2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020161

Emotion Recognition on StackOverflow Posts Using BERT

1st Donald Bleyl
school of computer science
georgia institute of technology
Atlanta, GA
dbleyl3@gatech.edu

2nd Elham Khorasani Buxton
department of computer science
University of Illinois at Springfield
Springfield, IL
esahe2@uis.edu

***Abstract*—Social programming websites like GitHub and StackOverflow have become an increasingly important aspect of software development and the publicly available datasets provide a rich source of data for exploring challenging NLP problems. One such problem is emotion recognition. This work applies deep NLP methods for detecting emotions in StackOverflow content. Several BERT models were trained and fine-tuned on a small, sparse, hand-labeled and highly-imbalanced dataset of StackOverflow comments. Text augmentation techniques were used to balance the data and the model's vocabulary was enhanced with common domain-specific terms and emoticons. Unsupervised post-training was applied on a large unlabeled StackOverflow dataset to learn representations for added vocabulary before fine-tuning on labeled data. The final model was benchmarked and compared to prior studies on the same dataset.**

***Index Terms*—Emotion Detection, StackOverflow, Natural Language Processing, BERT**

I. Introduction

The impact of emotion and sentiment on the social programmer ecosystem has become an emerging area of study for NLP researchers, and several challenging aspects of the subject have been identified [1]. Although there have been many recent ground-breaking discoveries in the field of deep learning and NLP, these techniques are only now being applied to sentiment analysis in the context of the social programmer ecosystem.

Publicly available datasets for social programming websites are numerous and widely available; however, only a few of them have labels for supervised learning. Of those datasets that contain labels, most of them are limited to polarity and lack labels for more fine-grained emotions such as joy, anger, or fear. Prior research has shown that the emotional style of a post on StackOverflow can play a part in receiving up or down votes or in obtaining a satisfying answer to a technical question [2]. Traditional sentiment analysis has focused on the task of classifying observations as positive or negative, and sometimes neutral. Although useful, it only scratches the surface of what can be done with natural language processing. In its basic form, sentiment analysis can be considered a binary or a multi-class, single-label classification problem. In contrast, *emotion classification* is often a *multi-class*, *multi-label* problem, since humans are capable of expressing a complex range of emotions within a single observation.

Regardless of the model used, lack of sufficient labeled data makes it difficult to train a robust classifier to detect emotions from text. To compensate for insufficiently labeled data, a common practice in Natural Language Processing (NLP) is to use models that have been pre-trained on general, large unlabeled text corpora and fine-tune them on a much smaller labeled dataset in the target domain. In particular, fine-tuning pre-trained transformer-based models such as BERT (Bidirectional Encoder Representations From Transformers) have dominated NLP applications in recent years and achieved state of the art results [3]. This study describes the application of BERT in detecting emotions of StackOverflow posts. Although BERT has been used for emotion recognition in a general context [4]–[9], to the best of our knowledge, no prior work has fine-tuned BERT for fine-grained emotion detection on StackOverflow or other social programming data.

The contributions of this work are: 1- applying Bert for fine-grained emotion detection on StackOverflow posts 2- adding frequently used technical terms in StackOverflow to Bert's vocabulary. 3- unsupervised fine-tuning of BERT using Masked Language Modeling on a large dataset of unlabeled StackOverflow posts and 4- supervised fine-tuning of BERT on a small labeled StackOverflow emotion detection dataset.

II. Dataset

A Gold Standard for Emotion Annotation in StackOverflow is a hand-labeled dataset described in [10]. This dataset is an annotated sampling from the original unlabeled StackOverflow dataset. In [10], Novielli et al described a method of constructing a high-quality dataset for StackOverflow affective expression research, and published a dataset of 4,800 observations annotated by 12 volunteers. Each observation was annotated by three participants for six basic emotions in Parrot's framework: love, joy, surprise, anger, sadness and fear [11]. A simple majority voting scheme was used to label the observation-emotion pairs. If at least two of the three participants annotated an observation as having a particular emotion, the observation was labeled with that emotion. Not all observation-emotion pairs have labels, and the labels are not mutually exclusive. As a result, the data lends itself to a multi-class, multi-label classification problem. tableI shows examples observations from the Gold Standard dataset.

