Evaluating the Utility of Model Configurations and Data Augmentation on Clinical Semantic Textual Similarity

Yuxia Wang **Fei Liu** **Karin Verspoor** **Timothy Baldwin**
School of Computing and Information Systems
The University of Melbourne
Victoria, Australia
{yuxiaw, fliu3}@student.unimelb.edu.au
karin.verspoor@unimelb.edu.au tb@ldwin.net

Abstract

In this paper, we apply pre-trained language models to the Semantic Textual Similarity (STS) task, with a specific focus on the clinical domain. In low-resource setting of clinical STS, these large models tend to be impractical and prone to overfitting. Building on BERT, we study the impact of a number of model design choices, namely different fine-tuning and pooling strategies. We observe that the impact of domain-specific fine-tuning on clinical STS is much less than that in the general domain, likely due to the concept richness of the domain. Based on this, we propose two data augmentation techniques. Experimental results on N2C2-STS[1] demonstrate substantial improvements, validating the utility of the proposed methods.

1 Introduction

Semantic Textual Similarity (STS) is a language understanding task, involving assessing the degree of semantic equivalence between two pieces of text based on a graded numerical score (Corley and Mihalcea, 2005). It has application in tasks such as information retrieval (Hliaoutakis et al., 2006), question answering (Hoogeveen et al., 2018), and summarization (AL-Khassawneh et al., 2016). In this paper, we focus on STS in the clinical domain, in the context of a recent task within the framework of N2C2 (the National NLP Clinical Challenges)[1], which makes use of the extended MedSTS data set (Wang et al., 2018), referring to N2C2-STS, with limited annotated sentences pairs (1.6K) that are rich in domain terms.

Neural STS models typically consist of encoders to generate text representations, and a regression layer to measure the similarity score (He et al., 2015; Mueller and Thyagarajan, 2016; He and Lin, 2016; Reimers and Gurevych, 2019). These architectures require a large amount of training data, an unrealistic requirement in low resource settings.

Recently, pre-trained language models (LMs) such as GPT-2 (Radford et al., 2018) and BERT (Devlin et al., 2019) have been shown to benefit from pre-training over large corpora followed by fine tuning over specific tasks. However, for small-scale datasets, only limited fine-tuning can be done. For example, GPT-2 achieved strong results across four large natural language inference (NLI) datasets, but was less successful over the small-scale RTE corpus (Bentivogli et al., 2009), performing below a multi-task biLSTM model. Similarly, while the large-scale pre-training of BERT has led to impressive improvements on a range of tasks, only very modest improvements have been achieved on STS tasks such as STS-B (Cer et al., 2017) and MRPC (Dolan and Brockett, 2005) (with 5.7k and 3.6k training instances, resp.). Compared to general-domain STS benchmarks, labeled clinical STS data is more scarce, which tends to cause overfitting during fine-tuning. Moreover, further model scaling is a challenge due to GPU/TPU memory limitations and longer training time (Lan et al., 2019). This motivates us to search for model configurations which strike a balance between model flexibility and overfitting.

In this paper, we study the impact of a number of model design choices. First, following Reimers and Gurevych (2019), we study the impact of various pooling methods on STS, and find that convolution filters coupled with max and mean pooling outperform a number of alternative approaches. This can largely be attributed to their improved model expressiveness and ability to capture local interactions (Yu et al., 2019). Next, we consider different parameter fine-tuning strategies, with varying degrees of flexibility, ranging from keeping all parameters frozen during training to allowing all pa-

[1] https://portal.dbmi.hms.harvard.edu/projects/n2c2-2019-t1/

Proceedings of the BioNLP 2020 workshop, pages 105–111

rameters to be updated. This allows us to identify the optimal model flexibility without over-tuning, thereby further improving model performance.

Finally, inspired by recent studies, including sentence ordering prediction (Lan et al., 2019) and data-augmented question answering (Yu et al., 2019), we focus on data augmentation methods to expand the modest amount of training data. We first consider segment reordering (SR), in permuting segments that are delimited by commas or semicolons. Our second method increases linguistic diversity with back translation (BT). Extensive experiments on N2C2-STS reveal the effectiveness of data augmentation on clinical STS, particularly when combined with the best parameter fine-tuning and pooling strategies identified in Section 3, achieving an absolute gain in performance.

