Is Character Trigram Overlapping Ratio Still the Best Similarity Measure for Aligning Sentences in a Paraphrased Corpus?

Aleksandra Smolka
SNHCC TIGP
Institute of Information Science
Academia Sinica
aleksandra.smolka@hotmail.com

Hsin-Min Wang
Institute of Information Science
Academia Sinica
whm@iis.sinica.edu.tw

Jason S. Chang
Department of Computer Science
National Tsing Hua University
jason@nlplab.cc

Keh-Yih Su
Institute of Information Science
Academia Sinica
kysu@iis.sinica.edu.tw

Abstract

Sentence alignment is an essential step in studying the mapping among different language expressions, and the *character trigram* overlapping ratio was reported to be the most effective similarity measure in aligning sentences in the text simplification dataset. However, the appropriateness of each similarity measure depends on the characteristics of the corpus to be aligned. This paper studies if the character trigram is still a suitable similarity measure for the task of aligning sentences in a paragraph paraphrasing corpus. We compare several embedding-based and non-embeddings model-agnostic similarity measures, including those that have not been studied previously. The evaluation is conducted on parallel paragraphs sampled from the Webis-CPC-11 corpus, which is a paragraph paraphrasing dataset. Our results show that modern BERT-based measures such as Sentence-BERT or BERTScore can lead to significant improvement in this task.

Keywords: sentence alignment, sentence similarity, sentence embedding

1 Introduction

Monolingual text matching is necessary for many downstream applications, such as Paraphrase Identification and Extraction (Qiu et al., 2006), Question Answering (Weiss et al., 2021), Natural Language Inference (MacCartney et al., 2008), and Text Generation (Barzilay and McKeown, 2005). Take the QA task as an example, identifying the text fragments that match the given question within the associated passage is often required for locating the desired answer.

However, modern neural-network (NN) approaches to text matching often suffer from certain limitations when two sequences contain considerably different lexicons or diverse grammatical structures (McCoy et al., 2019). For example, when the verb "*decide"* in the sentence "*They **decided** to go*" is nominalized to the noun "*decision"* in its paraphrase "*They **made a decision** to go*", the popular word embedding similarity approach might fail as the embedding-vectors of "*decide*" and "*decision*" are quite different [1]. Another example is a pair of sentences "*A cat is chasing a dog.*" and "*A dog is chasing a cat.*", which contain the same set of lexicons and syntactic structure but with opposite meanings.

Furthermore, the NN approaches frequently fail while the matching involves multi-word expressions, or when expressions require compositionality handling (Blevins et al., 2018; Hupkes et al., 2020; Zhou et al., 2020). For example, it is difficult to match expressions "*put off*" and "*procrastinate*" using basic word embeddings, as the real meaning of the idiom "*put off*" is not the sum of the meanings of its tokens.

We found that the limitations of NN models in text matching could be greatly alleviated by utilizing lexico-syntactic paraphrasing patterns such as $[_{VP}[_{VBN}[see]_{NP}[X_1]]] \rightarrow [_{S}[_{NP}[X_1]_{VP}[_{VBD}[be]$

[1] The nearest semantic associates of the verb *decide* based on the cosine similarity between the word2vec vectors (trained on English Wikipedia) are those verbs such as: *choose* (0.64), *opt* (0.62), *persuade* (0.61), *want* (0.58), *refuse* (0.57), *insist* (0.56). However, the noun *decision* only has a similarity score 0.512, which means that its similarity to the verb *decide* is even less than that between *decide* and its quasi-antonymous *refuse*.

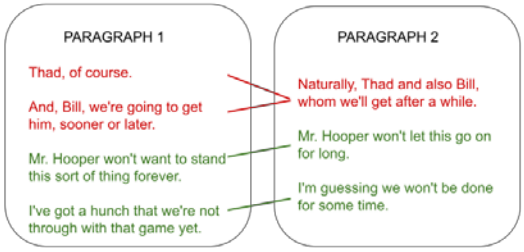

Figure 1: Sentence alignment for extracting paraphrased sentence-pairs. Sentences pairs in green are those we want to extract; sentences in red are in multi-to-one relation and do not constitute sentential paraphrases.

$_{VP}$*[observe]]]*[2], which denotes the conversion from active to passive voice for the phrase pair "*see the lion*" and "*the lion was observed*". Since some key lexicons are involved in the pattern, it would be difficult to exhaustively list such patterns by a human. It is preferable to automatically extract them from a large paraphrase corpus.

To collect such lexico-syntactic patterns, a high-quality paraphrased sentence-pair dataset is essential. Unfortunately, current *sentence-aligned* paraphrase datasets (e.g., MRPC (Dolan and Brockett, 2005), PPDB (Ganitkevitch et al., 2013), and QQP (Aghaebrahimian, 2017)) are too trivial for this task, as they mainly contain lexical paraphrases that could be easily handled by a NN. On the other hand, some *paragraph-aligned* paraphrase corpora, containing different human translations from the same source text, fit our needs well. To utilize those paragraph-aligned paraphrase corpora, monolingual sentence alignment is the first step in retrieving the desired patterns.

Figure 1 shows how a correct sentence alignment could help extract paraphrased sentence pairs from longer paraphrased texts. Unless we correctly identify which sentences are in 1-to-1 relationships (green in the figure), we cannot correctly identify the desired paraphrased pattern.

Monolingual sentence alignment approaches could be classified into two categories: *model-based* approaches (e.g., Jiang et al., 2020), which adopt specific models to encode the input sentences and perform alignment, and *model-agnostic* approaches (Štajner et al., 2018), which can be directly applied to the selected dataset, without the necessity of training a neural model in advance. In our work, we focus on model-agnostic approaches, as they do not require additional labeled data to train the model.