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

Group	Set	Id	Text	Rater 1	Rater 2	Rater 3	Gold Label
A	Second	1	SVG transform on text attribute works …	X		X	LOVE
A	Second	2	Excellent! This is exactly what I needed. …	X	X	X	LOVE
A	Second	3	Have added a modern solution as of May …				

TABLE I
EXAMPLES FROM *A Gold Standard for Emotion Annotation in StackOverflow* [10]

III. RELATED WORK

Prior studies have shown that off-the-shelf sentiment analysis tools trained on social media perform poorly on software engineering data [12]–[15]. As a result, there have been several efforts to explore sentiment analysis customized to the social programming domain. [16] customized *SentiStrength* (a general purpose lexicon-based sentiment analysis tool) for software engineering texts. Udding and Khomh [17] focused on sentiment analysis in a sample of API reviews extracted from StackOverFlow. Ahmed et. al. [18] created a dataset of StackOverflow code review samples hand-labeled as positive, negative or neutral and trained several supervised learning models on this dataset. Lin et. al., [19] trained a Recurrent Neural Network (RNN) on a dataset of manually labeled sentences extracted from StackOverflow but their model did not achieve a satisfactory performance on classifying positive and negative sentences.

The *Gold Standard* dataset used in this study has been used in other works to create sentiment analysis tools for software engineering domain. Calefato, et. al. [20] developed a sentiment analysis tool, *Senti4SD*, that utilizes a set of manually crafted features to train a supervised learning model on the Gold Standard dataset to detect the polarity of a StackOverflow post.

A few recent studies have applied BERT for sentiment analysis on StackOverflow posts. Zhang et. al [19] compared the performance of a fine-tuned BERT model and some of its variants to the prior sentiment analysis tools for software engineering (such as SentiCR, SentiStrength and Senti4SDR) across six datasets with annotated polarities. Their findings showed that the fine-tuned BERT models significantly outperformed the prior tools. A similar conclusion was reached in [21] where authors evaluated an ensemble of fine-tuned BERT variants for sentiment classification on StackOverflow posts, JIRA issues, and Github commit comments. In [22] authors fine-tuned a pre-trained BERT model on a dataset of 4000 annotated sentences from StackOverflow posts and showed its superior performance compared to an RNN based model developed by the same group in a prior study.

All of the aforementioned studies focused on sentiment and polarity classification as opposed to more fine-grained emotions. Emotion detection is significantly more challenging than sentiment analysis because labels overlap and a single observation can express multiple emotions. In addition, labels are often noisy and there might be no consensus among human raters on the emotion labels for some observations.

Unlike sentiment detection, there have not been many studies on emotion detection in a software engineering context. Murgia et. al. [23] hand-labeled a dataset of 1600 issue comments of the Apache Software Foundation into six primary emotions in Parrot's framework. They found that human raters agree more on the presence of love, sadness and joy as opposed to fear, anger, and surprise. They converted each comment to a bag of words and trained a separate classifier for love, sadness, and joy emotion categories. Anger, fear, and surprise emotions were not evaluated for lack of sufficient data. Their best classifier obtained a high precision in detection of love, joy and sadness but a low recall in detection of joy and sadness. A different emotion framework on the valence and arousal space was used in [24] and [25] to detect emotion categories excitement, stress, depression, relaxation, neutral, and average on StackOverFlow posts.

The most similar work to this study is [26] where authors developed a toolkit called *EmoTxt* for detecting emotions using the same Gold Standard dataset of StackOverflow posts used here. EmoTxt uses a set of lexical features to train a linear SVM classifier and achieves a relatively high F1 scores on detecting *love* and anger, a moderate F1 score on detecting *sadness* and *joy* and a relatively poor F1 score on detecting *surprise* and *fear*. This is likely because the number of observations labeled with *surprise* and fear are significantly lower in the Gold Standard dataset. Data imbalance is only addressed by under-sampling in EmoTxt and features selected did not take into account the order of words and their relation in texts. In contrast, BERT-based models used in this study can effectively represent bidirectional semantic relationship between words in a sentence. In addition, we balanced data using various text augmentation techniques to compensate for lack of sufficient data in emotion categories with lower number of observations.