2 Related Work

2.1 Model Configurations

In pre-training, a spectrum of design choices have been proposed to optimize models, such as the pre-training objective, training corpus, and hyperparameter selection. Specific examples of objective functions include masked language modeling in BERT, permutation language modeling in XLNet (Yang et al., 2019), and sentence order prediction (SOP) in ALBERT (Lan et al., 2019). Additionally, RoBERTa (Liu et al., 2019) explored benefits from a larger mini-batch size, a dynamic masking strategy, and increasing the size of the training corpus (16G to 160G). However, all these efforts are targeted at improving downstream tasks indirectly by optimizing the capability and generalizability of LMs, while adapting a single fully-connected layer to capture task features.

Sentence-BERT (Reimers and Gurevych, 2019) makes use of task-specific structures to optimize STS, concentrating on computational and time efficiency, and is evaluated on relatively larger datasets in the general domain. For evaluating the impact of number of layers transferred to the supervised target task from the pre-trained language model, GPT-2 has been analyzed on two datasets. However, they are both large: MultiNLI (Williams et al., 2018) with >390k instances, and RACE (Lai et al., 2017) with >97k instances. These tasks also both involve reasoning-related classification, as opposed to the nuanced regression task of STS.

2.2 Data Augmentation

Synonym replacement is one of the most commonly used data augmentation methods to simulate linguistic diversity, but it introduces ambiguity if accurate context-dependent disambiguation is not performed. Moreover, random selection and replacement of a single word used in general texts is not plausible for term-rich clinical text, resulting in too much semantic divergence (e.g *patient* to *affected role* and *discharge to home* to *spark to home*). By contrast, replacing a complete mention of the concept can increase error propagation due to the prerequisite concept extraction and normalization.

Random insertion, deletion, and swapping of words have been demonstrated to be effective on five text classification tasks (Wei and Zou, 2019). But those experiments targeted topic prediction, in contrast to semantic reasoning such as STS and MultiNLI. Intuitively, they do not change the overall topic of a text, but can skew the meaning of a sentence, undermining the STS task. Swapping an entire semantic segment may mitigate the risk of introducing label noise to the STS task.

Compared to semantic and syntactic distortion potentially caused by aforementioned methods, back translation (BT) (Sennrich et al., 2016) — translating to a target language then back to the original language — presents fluent augmented data and reliable improvements for tasks demanding for adequate semantic understanding, such as low-resource machine translation (Xia et al., 2019) and question answering (Yu et al., 2019). This motivates our application of BT on low-resource clinical STS, to bridge linguistic variation between two sentences. This work represents the first exploration of applying BT for STS.

3 STS Model Configurations

In this section, we study the impact of a number of model design choices on BERT for STS, using a 12-layer base model initialized with pretrained weights.

3.1 Hierarchical Convolution (HConv)

The resource-poor and concept-rich nature of clinical STS makes it difficult to train a large model end-to-end on sentence pairs. To address this, most recent studies have made use of pre-trained language models, such as BERT. The most straightforward way to use BERT is the feature-based approach, where the output of the last transformer block is

taken as input to the task-specific classifier. Many have proposed the use of a dummy CLS token to generate the feature vector, where CLS is a special symbol added in front of every sequence during pre-training, with its final hidden state always used as the aggregate sequence representation for classification tasks, referring to CLS pooling. Other types of pooling, such as mean and max pooling, are investigated by Reimers and Gurevych (2019).

However, this results in inferior performance as shown in the first row of Table 1.[2] As a consequence, the best strategy for extracting feature vectors to represent a sentence remains an open question.

