BERT-based Acronym Disambiguation with Multiple Training Strategies

Chunguang Pan [1],**Bingyan Song** [1], **Shengguang Wang** [1], **Zhipeng Luo** [1]

[1] DeepBlue Technology (Shanghai) Co., Ltd

{panchg, songby, wangshg, luozp} @deepblueai.com

Abstract

Acronym disambiguation (AD) task aims to find the correct expansions of an ambiguous ancronym in a given sentence. Although it is convenient to use acronyms, sometimes they could be difficult to understand. Identifying the appropriate expansions of an acronym is a practical task in natural language processing. Since few works have been done for AD in scientific field, we propose a binary classification model incorporating BERT and several training strategies including dynamic negative sample selection, task adaptive pretraining, adversarial training and pseudo labeling in this paper. Experiments on SciAD show the effectiveness of our proposed model and our score ranks 1st in SDU@AAAI-21 shared task 2: Acronym Disambiguation.

1 Introduction

An **acronym** is a word created from the initial components of a phrase or name, called the **expansion** (Jacobs, Itai, and Wintner 2020). In many literature and documents, especially in scientific and medical fields, the amount of acrnomys is increasing at an incredible rate. By using acronyms, people can avoid repeating frequently used long phrases. For example, CNN is an acronym with the expansion Convolutional Neural Network, though it has additional expansion possibilities depending on context, such as Condensed Nearest Neighbor.

Understanding the correlation between acronyms and their expansions is critical for several applications in natural language processing, including text classification, question answering and so on.

Despite the convenience of using acronyms, sometimes they could be difficult to understand, especially for people who are not familiar with the specific area, such as in scientific or medical field. Therefore, it is necessary to develop a system that can automatically resovle the appropriate meaning of acronyms in different contextual information.

Given an acronym and several possible expansions, **acronym disambiguation(AD)** task is to determine which expansion is correct for a particular context. The scientific

Figure 1: An example of acronym disambiguation

acronym disambiguation task is challenging due to the high ambiguity of acronyms. For example, as shown in Figure 1, SVM has two expansions in the dictionary. According to the contextual information from the input sentence, the SVM here represents for the Support Vetor Machine which is quite smilar to State Vector Machine.

Consequently, AD is formulated as a classification problem, where given a sentence and an acronym, the goal is to predict the expansion of the acronym in a given candidate set. Over the past two decades, several kinds of approaches have been proposed. At the begining, pattern-matching techniques were popular. They (Taghva and Gilbreth 1999) designed rules and patterns to find the corresponding expansions of each acronym. However, as the pattern-matching methods require more human efforts on designing and tuning the rules and patterns, machine learning based methods (i.e. CRF and SVM) (Liu, Liu, and Huang 2017) have been preferred. More recently, deep learning methods (Charbonnier and Wartena 2018; Jin, Liu, and Lu 2019) are adopted to solve this task.

Recently, pre-trained language models such as ELMo (Peters et al. 2018) and BERT (Devlin et al. 2018), have shown their effectiveness in contextual representation. Inspired by the pre-trained model, we propose a binary classification model that is capable of handling acronym disambiguation. We evaluate and verify the proposed method on the dataset released by SDU@AAAI 2021 Shared Task: Acronym Disambiguation (Veyseh et al. 2020a). Experimental results show that our model can effectively deal with the task and

we win the first place of the competition.

2 Related Work

Acronym Disambiguation

Acronym diambiguation has received a lot of attentions in vertical domains especially in biomedical fields. Most of the proposed methods (Schwartz and Hearst 2002) utilize generic rules or text patterns to discover acronym expansions. These methods are usually under circumstances where acronyms are co-mentioned with the corresponding expansions in the same document. However, in scientific papers, this rarely happens. It is very common for people to define the acronyms somewhere and use them elsewhere. Thus, such methods cannot be used for acronym disambiguation in scientific field.

There have been a few works (Nadeau and Turney 2005) on automatically mining acronym expansions by leveraging Web data (e.g. click logs, query sessions). However, we cannot apply them directly to scientific data, since most data in scientific are raw text and therefore logs of the query sessions/clicks are rarely available.

Pre-trained Models

Substantial work has shown that pre-trained models (PTMs), on the large unlabeled corpus can learn universal language representations, which are beneficial for downstream NLP tasks and can avoid training a new model from scratch.