IV. DATA PROCESSING

The dataset contains one worksheet per emotion label {love, joy, surprise, anger, sadness, fear}. Data was validated by comparing the dimensions of every worksheet to ensure that they all have the same number of observations and features then worksheets were combined into a single dataset with one column per label. Labels were *multi-hot encoded* by converting non-null values into one and null values to zero. table II shows example observations from the cleaned dataset. The number of observations is highly imbalanced across emotion labels with multiple minority classes as shown in figure 1. Observations that did not have a label were assumed to be *neutral* (or expressed none of the other emotion labels). About 56% of comments have a single emotion label, 6% have two or more labels and the rest are neutral or missing labels.

We examined two text augmentation techniques for balancing data in underrepresented emotion categories:

text	Love	Joy	Surprise	Anger	Sadness	Fear
SVG transform on text attribute works excellen excellen...	1	0	0	0	0	0
Excellent! This is exactly what I needed. Thanks!	1	0	0	0	0	0
Have added a modern solution as of May 2014 in...	0	0	0	0	0	0

TABLE II
EXAMPLE OF ENCODED OBSERVATIONS

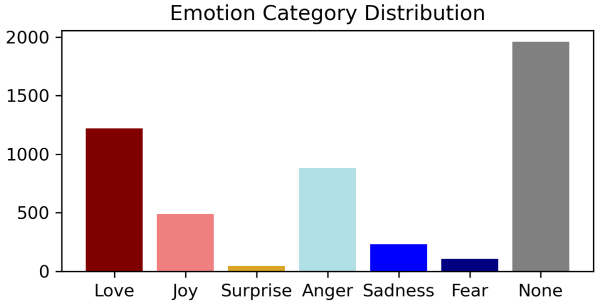

Fig. 1. Observation counts per emotion category

- **synonym replacement**: 30% of words in the text were randomly chosen and replaced by one of their synonyms from WordNet lexical database.
- **contextual word embedding**: 30% of words in the text were randomly chosen and replaced by their closest words in the contextual embedding space obtained by a BERT model.

V. PROPOSED BERT MODELS FOR EMOTION DETECTION ON STACKOVERFLOW COMMENTS

BERT [3] is a transformer based language model pretrained on large text corpora using a combination of Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) objectives. BERT is used to extract representations from text and can be fine-tuned to a downstream classification task.

We used the BERT-Base Uncased model as introduced in [3] with 12 transformer blocks, each with 12 self-attention heads and an embedding dimension of 768. This is a general-purpose language model designed to create representations that work across multiple domains. As a result, BERT's vocabulary does not include many of the frequent technical words used in StackOverflow comments (such as *docs, json, url, javascript, jquery, div, snippet, debug, CSS, mysql*, etc.). Furthermore, while emoticons are important indicators of sentiment and emotions in text, they are absent in BERT's vocabulary. As a result, domain specific words and emoticons are tokenized to subwords. For instance, the BERT WordPiece tokenizer tokenizes the comment ”Ah yes, I must have been thinking of MySQL syntax :D” as ⟨’ah’, ’yes’, ’,’, ’i’, ’must’, ’have’, ’been’, ’thinking’, ’of’, ’my’, ’##s’, ’##q’, ’##l’, ’syntax’, ’:’, ’d’⟩ where the words ”mysql” and the emoticon ”:D” are split to subwords.