In this work, we first experiment with the feature-based approach, coupled with convolutional filters. This is inspired by the use of convolutional filters in QANet (Yu et al., 2019) to capture local interactions. The difference lies in where convolutional filters are applied. With QANet, multiple conv filters are incorporated into each transformer encoder block to process the input from the previous layer. In contrast, HConv-BERT is largely based on BERT, with the addition of a single task-specific classifier placed on top of BERT consisting of conv filters organised in a hierarchical fashion. This results in a much simplified model, making HConv-BERT less prone to overfitting.

Specifically, we run a collection of convolutional filters with a kernel of size $k \in [2, 4]$, each with $J = 768$ output channels (indexed by $j \in [1, J]$), over the temporal axis (indexed by $i \in [1, T]$):

$$c_{i,k_j} = \mathbf{w}_{k_j} * \mathbf{x}_{i:i+k-1} + b_{k_j} \quad (1)$$

$$\mathbf{c}_{i,k} = [c_{i,k_1}; \ldots; c_{i,k_J}] \quad (2)$$

where $\mathbf{x}_{i:i+k-1}$ is the output BERT features for the token span i to $i + k - 1$, $*$ is the convolution operation, $\mathbf{w}_{k_j}$ and b_{k_j} are the convolution filter and bias term for the j-th kernel of size k, and $[\mathbf{a}; \mathbf{b}]$ denotes the concatenation of $\mathbf{a}$ and $\mathbf{b}$.

To capture interactions between distant elements, we feed the output $\mathbf{c}_{i,k}$ into another convolution layer of kernel size 2 with $M = 128$ output channels (indexed by $m \in [1, M]$):

$$c^k_{i,m} = \mathbf{w}_m * \mathbf{c}_{i:i+1,k} + b_m \quad (3)$$

$$\mathbf{c}^k_i = \left[c^k_{i,1}; \ldots; c^k_{i,M}\right] \quad (4)$$

[2]Due to space constrains, we limit our comparison to the CLS pooling strategy, based on the observation of little improvements when using other types of pooling (mean, max) and concatenation, or sequence processing recurrent units.

Model	SICK-R	STS-B	N2C2-STS
Feature-based:			
CLS-BERT	53.6/52.1	49.3/67.9	14.6/28.4
HConv-BERT	80.1/73.6	83.0/83.2	79.4/74.4
Fine-tuning:			
CLS-BERT	88.6/82.9	90.0/89.6	86.7/81.9
HConv-BERT	88.7/83.5	90.1/89.6	87.7/80.7

Table 1: Pearson and Spearman correlation (r/ρ) between the predicted score and the gold labels for three STS datasets using the feature-based approach (upper half) and fine-tuning (bottom half) with CLS-BERT and HConv-BERT. Performance is reported by convention as $r/\rho \times 100$.

where $\mathbf{c}_{i:i+1,k}$ is the output of the first convolutional layer over the span i to $i + 1$ as defined in Equation (2), and $\mathbf{w}_m$ and b_m are the filter and bias term for the second convolutional layer with, a kernel size of 2 and output dimension of $M = 128$.

Lastly, we extract feature vectors by max and mean pooling over the temporal axis and then concatenation:

$$\mathbf{v}^k_{\text{max}} = \max\left(\mathbf{c}^k_i\right) \qquad \mathbf{v}^k_{\text{mean}} = \text{avg}\left(\mathbf{c}^k_i\right) \quad (5)$$

$$\mathbf{v} = \left[\mathbf{v}^2_{\text{max}}; \mathbf{v}^3_{\text{max}}; \mathbf{v}^4_{\text{max}}; \mathbf{v}^2_{\text{avg}}; \mathbf{v}^3_{\text{avg}}; \mathbf{v}^4_{\text{avg}}\right]. \quad (6)$$

The upper half of Table 1 shows that the proposed hierarchical convolutional (HConv) architecture provides substantial performance gains.

3.2 Model Flexibility

We also evaluate the utility of this mechanism in the fine-tuning setting with varying modelling flexibility. Concretely, we progressively increase the number of trainable parameters by transformer blocks. That is, for the base BERT model with 12 layers, we allow errors to be back-propagated through the last l layers while keeping the rest $(12 - l)$ fixed.

