup pilot on the NASA operated Apollo 12 on which David Scott the commander and Alan Bean, who spent a total of 100305 minutes in space, was a crewman.
WEBSPLIT output 2	Alan Bean was on the crew of Apollo 12. <::::> Alfred Worden was a backup pirlot of Apollo 12. <::::> The Apollo 12 operator is NASA. <::::> Apollo 12 was commanded by David Scott. <::::> Alan Bean spent 100305 minutes in space.
WIKISPLIT output 2	Alfred Worden was a backup pilot on the NASA operated Apollo 12 on which David Scott the commander and Alan Bean. <::::> He spent a total of 100305 minutes in space, was a crewman.
Input 3	11 Diagonal Street is located in South Africa where Jacob Zuma is a leader and some black people live.
WEBSPLIT output 3	11 Diagonal Street is located in South Africa. <::::> The leader of South Africa is Jacob Zuma. <::::> One of South Africa ’s leader is called Jacob Zuma. <::::> One of South Africa ’s leader is called Jacob Zuma.
WIKISPLIT output 3	11 Diagonal Street is located in South Africa. <::::> It is a leader and some black people live.

Table 5: Example model predictions on the WEBSPLIT test set from LSTM-AG18 trained by WEBSPLIT and WIKISPLIT training set, illustrating **unsupported facts**, **missing facts**, **repeated facts**, **grammar errors**, **insufficient splitting**. LSTM-AG18 model is the previous best model by Aharoni and Goldberg (2018).

3.4.2 Evaluation Results of Automatic Metrics

System Output	BLEU	FKGL	#S/C	#T/S	SARI
WEBSPLIT	30.72	8	2.55	16.09	43.86
WIKISPLIT	59.95	6	2.00	11.73	34.30

Table 6: Results on the predictions of WEBSPLIT test set when varying the training data (WEBSPLIT and WIKISPLIT training sets) while holding model architecture (AG18) fixed. Corpus-level BLEU (multiple references), Flesch-Kincaid Grade Level, #S/C: simple sentences per complex sentence, #T/S: tokens per simple sentence (micro-average), corpus-level SARI (multiple references).

Table 6 shows the evaluation results of our two training configurations on the test set of WEBSPLIT. As can be seen, compared to WEBSPLIT, the output of the model trained by WIKISPLIT has a higher BLEU score, lower FKGL and lower #T/S. By the construction of each metric, this indicates that the system output trained by WIKISPLIT is more grammatical, has higher structural simplicity and smaller word length of simple sentences. On the other hand, WEBSPLIT has a higher #S/C and SARI score. This indicates that the model trained by WEBSPLIT has a stronger ability to split a complex sentence to more simple sentences. Moreover, it is more readable.

3.4.3 Evaluation Results of Human Crowdsourcing

System Output	G	M	S	StS
WEBSPLIT	3.58	0.63	3.8	0.79
WIKISPLIT	4.27	0.75	3.35	0.68

Table 7: Human evaluation results over the simple text predicted by LSTM-AG18 for a random sample of 30 inputs from WEBSPLIT test set. G represents *grammaticality*; M represents *meaning preservation*; S represents *simplicity*; StS represents *structural simplicity*.

Table 7 shows the average human ratings over our system output. The score for grammaticality and simplicity range from 1 to 5 as they correspond to 1-5 scale rating problems in Table 3; the score for meaning preservation and structural simplicity range from 0 to 1 as they correspond to YES / NO questions in Table 3. According to Tables 6 and 7, we show BLEU can somehow capture the human judgment on grammaticality and meaning preservation, but not on structural simplicity and simplicity. #S/C and SARI are correlated with structural simplicity and simplicity, respectively, to some extent. However, #S/C and SARI show nothing about grammaticality and meaning preservation. FKGL and #T/S are not correlated with the human judgement on either structural simplicity or simplicity for split-and-rephrase output. Therefore, we conclude that BLEU cannot be used as the only metric for split-and-rephrase evaluation. We can probably build a new metric combining BLEU and SARI such that it has a high correlation with human judgement on split-and-rephrase. However, the most reliable metric should still be the human evaluation tool we built in our paper, regardless of its higher cost compared to automatic metric.

3.4.4 Analysis

Based on the results of human ratings, we can compare the quality of WEBSPLIT and WIKISPLIT for training LSTM-AG18. Obviously, the model trained by WEBSPLIT performs better in sentence splitting and readability, while the model trained by WIKISPLIT performs better in meaning preservation and grammaticality. This result is mainly caused by the nature of training data. For WEBSPLIT, since its vocabulary size is not large enough, it may produce repeated or meaningless words when there are unseen words in complex sentence for prediction, which results to less meaning preservation. For WIKISPLIT, since each complex sentence in its dataset only corresponds to two simple sentences, it cannot learn how to split a complex sentence to more than two sentences, which is problematic. For model predictions, this would either result to over-splitting when a complex sentence is too short, or insufficient splitting when a complex sentence is too long.

4 Conclusion

In this paper, to address the limitation of BLEU for split-and-rephrase evaluation, we propose a human evaluation tool focusing on four aspects, grammaticality, simplicity, structural simplicity, and meaning preservation. We also introduce a methodology to compare the quality of datasets for training a split-and-rephrase model, which is built based on our human evaluation tool. From the results of our human ratings, both WEBSPLIT and WIKISPLIT are not good datasets for training a split-and-rephrase model. WEBSPLIT has limited vocabulary size and WIKISPLIT has problematic sentence splitting limitation.

References

- [1] Shashi Narayan, Clair Gardent, Shay B. Cohen, and Anasrasia Shimorina. Split and rephrase. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.
- [2] Jan A. Botha, Manaai Faruqui, John Alex, Jason Baldridge, and Dipanjan Das. Learning to split and rephrase from wikipedia edit history. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [3] Roei Aharoni and Yoav Goldberg. Split and rephrase: Better evaluation and a stronger baseline. In *Proceedings of the 56th Annual Meeting on Association for Computational Linguistics (ACL)*, 2018.
- [4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL)*, 2002.
- [5] Elior Sulem, Omri Abend, and Ari Rappoport. Bleu is not suitable for the evaluation of text simplification. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2018.
- [6] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. In *Neural computation*, 1997.
- [7] Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention based neural machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2015.
- [8] Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.
- [9] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.
- [10] Guillaume Klein, Yoon Kim, Yuntian Deng, Senellart Jean, and Alexander Rush. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, 2017.
- [11] Denny Britx, Anna Goldie, Minh-Thang Luong, and Quoc Le. Massive exploration of neural machine translation architectures. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.
- [12] J. P. Kincaid, R. P. Fishburne, R.L. Rogers, and B. S. Chissom. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. In *Chief of Naval Technical Training: Naval Air Station Memphis*, 1975.
- [13] Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. Optimizing statistical machine translation for text simplification. In *Transactions of the Association for Computational Linguistics (TACL)*, 2016.
- [14] Sander Wubben, Antal van den Bosch, and Emiel Krahmer. Sentence simplification by mono-lingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2012.
- [15] Sanja Štajner, Ruslan Mitkov, and Horacio Saggion. One step closer to automatic evaluation of text simplification systems. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)*, 2014.