The downside of previous model-agnostic approaches (Štajner et al., 2017; 2018) is that they only test the early word2vec word embeddings, and do not explore those more advanced NN approaches such as Sentence-BERT (Reimers and Gurevych, 2019) and BERTScore (Zhang et al., 2020). Also, they are mainly evaluated on Text Simplification (TS) datasets, which are different from our paraphrasing datasets.

In the TS dataset, the original and the simplified text often share a considerable number of keywords, which remain unchanged and are rarely substituted with synonyms. However, this property does not hold in our paraphrasing corpus, as its paraphrasing expressions usually possess diverse syntactic structures with many different lexical items.

Therefore, we suspect that the character trigram overlapping ratio, reported as the best for monolingual sentence alignment in previous works (Štajner et al., 2017; 2018), would not perform best on our data. Since our paraphrasing corpus contains considerably different lexicons and word order, the string-based method such as character ngram similarity would lose its edge. Previously reported text similarity measures thus should be re-evaluated for our task, and more advanced NN approaches should be explored.

In this work, we not only compare various previously reported text similarity measures on a paraphrased paragraph corpus but also additionally test some new measures based on the most recent NN sentence embedding methods. We utilize those above measures with two sentence alignment approaches: simple greedy match (e.g., Štajner et al. 2018), and sequence match (Gale and Church, 1993; Barzilay and McKeown, 2001). We conduct the evaluation on a manually annotated sentence-aligned dataset with 400 paraphrased paragraph pairs randomly sampled from the multiple translation corpus Webis-CPC-11 (Burrows et al., 2013).

Our contributions include:

(1) To the best of our knowledge, we present the first study on aligning sentences on a paragraph paraphrased corpus;

(2) We show that character trigram similarity is not the best measure for aligning

[2] The structure is annotated in bracketed form analogically to phrase-parsing annotation and X_i, i =*1,2,*… marks matching variables. We use the same tagset as that adopted in Penn Treebank (Marcus et al., 1993)

paraphrasing corpora. Instead, BERT-based embedding methods achieve significantly better results even without fine-tuning on the target dataset;

(3) We test several NN-related sentence similarity measures (other than word2vector) that have not been evaluated before for model-agnostic monolingual sentence alignment;

(4) We confirm and expand the observation of Choi et al., (2021), showing that [CLS] token representation is not necessarily superior to averaging individual word vectors for sentence representation while aligning paraphrased text under BERT.

2 Sentence Alignment Procedure

Our sentence alignment procedure is implemented with two main elements: (1) selecting an appropriate search mechanism (either *Bi-Directional Best Match* or *Sequence Match*); (2) adopting a specified sentence similarity measure, either string- or embedding-based.

2.1 Search Mechanisms

We adopt two approaches to conduct sentence alignment: Directional Best Match and Sequence Match.

2.1.1 Bi-directional Best Match

This is a simple greedy approach that ignores the adjacency and dependency information within sentences during matching. We adopt an approach similar to that reported in Štajner et al. (2018). However, in addition to *Uni-directional* Best Match adopted by Štajner et al. (2018), we also test *Bi-directional* Best Match, where we align the sentences bi-directionally. We believe that the bi-directional approach will be more applicable in our case since our data is symmetric, while the data tested in Štajner et al. (2018) is not.

In both versions, we take two sets of sentences as the input and calculate the similarity of each sentence pair that can be formed between these two sets. Based on the sentence similarity scores, for each sentence in one set, we select the sentence from the second set that possesses the highest similarity score, forming a set of sentence pairs. In the uni-directional version, those pairs are directly selected as the final alignments.

In contrast, for the bi-directional approach, we additionally repeat the same selection procedure from the opposite direction for each sentence in the second set to form another set of sentence pairs. Afterward, we take the intersection of these two sets to obtain the final aligned sentence pairs.

2.1.2 Sequence Match

Based on the selected similarity measure, this approach adopts dynamic programming to find out the best alignment sequence among the sentences within the given paragraph pair (Gale and Church, 1993; Barzilay and McKeown, 2001).

2.2 Similarity Measures

The text similarity measures adopted in our experiments fall into two main categories: (a) string-based approaches, in which the similarity is calculated purely based on the sentence strings; (b) embedding-based approaches, in which a neural model is first used to convert each sentence into its corresponding embedding-vector, and then the cosine similarity between these two sentence embedding-vectors is taken as the sentence similarity.

2.2.1 String-Based Sentence Similarity

We adopt two different overlapping ratios: (1) *Character ngram*, which is reported as the state-of-art on the text simplification corpus (Štajner, 2018), and (2) *token string*, which is commonly used in sentence alignment tasks (e.g., Barzilay and McKeown, 2001).

Character Ngram

We follow Štajner et al. (2018) to calculate the ngram similarity based on the *Character Ngram Similarity* model with tf-idf weighting (adapted from McNamee and Mayfield (2004)). We experiment with different ngram sizes (1 to 5) and use NGRAM to refer to this measure. We add Laplace smoothing to account for those unseen ngrams in the test set. The final similarity is calculated by taking cosine similarity.

Token String

For calculating token-based sentence similarity, we use the following token overlap formula:

$$similarity_{token} = \frac{|tokens_1 \cap tokens_2|}{|tokens_1|+|tokens_2|} \quad (1)$$

where $tokens_1$ is the set of tokens in the first sentence, $tokens_2$ is the set of tokens in the second sentence, and the function | | specifies the cardinality of the token set. We consider two

different normalization mechanisms for comparing two tokens: (1) converting the strings into their associated lemmas before comparison (abbreviated as TOKENstring); (2) also taking synonyms as exactly matched lemmas during comparison (abbreviated as TOKENsyn). Token lemmas for each sentence are retrieved using an automatic tokenizer and lemmatizer (Qi et al., 2020). Synonymic relationships are taken from WordNet (Fellbaum, 1998).