The results on STS-B and N2C2-STS are shown in Figure 1. We observe performance crossover of HConv and CLS-pooling on both datasets as the number of trainable transformer layers increases. While HConv reaches peak performance before the crossover, CLS-pooling often requires more blocks to be trainable to achieve comparable accuracy, rendering the model much slower. Notably, the proposed mechanism peaks with much fewer trainable blocks on N2C2-STS than STS-B. We speculate that this is due to the size difference between the two datasets. To verify this hypothesis, we further look into the relationship between the number of

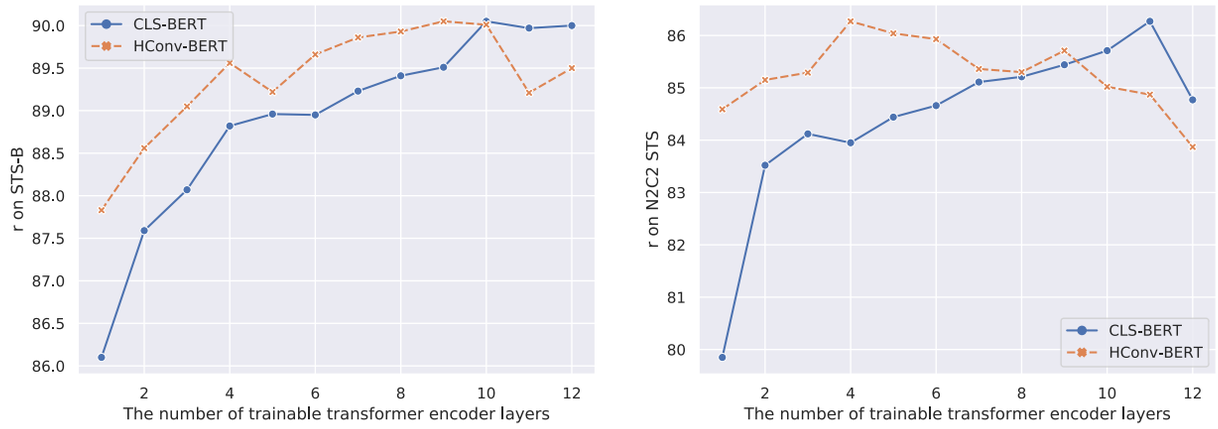

Figure 1: Evaluation of CLS-BERT and HConv-BERT over datasets from the general (STS-B) and clinical (N2C2) domains. r refers to Pearson correlation. N2C2-STS is split into 1233 and 409 instances for training and dev.

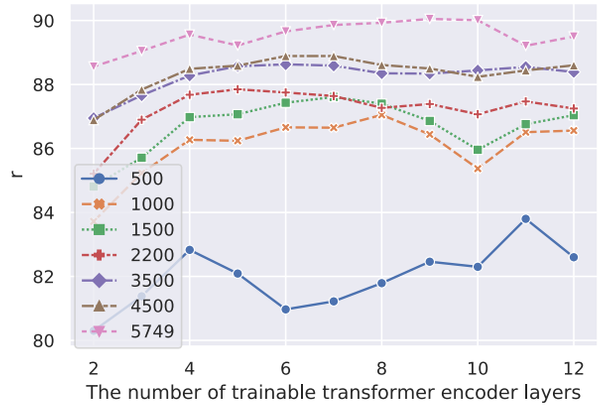

Figure 2: Impact of number of trainable transformer blocks based on HConv-BERT over different data size, randomly sampled from STS-B, ranging from 500 to full set (5,749).

trainable transformer blocks and training data size. In Figure 2, we observe performance degradation as the size of training data shrinks, with the models trained on the full set achieving far superior Pearson correlation to those trained on the smaller subsets. Zooming into the curve representing each subset, we find that peak performance is attained at different points depending on data size: with the smallest dataset (500 instances), the number of parameter updates is also limited. Only updating the top few layers of transformer blocks is simply not enough to make the model fully adapt to the task. It is therefore beneficial to allow the model access to more trainable layers (e.g., 11) to improve performance.