The first-generation PTMs aim to learn good word embeddings. These models are usually very shallow for computational efficiencies, such as Skip-Gram (Mikolov et al. 2013) and GloVe (Pennington, Socher, and Manning 2014), because they themselves are no longer needed by downstream tasks. Although these pre-trained embeddings can capture semantic meanings of words, they fail to caputre higher-level concepts in context, such as polysemous disambiguation and semantic roles. The second-generation PTMs focus on learning contextual word embeddings, such as ELMo (Peters et al. 2018), OpenAI GPT (Radford et al. 2018) and BERT (Devlin et al. 2018). These learned encoders are still needed to generate word embeddings in context when being used in downstream tasks.

Adversarial Training

Adversarial training (AT) (Goodfellow, Shlens, and Szegedy 2014) is a mean of regularizing classification algorithms by generating adversarial noise to the training data. It was first introduced in image classification tasks where the input data is continuous.

Miyato, Dai, and Goodfellow (2017) extend adversarial and virtual adversarial training to the text classification by applying perturbation to the word embeddings and propose an end-to-end way of data perturbation by utilizing the gradient information. Zhu, Li, and Zhou (2019) propose an adversarial attention network for the task of multi-dimensional emotion regression, which automatically rates multiple emotion dimension scores for an input text.
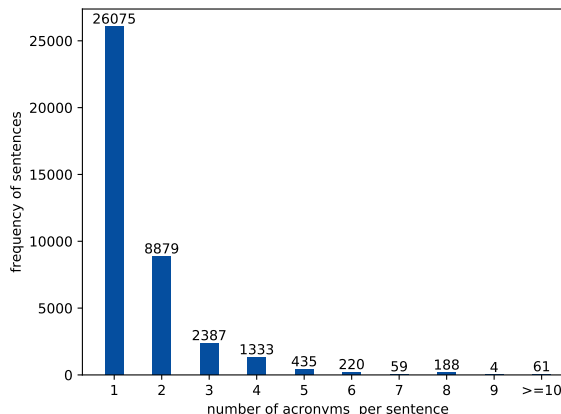

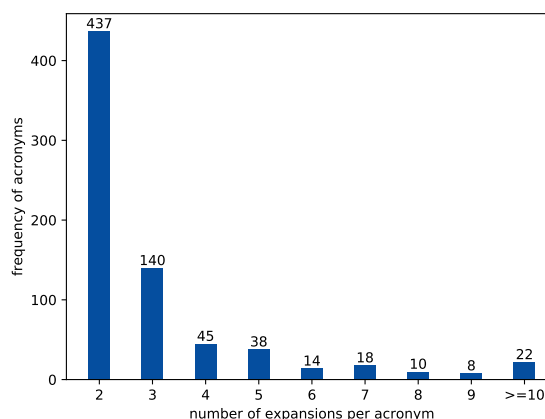Figure 2: Number of acronyms per sentence

Figure 3: Number of expansions per acronym

There are also other works for regularizing classifiers by adding random noise to the data, such as dropout (Srivastava et al. 2014) and its variant for NLP tasks, word dropout (Iyyer et al. 2015). Xie et al. (2019) discusses various data noising techniques for language models and provides empirical analysis validating the relationship between nosing and smoothing. Søgaard (2013) and Li, Cohn, and Baldwin (2017) focus on linguistic adversaries.

Combining multiple advantages in above works, we propose a binary classification model utilizing BERT and several training strategies such as adversarial training and so on.

3 Data

In this paper, we use the AD dataset called **SciAD** released by Veyseh et al. (2020b). They collect a corpus of 6,786 English papers from arXiv and these papers consist of 2,031,592 sentences that could be used for data annotation.