2.2.2 Embedding-Based Sentence Similarity

We adopt three different approaches to calculate the similarity score between two sentences: (1) *word-embedding* based, where we first look up the word embedding-vector for every token in each sentence from a pretrained model and then combine them into their associated sentence embedding-vector by vector averaging (Putra and Tokunaga, 2017). Afterward, we calculate the similarity between the two obtained sentence embedding vectors. (2) *sentence-embedding* based, where we use a model, such as BERT (Devlin et al., 2019) or Sentence-BERT (Reimers and Gurevych, 2019), to directly embed a sentence into its associated sentence-embedding. We then calculate the similarity between these two sentence embedding vectors. (3) *BERTScore* (Zhang et al., 2020), which uses BERT to directly generate the similarity value between two sentences.

Word-embedding Similarity

For directly retrieving the token-associated embedding vector from a pretrained embedding lookup table, we test both word2vec (Mikolov et al., 2013) and Glove (Pennington et al., 2014) embeddings. Additionally, we also test contextualized word embeddings retrieved from BERT (Devlin et al., 2019).

Moreover, while it is common to use the [CLS] token yielded by the BERT encoder to represent the whole encoded sentence, recent works note that this might not be the best solution for all downstream tasks (Choi et al., 2021). We therefore additionally test the following approach: generate the sentence embedding via averaging the contextual word embeddings retrieved from the BERT model.

Regardless of the way of selecting word embedding, we combine the associated embedding vectors into the corresponding sentence representation by taking an average over them (Putra and Tokunaga, 2017). The sentence similarity is then calculated as the cosine similarity between the two sentence embedding vectors.

Among various types of word-embedding, only Word2vec is tested by Štajner et al. (2018). But it was reported not the best one in their experiments (the best one is character trigram in their task).

Sentence-embedding Similarity

Another way to generate the sentence-embedding is to adopt BERT to transform all its associated token-embeddings into it. We test two methods of obtaining sentence representation via BERT. First, we take the [CLS] token from the BERT to represent the whole sentence. Alternatively, we use Sentence-BERT (Reimers and Gurevych, 2019), which is an alternative method of obtaining sentence representation from BERT-type models, suggested as a better alternative for directly adopting [CLS] token embedding. We use Sentence-BERT to separately obtain a single embedding for each sentence in the pair. The sentence similarity is then calculated between two obtained sentence embedding vectors.

BERTScore

Last, we can directly generate the desired similarity value among two sentences by adopting the BERTScore (Zhang et al., 2020) approach, which is originally developed as an automatic evaluation metric for comparing various text generation systems. This approach first uses BERT to obtain the word embeddings of all input tokens. The pairwise similarity is then calculated for each possible token pair. Afterward, for each token from the first input sequence (i.e., the sentence from the "*original*" paragraph), BERTScore finds its matching token in the second sequence (i.e., the sentence from the "*paraphrased*" paragraph) via greedy search. Last, it calculates both precision and recall based on the matching result.

As BERTScore is designed to evaluate the similarity between the ground truth and the generated text, we thought it should be also suitable for measuring the sentence similarity for our task. Typically, BERTScore will report precision, recall, and f1-score at the same time. We take each of these values to represent a specific sentence pair similarity measure; and we refer to them as BERTprec, BERTrec, and BERTf1, respectively.

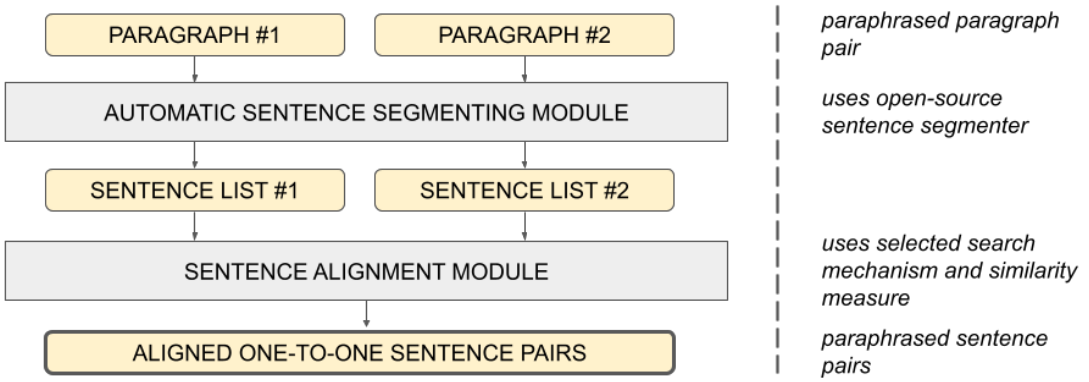

Figure 2: Operation flow for obtaining one-to-one sentence alignment within paraphrased paragraph pairs.

3 Experiments

Figure 2 shows the operation flow adopted in the experiments. Taking a pair of paraphrased paragraphs as input, the paragraphs are first preprocessed and split into sentences. Then, we use the sentence alignment module with the selected search mechanism and similarity measure to generate the desired sentence alignments. Those one-to-one sentence alignments are then extracted and output as the answer.

Following subsections give details of the experiment setting and results.