Based on this, we set the number of trainable blocks to 6 for SICK-R (consisting of $4,500$ training instances), as presented in the bottom half of Table 1, with HConv outperforming CLS-pooling.

4 Data Augmentation

The accuracy of an STS model unsurprisingly depends on the amount of labeled data. This is reflected in Figure 2, where models trained with more data outperform those with fewer training instances. In this section, we propose two data augmentation methods, namely segment reordering (SR) and back translation (BT), to address the data sparsity issue in clinical STS.

Segment reordering. Clinical texts often consist of text segments describing multiple events and patient symptoms. Each segment is often an independent semantic unit, separated by commas or semicolons. Inspired by the random word swapping of Wei and Zou (2019), we exploit this property and propose a heuristic, named segment reordering (SR), to generate permutations of the original sequence based on these segments. While we expect this to introduce some noise to the training data, our hypothesis is that the increase in training data size will outweigh this. For instance, consider the text *new confusion or inability to stay alert and awake; feeling like you are going to pass out*. Flipping the order of the two segments *new confusion or inability to stay alert and awake* and *feeling like you are going to pass out* will not hinder the overall understanding of the text. More formally, for a given pair of sentences S_1 and S_2, each consisting of a sequence of segments $S_1 = \{s_{11}, \ldots, s_{1m}\}$ and $S_2 = \{s_{21}, \ldots, s_{2n}\}$, we generate a new pair by randomly permuting the segment order, effectively doubling the size of the training corpus.

Back translation. Inspired by the work of Yu et al. (2019), we make use of machine translation tools to perform back translation (BT). Here, we choose Chinese as the pivot language as it is linguistically distant to English and supported by mature commercial translation solutions. That is, we first translate from English to Chinese and then back to English. We use Google Translate to translate each sentence in a sentence pair from English to Chinese, and Baidu Translation[3] to translate back to English. For example, for the original sentence *negative for cough and stridor*, the backtranslated result is *bad for coughing and wheezing*. We apply this to each sentence pair, doubling the amount of training data.

5 Experiments

5.1 Experimental Setup

We evaluate the effectiveness of SR and BT on N2C2-STS with four baseline models: $\text{BERT}_{\text{base}}$ (Devlin et al., 2019) and $\text{BERT}_{\text{clinical}}$ (Alsentzer et al., 2019), both using CLS-pooling and consisting of 12 layers; $\text{ConvBERT}_{\text{base}}$, based on $\text{BERT}_{\text{base}}$ with hierarchical convolution and fine-tuning over the last 4 layers (consistent with our findings of the best model configuration in Section 3); and $\text{ConvBERT}_{\text{STS-B}}$, where we take $\text{ConvBERT}_{\text{base}}$ and fine-tune first over STS-B, before N2C2-STS.

We split the training partition of N2C2-STS into 1,233 (train) and 409 (dev) instances, and report results on the test set (412 instances).

5.2 Results

Experimental results are presented in Table 2. We see clear benefits of the two proposed data augmentation methods, consistently boosting performance across all categories, with BT providing larger gains than SR. This is likely caused by the rather naïve implementation of SR, resulting in unnatural segment sequences. A possible fix to this is to further filter out such irregular statements with a language model pre-trained on clinical corpora. We leave this for future work.

It is impressive that the best-performing configuration $\text{ConvBERT}_{\text{STS-B}}$ + BT is capable of achieving comparable results with the state-of-the-art IBM-N2C2, an approach heavily reliant on external, domain-specific resources, and an ensemble of multiple pre-trained language models.

[3] https://fanyi.baidu.com/

Model	r	ρ
IBM-N2C2	90.1	—
$\text{BERT}_{\text{base}}$	86.7	81.9
+ SR	87.1	80.8
+ BT	87.2	81.7
$\text{BERT}_{\text{clinical}}$	86.1	81.4
+ SR	87.4	82.7
+ BT	88.6	82.4
$\text{Conv1dBERT}_{\text{base}}$	87.7	80.7
+ SR	88.0	81.4
+ BT	88.1	82.2
$\text{Conv1dBERT}_{\text{STS-B}}$	87.9	82.5
+ SR	88.6	**83.1**
+ BT	**89.4**	83.0

Table 2: Pearson r and Spearman ρ on N2C2-STS for models with and without segment reordering ("SR") and back translation ("BT").