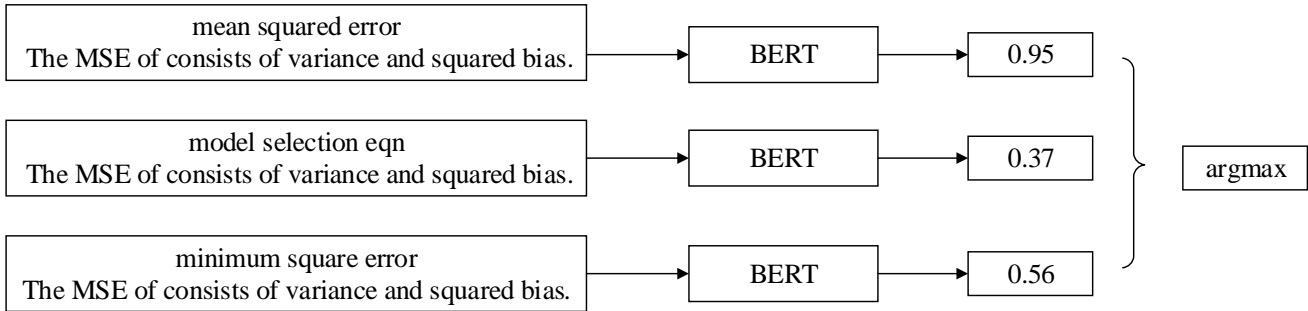The dataset contains 62,441 samples where each sample

Figure 4: Acronym disambiguation based on binary classification model. For each sample, the model needs to predict whether the given expansions matches the acronym or not, and find the expansion with the highest score as the correct one.

involves a sentence, an ambiguous acronym, and its correct meaning (one of the meanings of the acronym recorded by the dictionary, as shown in 1).

Figure 2 and Figure 3 demonstrate statistics of SciAD dataset. More specifically, Figure 2 reveals the distribution of number of acronyms per sentence. Each sentence could have more than one acronym and most sentences have 1 or 2 acronyms. Figure 3 shows the distribution of number of expansions per acronym. The distribution shown in this figure is consistent with the same distribution presented in the prior work (Charbonnier and Wartena, 2018) in which in both distributions, acronyms with 2 or 3 meanings have the highest number of samples in the dataset (Veyseh et al. 2020b).

4 Binary Classification Model

The input of the binary classification model is a sentence with an ambiguous acronym and a possible expansion. The model needs to predict whether the expansion is the corresponding expansion of the given acronym. Given an input sentence, the model will assign a predicted score to each candidate expansion. The candidate expansion with the highest score will be the model output. Figure 4 shows an example of the procedure.

Input Format

Since BERT can process multiple input sentences with segment embeddings, we use the candidate expansion as the first input segment, and the given text as the second input segment. We separat these two input segments with the special token [CLS]. Furthermore, we add two special tokens <start> and <end> to wrap the acronym in the text, which enables that the acronym can get enough attention from the model.

Binary Model Architecture

The model architecture is described in Figure 5 in detail. First, we use a BERT encoder to get the representation of input segments. Next, we calculate the mean of the start and end positions of the acronym, and concatenate the representation with the [CLS] position vector. Then, we sent this

concatenated vector into a binary classifier for prediction. The represenation first pass through a dropout layer (Srivastava et al. 2014) and a feedforward layer. The output of these layers is then feed into a ReLU (Glorot, Bordes, and Bengio 2011) activation. After this, the calculated vector pass through a dropout layer and a feedforward layer again. The final prediction can be obtained through a sigmoid activation.

Training Strategies

Pretrained Models Experiments from previous work have shown the effectiveness of pretrained models. Starting from BERT model, there are many improved pretrained models. Roberta uses dynamic masks and removes next sentence prediction task. In our experiments, we compare BERT and Roberta models trained on corpus from different fields.

Dynamic Negative Sample Selection During training, we dynamicly select a fixed number of negative samples for each batch, which ensures that the model is trained on more balanced positive and negative data, and all negative samples are used in training at the same time.

Task Adaptive Pretraining Gururangan et al. (2020) shows that task-adaptive pretraining (TAPT) can effectively improve model performance. The task-specific dataset usually covers only a subset of data used for general pretraining, thus we can achieve significant improvement by pretraining the masked language model task on the given dataset.

Adversarial Training Adversarial training is a popular approach to increasing robustness of neural networks. As shown in Miyato, Dai, and Goodfellow (2017), adversarial training has good regularization performance. By adding perturbations to the embedding layer, we can get more stable word representations and a more generalized model, which significantly improves model performance on unseen data.