3.1 Dataset

We randomly sampled 400 paragraph pairs from the Webis-CPC-11 corpus (out of which 7 were found to be incorrectly marked as paraphrases and removed from the evaluation data). However, for checking if we can automatically detect if the given paragraph pair is a paraphrased one, we still reserve them as additional data on which we can experiment with a method of filtering out such undesired input.

As all tested similarity measures are model-agnostic, we do not require a training set. Therefore, we split all the aligned paragraph pairs (i.e., excluding those non-paraphrased pairs) into the development set and the test set with a 1:7 ratio. As a result, we end up with 48 paragraph pairs in the development set and 345 paragraph pairs in the test set. We use the development set for selecting hyper-parameters such as similarity cutting threshold and alignment type probabilities for the Gale-Church algorithm (Gale and Church, 1993).

Table 1 gives the associated dataset statistics. Within them, 566 1-to-1 paraphrased sentence pairs (77% among all aligned passage pairs) exist in the test set. This set of 1-to-1 sentence pairs (i.e., sentential paraphrases) is the desired output in our task and the ground truth for our evaluation.

3.2 Pre-processing

Because the Webis-CPC dataset only contains un-segmented paragraphs, it must be first converted into a collection of sentences. We use an off-the-shelf sentence segmenter (Qi et al., 2020) to split each paragraph into sentences. The output is thus two sets of sentences, one for each of the paragraphs.

3.3 Experiment Settings

Since the character trigram is reported as the best measure by Štajner et al. (2018), and no easily applicable code is released, we re-implement it as our baseline. The character ngram similarity is calculated as described by Štajner et al. (2018), including the tf-idf weighting adopted in the

	all	dev	test
#input paraphrased paragraph-pairs	393	48	345
#input non-paraphrased pairs (dataset errors)	7	2	5
avg. paragraph length (#sentences)	2.3	2.4	2.3
paragraph range (# sentences)	1-7	1-6	1-7
avg. sentence length (#tokens)	20.9	19.3	21.1
# 1-1 alignments (ground truth)	633 (77%)	67 (77%)	566 (77%)

Table 1: Dataset Statistics (without non-paraphrase cases). #Min-#Max specifies the range.

measure	% on the test set			Best TH
	prec	rec	F1	
NGRAM(n=1)*	77.8	82.2	79.9	0.3
NGRAM(n=2)*	77.8	82.2	79.9	0.3
NGRAM(n=3)	79.9	72.5	76.1	0.3
NGRAM(n=4)	77.8	82.2	79.9	0.3
NGRAM(n=5)	77.8	82.2	79.9	0.3
TOKENstring*	83.7	73.1	78.1	0.2
TOKENsyn	77.1	71.5	74.2	0.1
W2V	79.7	74.5	77.0	0.8
GLOVE	73.5	81.2	77.1	0.95
BERTword*	78.5	**87.0**	**82.5**	0.75
BERTcls	81.9	67.9	74.3	0.9
SBERTbert	75.2	90.8	82.3	0.6
SBERTalbert	82.9	70.7	76.9	0.35
SBERTmini*	78.4	85.2	81.6	0.6
BERTprec*	86.5	72.9	79.1	0.9
BERTrec*	83.5	74.9	80.4	0.9
BERTf1	**86.8**	74.9	80.4	0.9

Table 2: Alignment results for the Uni-directional Best Match strategy across all similarity measures. TH is the threshold value, selected on the development set based on the f1 value for each measure. The asterisk * marks the metrics that outperforms NGRAM baseline (n=3) with $p \leq 0.05$.

measure	% on the test set			Best TH
	prec	rec	F1	
NGRAM(n=1)	80.5	81.8	81.1	0.3
NGRAM(n=2)	80.5	81.8	81.1	0.3
NGRAM(n=3)	78.9	87.0	82.7	0.1
NGRAM(n=4)	80.5	81.8	81.1	0.3
NGRAM(n=5)	80.5	81.8	81.1	0.3
TOKENstring	84.7	73.1	78.5	0.2
TOKENsyn	78.6	81.8	80.2	0.05
W2V	81.1	87.6	84.2	0.6
GLOVE	79.7	78.0	78.8	0.95
BERTword	82.3	86.4	84.3	0.75
BERTcls	**86.2**	66.5	75.1	0.9
SBERTbert	79.1	88.6	83.6	0.6
SBERTalbert	80.6	89.8	84.9	0.25
SBERTmini*	80.7	90.2	**85.1**	0.25
BERTprec	80.9	88.2	84.4	0.85
BERTrec	79.7	88.2	83.7	0.85
BERTf1	79.9	**90.8**	85.0	0.9

Table 3: Alignment results for the Bi-directional Best Match strategy across all similarity measures. TH is the threshold value, selected on the development set based on the F1 value for each measure. The asterisk * marks the metrics that outperforms NGRAM baseline (n=3) with $p \leq 0.05$.

original work. We do not test our implementation on the original data adopted by them, as they only used human evaluation, without indicating which dataset was used for evaluation. Therefore, directly verifying our implementation with their results is impossible.

When experimenting with various search mechanisms, we additionally impose similarity score thresholding, which filters out those obtained 1-1 sentence pairs with their similarities below the specified threshold. The threshold value is selected for each similarity measure separately, based on the development set results.

For the approach of adopting [CLS] for sentence representation, we use a pretrained BERT-base model (Devlin et al., 2019). For the Sentence-BERT approach, we test three different pretrained versions released by an open resource[3]: BERT (Devlin et al., 2019; abbreviated as SBERTbert), ALBERT-mini (Lan et al., 2020; abbreviated as SBERTalbert), and MiniLM (Wang et al., 2020; abbreviated as SBERTmini). Among them, SBERTbert is trained with various *Natural Language Inference* data sets; in contrast, the last two versions are trained on various paraphrasing datasets [4]. The pre-trained model used for calculating the BERTScore is ROBERTA-Large (Liu et al., 2019).[5]

3.4 Various Experiments

We measure precision, recall, and F1-score for the two alignment strategies with various similarity measures. Furthermore, we use the McNemar test (Dietterich, 1998) to check if a given configuration (i.e., the adopted search mechanism and the specified similarity measure) yields significantly different results from the baseline (taking $p \leq 0.05$ as the significance test threshold).