We additionally conduct a cross-domain experiment on BIOSSES (Soğancıoğlu et al., 2017), a biomedical literature STS dataset comprising 100 sentence pairs derived from the Text Analysis Conference Biomedical Summarization task with scores ranging from 0 (complete unrelatedness) to 4 (exact equivalence). Specifically, baseline model Pooling $\text{BERT}_{\text{base}}$ and proposed $\text{ConvBERT}_{\text{STS-B}}$ + BT are both fine-tuned on N2C2-STS, and then applied with no further training to BIOSSES. Despite the increase in task difficulty, the proposed method demonstrates strong generalisability, outperforming the baseline by an absolute gain of 2.4 and 3.9 to 85.42/82.83 (r/ρ).

6 Conclusions

In this paper, we have presented an empirical study of the impact of a number of model design choices on a BERT-based approach to clinical STS. We have demonstrated that the proposed hierarchical convolution mechanism outperforms a number of alternative conventional pooling methods. Also, we have investigated parameter fine-tuning strategies with varying degrees of flexibility, and identified the optimal number of trainable transformer blocks, thereby preventing over-tuning. Lastly, we have verified the utility of two data augmentation methods on clinical STS. It may be interesting to see the impact of leveraging target languages other than Chinese in BT, which we leave for future work.

References

Yazan Alaya AL-Khassawneh, Naomie Salim, and Adekunle Isiaka Obasae. 2016. Sentence similarity techniques for automatic text summarization. *Journal of Soft Computing and Decision Support Systems*, 3(3):35–41.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. In *TAC*.

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada.

Courtney Corley and Rada Mihalcea. 2005. Measuring the semantic similarity of texts. In *Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment*, pages 13–18. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1576–1586, Lisbon, Portugal.

Hua He and Jimmy Lin. 2016. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 937–948, San Diego, California.

Angelos Hliaoutakis, Giannis Varelas, Epimenidis Voutsakis, Euripides GM Petrakis, and Evangelos Milios. 2006. Information retrieval by semantic similarity. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 2(3):55–73.

Doris Hoogeveen, Andrew Bennett, Yitong Li, Karin M Verspoor, and Timothy Baldwin. 2018. Detecting misflagged duplicate questions in community question-answering archives. In *Twelfth International AAAI Conference on Web and Social Media*.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. RACE: Large-scale ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark. Association for Computational Linguistics.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite BERT for self-supervised learning of language representations. *arXiv preprint 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. *arXiv preprint arXiv:1907.11692*.

Jonas Mueller and Aditya Thyagarajan. 2016. Siamese recurrent architectures for learning sentence similarity. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *URL https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language understanding paper.pdf*.

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)*, pages 3980–3990, Hong Kong, China.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany.

Gizem Soğancıoğlu, Hakime Öztürk, and Arzucan Özgür. 2017. BIOSSES: a semantic sentence similarity estimation system for the biomedical domain. *Bioinformatics*, 33(14):i49–i58.

Yanshan Wang, Naveed Afzal, Sunyang Fu, Liwei Wang, Feichen Shen, Majid Rastegar-Mojarad, and Hongfang Liu. 2018. MedSTS: a resource for clinical semantic textual similarity. *Language Resources and Evaluation*, pages 1–16.

Jason Wei and Kai Zou. 2019. EDA: Easy data augmentation techniques for boosting performance on text classification tasks. 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)*, pages 6381–6387, Hong Kong, China.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. 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 1112–1122, New Orleans, Louisiana. Association for Computational Linguistics.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5786–5796, Florence, Italy. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V Le. 2019. QANet: Combining local convolution with global self-attention for reading comprehension. In *The Sixth International Conference on Learning Representations (ICLR)*.