Pseudo-Labeling Pseudo labeling (Iscen et al. 2019; Oliver et al. 2018; Shi et al. 2018) uses network predictions with high confidence as labels. We mix these pseudo labels and the training set together to generate a new dataset. We
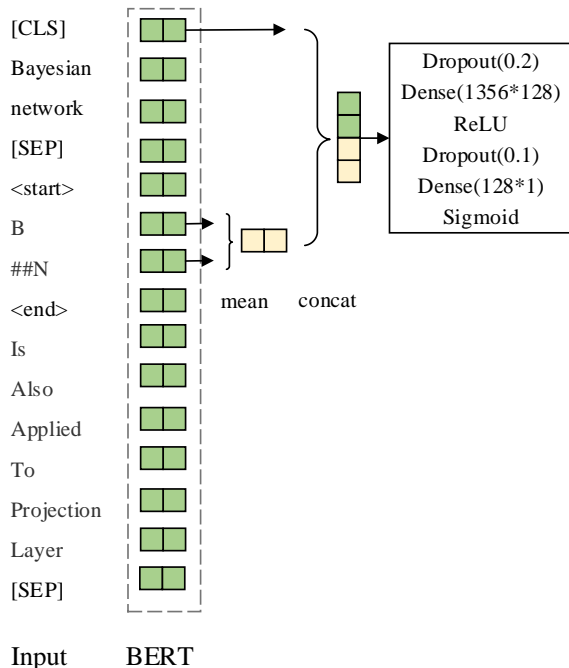
Figure 5: The binary classification model.

than use this new dataset to train a new binary classification model. Pseudo-labeling has been proved an effective approach to utilize unlabeled data for a better performance.

5   Experiments

Hyper parameters

The batch size used in our experiments is 32. We train each model for 15 epochs. The initial learning rate for the text encoder is 1.0×10^{-5}, and for other parameters, the initial learning rate is set to 5.0×10^{-4}. We evaluate our model on the validation set at each epoch. If the macro F1 score doesn't increase, we then decay the learning rate by a factor of 0.1. The minimum learning rate is 5.0×10^{-7}. We use Adam optimizer (Kingma and Ba 2017) in all our experiments.

Pretrained Models

Since different pretrained models are trained using different data, we do experiments on several pretrained models. Table 1 shows our experimental results on different pretrained models in validation set. The bert-base model gets the highest score in commonly used pretrained models (the top 3 lines in Table 1). Since a large ratio of texts in the given dataset come from computer science field, the cs-roberta model outperforms the bert-base model by 1.6 percents. The best model in our experiments is the scibert model, which achieves the F1 score of 89%.

Model	Precision	Recall	F1
bert-base-uncased	0.9176	0.8160	0.8638
bert-large-uncased	0.9034	0.7693	0.8311
roberta-base	0.9008	0.7687	0.8295
cs-roberta-base	0.9216	0.8415	0.8797
scibert-scivocab-uncased	**0.9263**	**0.8569**	**0.8902**

Table 1: Results on validation set using different pretrained models.

Training Procedure

We incorporate all the training strategies introduced above to improve the performance of our proposed binary classification model. According to the experiment result in Table 1, we choose scibert as the fundamental pretrained model and use the TAPT technique to train a new pretrained model. Then we add the dynamic negative sample selection and adversarial training strategies to train the binary classfication model. After this, we utilize the pseudo-labeling technique and obtain the final binary classification model.

Further Experiments

Combining training strategies   We do some futher experiments on validation set to verify the effectiveness of each strategy mentioned above. The results are shown in Table 2. As shown in the table, F1 score increases by 4 percents with dynamic sampling. TAPT and adversarial training further improve the performance on validation set by 0.47 percent. Finally, we use pseudo-labeling method. Samples from the test set with a score higher than 0.95 are selected and mixed with the training set. It still slightly improves the F1 score.

Model	Precision	Recall	F1
scibert-scivocab-uncased	0.9263	0.8569	0.8902
+dynamic sampling	0.9575	0.9060	0.9310
+task adaptive pretraining	0.9610	0.9055	0.9324
+adversarial training	**0.9651**	0.9082	0.9358
+pseudo-labeling	0.9629	**0.9106**	**0.9360**

Table 2: Results on validation set using different training approaches.