We test the following measures: (A) **String-based** similarities: including character ngram similarity with *n* from 1 to 5 (NGRAM), and token overlap similarity calculated with either token strings (TOKENstring) or token synonyms (TOKENsyn); (B) **Embedding-based** similarities: (1) word embedding-based similarities calculated with word2vec (W2V), Glove (GLOVE) and BERTbase (BERTword) embeddings; (2) sentence embedding-based similarity: (i) using [CLS] token yielded by BERTbase model (BERTcls), and (ii)

[3] https://huggingface.co/sentence-transformers
[4] The list of specific datasets used was not published by the open-source authors.
[5] https://github.com/Tiiiger/bert_score

measure	% on the test set			Best TH
	prec	rec	F1	
NGRAM(n=1)*	89.1	83.4	86.1	0.2
NGRAM(n=2)*	89.1	83.4	86.1	0.2
NGRAM(n=3)	89.7	84.2	86.9	0.1
NGRAM(n=4)*	89.1	83.4	86.1	0.2
NGRAM(n=5)*	89.1	83.4	86.1	0.2
TOKENstring	**92.7**	81.6	86.8	0.15
TOKENsyn	86.2	86.9	86.3	0
W2V	87.6	87.6	87.6	0.45
GLOVE	87.3	85.2	86.2	0.9
BERTword	91.5	82.2	86.6	0.75
BERTcls	92.3	81.4	86.5	0.85
SBERTbert*	89.8	**87.8**	**88.8**	0.6
SBERTalbert	91.1	85.8	88.3	0.25
SBERTmini	87.8	86.8	87.3	0.25
BERTprec	90.0	86.8	88.4	0.85
BERTrec*	89.9	87.6	88.7	0.85
BERTf1*	90.1	87.4	88.7	0.85

Table 4: Alignment results for the *Sequence Match* strategy across all similarity measures. *TH* is the threshold value, selected from the development set based on the F1 value for each measure. The asterisk * marks the metrics that outperforms NGRAM baseline (n=3) with $p \leq 0.05$.

Sentence-BERT embeddings with three different pretraining models (SBERTbert, SBERTalbert, and SBERTmini); (C) **BERTScore** with precision (BERTprec), recall (BERTrec) and F1-score (BERTf1).

Tables 2-4 compare all similarity measures under the *Best Match* (*Uni-* and *Bi-directional*, separately) strategy and the *Sequence Match* strategy, respectively. For each measure, we only report the results with the best threshold value, which is selected on the development set based on the F1 value. The threshold for each specific similarity measure is different and is noted in the corresponding table. Measures that outperform the character trigram baseline in a significant manner are marked with the asterisk *.

Overall, comparing the best result of each approach, the sequence match approach (with the best F1-score equaling 88.8%) outperforms both best match approaches (the best F1-score of 85.1% is from the bi-directional mode). We conjecture the sequence match performs the best as it additionally considers the adjacency and dependency information within sentences during matching.

Moreover, the Uni-directional Best Match approach performed the worst (only with 82.5% best F1) as expected. Since our data is symmetric, the matching results would be more reliable if the alignment is considered from both directions.

measure	mean	L-CI (0.95)	#pairs
NGRAM(n=3)	0.547	0.530	5
TOKENstring	0.221	0.214	4
TOKENsyn	0.141	0.136	4
SBERTbert	0.541	0.522	3
SBERTalbert	0.411	0.391	3
SBERTmini*	0.339	0.321	6
BERTprec*	0.914	0.911	**7**
BERTrec	0.917	0.914	5
BERTf1*	0.915	0.913	5

Table 5: Results of filtering out non-paraphrased paragraph pairs based on the 0.95 confidence interval. *Mean* is the mean similarity value for all (393) paraphrased paragraph pairs; *L-CI* is the left boundary of the Confidence Interval, and *#pairs* is the number of non-paraphrased pairs that fall outside the confidence interval (out of 7). Results with $p \leq 0.05$ are marked with the asterisk *.

Furthermore, the best similarity measure varies under different search mechanisms. In the sequence match approach, three BERT-type measures (i.e., SBERTbert (88.8% F1), BERTrec (88.7% F1), and BERTf1 (88.7% F1)) significantly outperform the baseline. The SentenceBERT measure performs best, surpassing the character-trigram baseline method by 1.9% (88.8% vs. 86.9%) because it is trained to encode the overall sentence meaning, not the specific meaning of individual tokens**,** which fits our task well. Similarly, BERTScore also delivers good results because it is directly trained to measure the similarity between two sequences.

On the other hand, in the bi-directional best match approach, the best result is again obtained by the Sentence-BERT measure (SBERTmini) with the best F1-score 85.1%, significantly outperforming the character ngram similarity measure at 82.7%. Also, both SBERTalbert and BERTf1 measures outperform the baseline with $p<0.06$. We believe that the above reasons given for the sequence match approach also apply here.

Last, in the uni-directional best match approach, several tested measures significantly outperform the baseline (76.1%), including BERTword (82.5%), SBERTbert (82.3%), SBERTmini (81.6%), BERTf1(80.4%), NGRAM with $n \neq 3$ (79.9%), BERTrec (79.7%), BERTprec (79.1%) and TOKENstring (78.1%). The measures that perform best in this search mechanism are again mostly those that encode the sentence as a whole, similar to other search mechanisms.