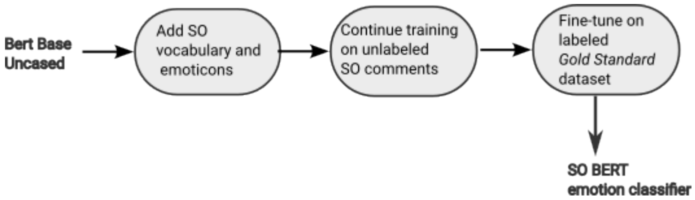
Bert Base Uncased
Add SO vocabulary and emoticons
Continue training on unlabeled SO comments
Fine-tune on labeled Gold Standard dataset
SO BERT emotion classifier

Fig. 2. Building the BERT classifier for StackOverflow emotion detaction

Prior approaches for adjusting BERT tokenizer to a new domain range from pretraining BERT from scratch with a domain-specific vocabulary [27] to fine-tuning the original model on domain-specific corpus without changing its vocabulary [28]. Other approaches such as [29] extended the original BERT architecture to include embedding layers for domain-specific vocabulary.

Pretraining BERT from scratch is both resource and data intensive. Instead, we added the top 993 frequent technical words in a large corpus of StackOverflow comments together with commonly used emoticons to BERT's tokenizer vocabulary. To learn representations for the added words, we continued training BERT on a set of 1 million unlabeled StackOverflow comments using masked language modeling and next sentence prediction objectives.

Finally, we split the labeled Gold Standard dataset for emotion detection into train, validation, and test sets, added a classification head to the fine-tuned BERT model and fine-tuned it again on the labeled training set using binary cross entropy loss.

Figure 2 illustrates the steps taken to build the BERT classifier for StackOverflow (SO) emotion detection.

A. *multi-label and one-vs-all emotion classifiers*

We examined two multi-class models: 1- a single multi-label model for all emotion categories, and 2- a separate binary model per category (i.e., one-vs.all classifiers).

Balancing data in a multi-label setting is challenging as samples might be associated with labels from both minority and majority classes so augmentation may result in increasing samples for both classes at the same time. We used a simple two-pass augmentation strategy: in the first pass we augmented random samples from minority classes (*joy, sadness, fear, and surprise*) in order to match the number of samples in the majority class (*love*). While this step resulted in a more balanced dataset, some imbalance was still present in the data due to the fact that some augmented samples were multi-labeled with majority and minority classes. To fix the remaining imbalance, we made a second pass through the data and this time only selected samples that were single-labeled with a minority class for augmentation. Finally, the *none/neutral* emotion class was down-sampled to match the rest of the classes.

In one-vs-all setting, the data was balanced for each emotion category separately. That is, for each emotion category for example *joy*, we divided the samples into two sets: samples labeled with *joy* and samples not labeled with *joy*. We augmented the set with fewer samples until we had roughly the same number of samples in both sets.

To prevent data leakage, only the training samples were balanced in both multi-label and one-vs.all settings.

B. Comparison with the GoEmotion BERT Model

GoEmotions [9] is a corpus of 58K curated Reddit comments manually labeled to 27 emotion categories and neutral. The emotion categories are: *admiration, amusement, anger, annoyance, approval, caring, confusion, curiosity, desire, disappointment, disapproval, disgust, embarrassment, excitement, fear, gratitude, grief, joy, love, nervousness, optimism, pride, realization, relief, remorse, sadness, surprise*. Collected by the Google Research Team, this is the largest human-annotated dataset for fine-grained emotion detection. The GoEmotions paper also includes a baseline model that is a pre-trained BERT-base model fine-tuned on the GoEmotions dataset.

We were interested to understand whether fine-tuning on the GoEmotions dataset would improve performance on StackOverflow emotion detection. To this end, we used the GoEmotions baseline BERT model to extract features from the Gold Standard dataset of StackOverflow comments and computed its performance on the holdout set.

VI. Results

We benchmarked the proposed SO BERT models against the Emotxt model [26].

The decision threshold for the multi-label SO BERT model and GoEmotion BERT model was set to 20%. This means that to each sample in the hold out set, we assigned all emotion labels with probabilities greater than or equal to 20%.

Table III lists the F1 scores on the holdout set for the proposed Bert base models versus Emotxt.

It is evident from table III that all BERT-based models significantly outperform the Emotxt toolkit across all emotion categories.