Error Analysis   We gather a sample of 100 development set examples that our model misclassified and look at these examples manually to do the error analysis.

From these examples, we find that there are two main cases where the model gives the wrong prediction. The first one is that the candidate expansions are too similar, even have the same meanings in different forms. For example, in the sentence 'The SC is decreasing for increasing values of ...', the correct expansion for 'SC' is 'sum capacities' while our prediction is 'sum capacity' which has the same meaning with the correct one but in the singular form.

The second one is that there is too little contextual information in the given sentence for prediction. For instance, the correct expansion for 'ML' in sentence 'ML models are usually much more complex, see Figure.' is 'model logic', the

predict expansion is 'machine learning'. Even people can hardly tell which one is right only based on the given sentence.

Time complexity   To analysis the time complexity of our proposed method, we show measurements of the actual running time observed in our experiments. The discussions are not that precise or exhaustive. However, we believe they are enough to offer readers rough estimations of the time complexity of our model.

We utilize TAPT strategy to further train the scibert model by using eight NVIDIA TITAN V (12GB). It takes three hours to train 100 epochs in total.

After getting the new pretrained model, we trained the binary classification model on two NVIDIA TITAN V. On average, each epoch of the training and inference time of adding adversarial training and pseudo-labeling are shown in Table 3 respectively. It begins to converge after five epochs. It takes nearly the same time to do the inference while the training time is twice as long after adversarial training is added.

Model	Train	Inference
	$1588s$	$150.42s$
+adversarial training	$3021s$	$149.64s$
+pseudo-labeling	$3328s$	$149.36s$

Table 3: Time complexity

Comparison Results   We compared our results with several other models. Precision, Recall and F1 of our proposed model are computed on testing data via the cross-validation method.

- **MF & ADE** Non-deep learning models that utilize rules or hand crafted features (Li et al. 2018).

- **NOA & UAD** Language-model-based baselines that train the word embeddings using the training corpus (Charbonnier and Wartena 2018; Ciosici and Assent 2019).

- **BEM & DECBAE** Models employ deep architectures (e.g., LSTM) (Jin, Liu, and Lu 2019; Blevins and Zettlemoyer 2020).

- **GAD** A deep learning model utilizes the syntactical structure of the sentence (Veyseh et al. 2020b).

Model	Precision	Recall	F1
MF	0.8903	0.4220	0.5726
ADE	0.8674	0.4325	0.5772
NOA	0.7814	0.3506	0.4840
UAD	0.8901	0.7008	0.7837
BEM	0.8675	0.3594	0.5082
DECBAE	0.8867	0.7432	0.8086
GAD	0.8927	0.7666	0.8190
Ours	**0.9695**	**0.9132**	**0.9405**
Human Performance	0.9782	0.9445	0.9610

Table 4: Results of different models on testing dataset

As shown in Table 4, rules/features fail to caputre all patterns of expressing the meanings of the acronym, resulting in poorer recall on expansions compared to acronyms. In contrast, the deep learning model has comparable recall on expansions and acronyms, showing the importance of pretrained word embeddings and deep architectures for AD. However, they all fall far behind human level performance. Among all the models, our proposed model achieves the best results on the SciAD and is very close to the human performance which shows the capability of the strategies we introduced above.

SDU@AAAI 2021 Shared Task: Acronym Disambiguation   The competition results are shown in Table 5. We show scores of the top 5 ranked models as well as the baseline model. The baseline model is released by the provider of the SciAD dataset (Veyseh et al. 2020b). Our model performs best among all the ranking list and outperforms the second place by 0.32%. In addition, our model outperforms the baseline model by 12.15% which is a great improvement.

Model	Precision	Recall	F1
Rank1	**0.9695**	**0.9132**	**0.9405**
Rank2	0.9694	0.9073	0.9373
Rank3	0.9652	0.9009	0.9319
Rank4	0.9595	0.8959	0.9266
Rank5	0.9548	0.8907	0.9216
Baseline	0.8927	0.7666	0.8190

Table 5: Leaderboard

6   Conclusion

In this paper, we introduce a binary classification model for acronym disambiguation. We utilize the BERT encoder to get the input representations and adopt several strategies including dynamic negative sample selection, task adaptive pretraining, adversarial training and pseudo-labeling. Experiments on SciAD show the validity of our proposed model and we win the first place of the SDU@AAAI-2021 Shared task 2.