We additionally note that in both versions of the Best Match approach, BERTword is significantly better (84.3% and 82.5% for bi- and uni-directional, respectively) than that is calculated with the [CLS] token embedding (BERTcls, 75.1%, and 74.3%). This is in line with the observation from Choi et al. (2021), who noted that interpreting the [CLS] token embedding as the sentence representation might be inferior to combining the individual sub-word embeddings obtained from BERT.

3.5 Exploring Features for Non-paraphrased Paragraph-pair Detection

Since the Webis-CPC-11 paraphrasing dataset is found to contain some non-paraphrased paragraph pairs (a total of 7 pairs are found among 400 pairs sampled), we also want to check if it is possible to automatically detect those outliers. As the paragraph is just a longer passage in comparison with the sentence, we expect that the measures adopted to calculate the sentence similarity could be also applied to evaluate the paragraph similarity. We thus further test whether the measures adopted for sentence alignment are discriminative enough to filter out those incorrectly annotated paragraph pairs (i.e., non-paraphrased pairs found).

We calculate paragraph similarity via the same approaches conducted for evaluating the sentence similarity and test some similarity measures which perform better for the sentence case (including Sentence-BERT, BERTScore, etc.). We fit the similarity values from all paraphrased paragraph pairs for each measure with specific normal distribution and then calculate its 0.95 confidence interval to check whether the non-paraphrased paragraphs can be detected as outliers outside this interval.

Table 5 shows the left boundary value of the 0.95 Confidence Interval as well as the number of non-paraphrased paragraph pairs (out of 7 in the data) that fall below this interval. We found that all non-paraphrased paragraphs can be detected as outliers and filtered out using BERTprec (with the nearest outlier sitting at p=0.01). It thus confirms the feasibility of adopting BERTprec for automatically filtering out those annotation errors.

4 Error Analysis

We analyzed 50 errors generated by our best approach (i.e., Sequence Match with SBERTmini), and categorized them based on their associated error sources: (1) mistaking 1-n mapping for 1-1 (46%); (2) associated with incorrect sentence boundary (26%), in which the sentences are split incorrectly before conducting alignment (e.g., a sentence is incorrectly split into two sequences by the sentence segmenter); (3) paraphrased sentences take different sequence-orders within two given paragraphs (16%); (4) others (12%), of which it is difficult to attribute each error to a specific reason.

The first error category, incorrectly marking 1-n alignment as 1-1, is likely due to two reasons. First, those proposed similarity measures are still incapable of truly reflecting the semantic similarity between two sentences when they are paraphrased in an abstract way; as a result, they might incorrectly convert a golden 1-n mapping into a 1-1 mapping. Second, because the alignment is selected based on the sentence similarity and the probability of each alignment type on the development set, the model has a preference for extracting 1-1 alignments as they are most common in the dataset (cf. Table 1).

The second error category (i.e., with incorrect sentence boundary) occurs when the pre-processing module incorrectly split the sentences within one of the input paragraphs. Finally, the last type of error is caused by the sequence search mechanism, which assumes all paraphrased passage pairs follow the same relative order within each paragraph. If this assumption is violated in the given paragraph pair, it will always return an incorrect answer.

5 Related Work

Sentence Alignment Mechanisms

Works on sentence alignment started with bilingual data (Brown et al., 1991; Gale and Church, 1993) adopted to train the statistical machine translation model. Monolingual sentence alignment appeared much later. Most of them are conducted on comparable corpora for developing text-to-text generation systems (e.g., Barzilay and Elhadad, 2003; Nelken and Shieber, 2006). Subsequently, it is also applied in the text simplification task (e.g., Hwang et al., 2015).

Based on the adopted search mechanism, both mono- and bilingual sentence alignment techniques can be split into greedy search (e.g., Brown et al., 1991; Hwang et al., 2015; Štajner et al., 2018) or sequence search (e.g., Gale and Church, 1993, Barzilay and McKeown, 2001).

Those previously reported monolingual alignment approaches are mainly model-agnostic, and adopt various similarity measures (Hwang et al., 2015; Štajner et al., 2018), as there is no need to additionally prepare annotated training data. As the development of NN progressed, model-dependent approaches (Huang et al., 2018; Jiang et al., 2020) also emerge, as they can deliver better performance with the cost of annotating a training dataset.

Sentence Similarity Measures

The early adopted sentence similarity measures are mostly string-based, including sentence-level tf-idf (Nelken and Shieber, 2006) or shared tokens (Barzilay and McKeown, 2001; Ganitkevitch et al., 2013). Later, to increase the possibility of recognizing those non-identical strings with similar semantic meanings, new methods are introduced: such as Word-Net similarity (e.g., Hatzivassiloglou et al., 1999), which use external resources to augment the matching scope by looking up their synsets, and WikNet similarity (Hwang et al., 2015), which is a semantic similarity based on Wiktionary.

Those embedding-based approaches appeared in literature only recently, using latent variable models (Guo and Diab, 2012) or neural models (Mueller and Thyagaraja, 2016; Neculoiu et al., 2016; Štajner et al., 2018).

6 Conclusions

We have presented the first comparison among various model-agnostic similarity measures used for aligning sentences among paraphrased paragraphs. For most cases, we find that embedding-based similarity measures outperform the string-based approaches (including the previous SOTA character trigram approach tested on the TS dataset), and sentence-embedding-based methods are preferable to the word-embedding-based methods for most search mechanisms except the uni-directional greedy matching.

Additionally, our results have shown that in calculating the similarity for sentence alignment, word vector averaging is better than adopting the [CLS] token when retrieving a representation of a whole sentence from a BERT-based model.

References

Ahmad Aghaebrahimian. 2017. Quora Question Answer Dataset. *Text, Speech, and Dialogue*. Springer International Publishing, pages 66-73. https://doi.org/10.1007/978-3-319-64206-2_8

Regina Barzilay and Noemie Elhadad. 2003. Sentence Alignment for Monolingual Comparable Corpora. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 25–32. https://aclanthology.org/W03-1004

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th Annual Meeting on Association for Computational Linguistics (ACL '01)*, pages 50–57. Association for Computational Linguistics. https://doi.org/10.3115/1073012.1073020

Regina Barzilay and Kathleen R McKeown. 2005. Sentence fusion for multidocument news summarization. *Computational Linguistics*. https://doi.org/10.1162/089120105774321091

Vuk Batanović and Dragan Bojić. 2016. Using Part-of-Speech Tags as Deep-Syntax Indicators in Determining Short-Text Semantic Similarity. *Computer Science and Information Systems*. 2015; 12(1):1−31. https://doi.org/10.2298/CSIS131127082B

Terra Blevins, Omer Levy, and Luke Zettlemoyer. 2018. Deep RNNs Encode Soft Hierarchical Syntax. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Melbourne, Australia. Association for Computational Linguistics, pages 14–19. http://dx.doi.org/10.18653/v1/P18-2003

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146. https://doi.org/10.1162/tacl_a_00051

Peter F. Brown, Jennifer C. Lai, and Robert L. Mercer. 1991. Aligning Sentences in Parallel Corpora. In *29th Annual Meeting of the Association for Computational Linguistics*, Berkeley, California, USA. Association for Computational Linguistics, pages 169–176. http://dx.doi.org/10.3115/981344.981366

Steven Burrows, Martin Potthast, and Benno Stein. 2013. Paraphrase Acquisition via Crowdsourcing and Machine Learning. In *Transactions on Intelligent Systems and Technology* (ACM TIST), pages 1-21. https://doi.org/10.1145/2483669.2483676

Xiaoqiang Chi, Yang Xiang and Ruchao Shen. 2020. Paraphrase Detection with Dependency Embedding. In *2020 4th International Conference on Computer Science and Artificial Intelligence (CSAI 2020)*, December 11-13, 2020, Zhuhai,

China. ACM, New York, NY, USA, 6. https://doi.org/10.1145/3445815.3445850

Hyunjin Choi, Judong Kim, Seongho Joe, and Youngjune Gwon. 2020. Evaluation of BERT and ALBERT Sentence Embedding Performance on Downstream NLP Tasks. In *Proceedings of the* 25th *International Conference on Pattern Recognition (ICPR 2020).* https://doi.org/10.48550/arXiv.2101.1064

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics. http://dx.doi.org/10.18653/v1/N19-1423

Thomas G. Dietterich. 1998. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation*. 10 (7): pages 1895–1923. https://doi.org/10.1162/089976698300017197

William B. Dolan and Chris Brockett. 2005. Automatically Constructing a Corpus of Sentential Paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*. https://aclanthology.org/I05-5002

Ying Ding, Junhui Li, Zhengxian Gong and Guodong Zhou. 2020. Improving neural sentence alignment with word translation. *Frontiers of Computer Science.* 15, 151302. https://doi.org/10.1007/s11704-019-9164-3

Mamdouh Farouk. 2019. Measuring Sentences Similarity: A Survey. *Indian Journal of Science and Technology*, Vol 12(25), July 2019. https://doi.org/10.17485/ijst/2019/v12i25/143977

Christiane Fellbaum. 1998 (ed.). *WordNet: An Electronic Lexical Database.* Cambridge, MA: MIT Press. https://doi.org/10.7551/mitpress/7287.001.0001

William A. Gale and Kenneth W. Church. 1993. A Program for Aligning Sentences in Bilingual Corpora. *Computational Linguistics*, 19(1):75–102. https://aclanthology.org/J93-1004

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia. Association for Computational Linguistics, pages 758–764. ttps://aclanthology.org/N13-1092

Hannaneh Hajishirzi, Wen-tau Yih, and Aleksander Kolcz. 2010. Adaptive near-duplicate detection via similarity learning. In *Proceedings of the Association for Computing Machinery Special Interest Group in Information Retrieval (ACM SIGIR)*, pages 419–426. https://doi.org/10.1145/1835449.1835520

Vasileios Hatzivassiloglou, Judith L. Klavans, and Eleazar Eskin. 1999. Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. https://aclanthology.org/W99-0625

Andrew Hickl and Jeremy Bensley. 2007. A Discourse Commitment-Based Framework for Recognizing Textual Entailment. In *Proceedings of the ACL-PASCAL Workshop on Textual Entailment and Paraphrasing*, Prague. Association for Computational Linguistics, pages 171-176. https://aclanthology.org/W07-1428

Tsutomu Hirao, Jun Suzuki, Hideki Isozaki, and Eisaku Maeda. 2004. Dependency-based Sentence Alignment for Multiple Document Summarization. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 446–452. https://doi.org/10.3115/1220355.1220419

Yonghui Huang, Yunhui Li, Yi Luan. 2018. *Monolingual sentence matching for text simplification*. Computing Research Repository, arXiv:1809.08703. Version 1. https://doi.org/10.48550/arXiv.1809.08703

Dieuwke Hupkes, Verna Dankers, Mathijs Mul, and Elia Bruni. 2020. Compositionality Decomposed: How do Neural Networks Generalise? (Extended Abstract). In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, pages 5065-5069. https://doi.org/10.24963/ijcai.2020/708