References

Blevins, T., and Zettlemoyer, L. 2020. Moving down the long tail of word sense disambiguation with gloss-informed biencoders. *arXiv preprint arXiv:2005.02590*.

Charbonnier, J., and Wartena, C. 2018. Using word embeddings for unsupervised acronym disambiguation. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2610–2619.

Ciosici, M., and Assent, I. 2019. Abbreviation explorer-an interactive system for pre-evaluation of unsupervised abbreviation disambiguation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, 1–5.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gururangan, S.; Marasović, A.; Swayamdipta, S.; Lo, K.; Beltagy, I.; Downey, D.; and Smith, N. A. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 8342–8360. Online: Association for Computational Linguistics.

Iscen, A.; Tolias, G.; Avrithis, Y.; and Chum, O. 2019. Label propagation for deep semi-supervised learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5070–5079.

Iyyer, M.; Manjunatha, V.; Boyd-Graber, J.; and Daumé III, H. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, 1681–1691.

Jacobs, K.; Itai, A.; and Wintner, S. 2020. Acronyms: identification, expansion and disambiguation. *Annals of Mathematics and Artificial Intelligence* 88(5):517–532.

Jin, Q.; Liu, J.; and Lu, X. 2019. Deep contextualized biomedical abbreviation expansion. *arXiv preprint arXiv:1906.03360*.

Kingma, D. P., and Ba, J. 2017. Adam: A method for stochastic optimization.

Li, Y.; Zhao, B.; Fuxman, A.; and Tao, F. 2018. Guess me if you can: Acronym disambiguation for enterprises. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 1308–1317.

Li, Y.; Cohn, T.; and Baldwin, T. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, 21–27.

Liu, J.; Liu, C.; and Huang, Y. 2017. Multi-granularity sequence labeling model for acronym expansion identification. *Information Sciences* 378:462–474.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, 3111–3119.

Miyato, T.; Dai, A. M.; and Goodfellow, I. 2017. Adversarial training methods for semi-supervised text classification. In *Proceedings of International Conference on Learning Representations*.

Nadeau, D., and Turney, P. D. 2005. A supervised learning approach to acronym identification. In *Conference of the Canadian Society for Computational Studies of Intelligence*, 319–329. Springer.

Oliver, A.; Odena, A.; Raffel, C. A.; Cubuk, E. D.; and Goodfellow, I. 2018. Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in neural information processing systems*, 3235–3246.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1532–1543.

Peters, M.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237.

Radford, A.; Narasimhan, K.; Salimans, T.; and Sutskever, I. 2018. Improving language understanding by generative pre-training (2018). *URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/language-unsupervised/language_ understanding_paper. pdf*.

Schwartz, A. S., and Hearst, M. A. 2002. A simple algorithm for identifying abbreviation definitions in biomedical text. In *Biocomputing 2003*. World Scientific. 451–462.

Shi, W.; Gong, Y.; Ding, C.; MaXiaoyu Tao, Z.; and Zheng, N. 2018. Transductive semi-supervised deep learning using min-max features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 299–315.

Søgaard, A. 2013. Part-of-speech tagging with antagonistic adversaries. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 640–644.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15(1):1929–1958.

Taghva, K., and Gilbreth, J. 1999. Recognizing acronyms and their definitions. *International Journal on Document Analysis and Recognition* 1(4):191–198.

Veyseh, A. P. B.; Dernoncourt, F.; Nguyen, T. H.; Chang, W.; and Celi, L. A. 2020a. Acronym identification and disambiguation shared tasksfor scientific document understanding. *arXiv preprint arXiv:2012.11760*.

Veyseh, A. P. B.; Dernoncourt, F.; Tran, Q. H.; and Nguyen, T. H. 2020b. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3285–3301.

Xie, Z.; Wang, S. I.; Li, J.; Lévy, D.; Nie, A.; Jurafsky, D.; and Ng, A. Y. 2019. Data noising as smoothing in neural network language models. In *5th International Conference on Learning Representations, ICLR 2017*.

Zhu, S.; Li, S.; and Zhou, G. 2019. Adversarial attention modeling for multi-dimensional emotion regression. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 471–480.