William Hwang, Hannaneh Hajishirzi, Mari Ostendorf, and Wei Wu. 2015. Aligning Sentences from Standard Wikipedia to Simple Wikipedia. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Denver, Colorado. Association for Computational Linguistics, pages 211–217. http://dx.doi.org/10.3115/v1/N15-1022

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural CRF Model for Sentence Alignment in Text Simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online. Association for Computational Linguistics, , pages

7943–7960. http://dx.doi.org/10.18653/v1/2020.acl-main.709

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI'16)*. AAAI Press, pages 2786–2792. https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12195

Paul Neculoiu, Maarten Versteegh, and Mihai Rotaru. 2016. Learning Text Similarity with Siamese Recurrent Networks. In *Proceedings of the 1st Workshop on Representation Learning for NLP*, , Berlin, Germany. Association for Computational Linguistics, pages 148–157. http://dx.doi.org/10.18653/v1/W16-1617

Rani Nelken and Stuart M. Shieber. 2006. Towards Robust Context-Sensitive Sentence Alignment for Monolingual Corpora. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. Association for Computational Linguistics, pages 161–168. https://aclanthology.org/E06-1021

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In *Proceedings of 8th International Conference on Learning Representations (ICLR 2020)*, Addis Ababa, Ethiopia, April 26-30, 2020. https://doi.org/10.48550/arXiv.1909.11942

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.* Computing Research Repository, arXiv:1907.11692. Version 1. https://doi.org/10.48550/arXiv.1907.11692

Bill MacCartney and Christopher D. Manning. 2008. Modeling Semantic Containment and Exclusion in Natural Language Inference. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, Manchester, UK. Coling 2008 Organizing Committee, pages 521–528. https://aclanthology.org/C08-1066

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330. https://aclanthology.org/J93-2004

Paul McNamee and James Mayfield. 2004. Character N-Gram Tokenization for European Language Text Retrieval. *Information Retrieval 7* (2004): 73-97. https://doi.org/10.1023/B:INRT.0000009441.78971.be

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics*, pages 3428–3448. http://dx.doi.org/10.18653/v1/P19-1334

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*. https://doi.org/10.48550/arXiv.1301.3781

Jessica Ouyang and Kathy McKeown. 2019. Neural Network Alignment for Sentential Paraphrases. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy. Association for Computational Linguistics, pages 4724–4735. http://dx.doi.org/10.18653/v1/P19-1467

Şaziye Betül Özateş, Arzucan Özgür, and Dragomir Radev. 2016. Sentence Similarity based on Dependency Tree Kernels for Multi-document Summarization. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, Portorož, Slovenia. European Language Resources Association (ELRA), pages 2833–2838. https://aclanthology.org/L16-1452

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, pages 528–540. http://dx.doi.org/10.18653/v1/N18-1049

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics, pages 1532–1543. http://dx.doi.org/10.3115/v1/D14-1162

Jan Wira Gotama Putra and Takenobu Tokunaga. 2017. Evaluating text coherence based on semantic similarity graph. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, Vancouver, Canada. Association for Computational Linguistics, pages 76–85. http://dx.doi.org/10.18653/v1/W17-2410

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, Hong Kong, China. Association for Computational Linguistics, pages 3982–3992. http://dx.doi.org/10.18653/v1/D19-1410

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. CATS: A Tool for Customized Alignment of Text Simplification Corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA). https://aclanthology.org/L18-1615

Sanja Štajner, Marc Franco-Salvador, Simone Paolo Ponzetto, Paolo Rosso, and Heiner Stuckenschmidt. 2017. Sentence Alignment Methods for Improving Text Simplification Systems. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Vancouver, Canada. Association for Computational Linguistics, pages 97–102. http://dx.doi.org/10.18653/v1/P17-2016

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python Natural Language Processing Toolkit for Many Human Languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, Online. Association for Computational Linguistics, pages 101–108. http://dx.doi.org/10.18653/v1/2020.acl-demos.14

Long Qiu, Min-Yen Kan, and Tat-Seng Chua. 2006. Paraphrase Recognition via Dissimilarity Significance Classification. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia. Association for Computational Linguistics, pages 18–26. https://aclanthology.org/W06-1603

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, Ming Zhou. 2020. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, Article 485, pages 5776–5788. https://dl.acm.org/doi/abs/10.5555/3495724.3496209

Daniela Brook Weiss, Paul Roit, Ayal Klein, Ori Ernst, and Ido Dagan. 2021. QA-Align: Representing Cross-Text Content Overlap by Aligning Question-Answer Propositions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics, pages 9879–9894. http://dx.doi.org/10.18653/v1/2021.emnlp-main.778

Bernard Lewis Welch. 1947. The generalization of 'STUDENT'S' problem when several different population variances are involved. *Biometrika*, Volume 34, Issue 1-2, pages 28–35. doi: 10.1093/biomet/34.1-2.28

Zhibiao Wu and Martha Palmer. 1994. Verb Semantics and Lexical Selection. In 32nd Annual Meeting of the Association for Computational Linguistics, Las Cruces, New Mexico, USA. Association for Computational Linguistics, pages 133–138. http://dx.doi.org/10.3115/981732.981751

Junru Zhou, Zhuosheng Zhang, and Hai Zhao. 2020. LIMIT-BERT: Linguistic Informed Multi-Task BERT. In Findings of the Association for Computational Linguistics: EMNLP 2020. Association for Computational Linguistics, pages 4450–4461. http://dx.doi.org/10.18653/v1/2020.findings-emnlp.399