Among the three BERT models we experimented with, the multi-label SO BERT model achieves the highest macro average F1 score followed by the GoEmotion BERT model and the one-vs-all SO BERT models. This is expected as the multi-label SO model and GoEmotion has the ability to capture correlation among different emotions. The lower F1 score for the surprise category for the multi-label and GoEmotion models can be attributed to its low sample size and the fact that surprise is an ambiguous emotion category that can be correlated with both positive and negative emotions.

We found that while fine-tuning on GoEmotions dataset improves the performance of the pre-trained BERT model on StackOverflow emotion detection, adding StackOverflow-specific vocabulary to BERT and fine-tuning it on unlabeled data yields better performance.

VII. Conclusion and Future Work

In this work we developed three BERT-based models for emotion detection of StackOverflow comments using a small hand-labeled dataset for emotion annotation described in [10]. These models include:

- one-vs-all StackOverflow BERT
- multi-label StackOverflow BERT
- multi-label BERT on GoEmotions

For the first two models above, we extended BERT vocabulary by adding the frequent technical words and emoticons found in StackOverflow comments and then fine-tuned it on one million unlabeled StackOverflow comments in order to learn representations for the added vocabulary. We found that this unsupervised fine-tuning step significantly boosted the performance of our SO BERT models. The latter model used the labeled GoEmotions dataset for fine-tuning BERT. We benchmarked all BERT-based models against Emotxt toolkit which is the state-of-the-art none-deep learning model for emotion detection from technical text. All three BERT models outperformed the Emotxt toolkit on the same hold out dataset by over 10% average F1 score.

The most important limitation of this work is the small size of the Gold Standard dataset used for fine-tuning the SO BERT models. Prior research has shown that fine-tuning BERT on small dataset is an unstable process [3], [30]. While human labeling of data for emotion detection is cost intensive, one could use heuristics for automatic weak labeling of text. For future work, we intend to construct a larger labeled dataset for fine-tuning SO BERT using emojis and emoticons as proxies for weak emotion labeling.

References

[1] N. Novielli, F. Calefato, and F. Lanubile, “The challenges of sentiment detection in the social programmer ecosystem,” in *Proceedings of the 7th International Workshop on Social Software Engineering*, 2015, pp. 33–40.

[2] ——, “Towards discovering the role of emotions in stack overflow,” in *Proceedings of the 6th international workshop on social software engineering*, 2014, pp. 33–36.

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

[4] F. A. Acheampong, H. Nunoo-Mensah, and W. Chen, “Transformer models for text-based emotion detection: a review of bert-based approaches,” *Artificial Intelligence Review*, pp. 1–41, 2021.

[5] H. Al-Omari, M. A. Abdullah, and S. Shaikh, “Emodet2: Emotion detection in english textual dialogue using bert and bilstm models,” in *2020 11th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2020, pp. 226–232.

[6] F. Ding, X. Kang, S. Nishide, Z. Guan, and F. Ren, “A fusion model for multi-label emotion classification based on bert and topic clustering,” in *International Symposium on Artificial Intelligence and Robotics 2020*, vol. 11574. International Society for Optics and Photonics, 2020, p. 115740D.

[7] C. Huang, A. Trabelsi, and O. R. Zaïane, “Ana at semeval-2019 task 3: Contextual emotion detection in conversations through hierarchical lstms and bert,” *arXiv preprint arXiv:1904.00132*, 2019.

[8] P. Zhong, D. Wang, and C. Miao, “Knowledge-enriched transformer for emotion detection in textual conversations,” 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)*. Association for Computational Linguistics, nov 2019, pp. 165–176.

Emotion	Augmentation	One-vs-all SO BERT F1	Multi-label SO BERT F1	GoEmotion F1	Emotxt F1	Support
Anger	None	78%	**80%**	63%	68%	235
Fear	Contextual Word Embedding	30%	**59%**	54%	6%	32
Joy	None	51%	**56%**	50%	38%	152
Love	Synonym	81%	**84%**	82%	69%	385
Sadness	None	50%	60%	**62%**	52%	68
Surprise	Contextual Word Embedding	**35%**	26%	26%	23%	17
Average		54%	**60.8%**	56%	42%	

TABLE III

PERFORMANCE OF ONE-VS-ALL AND MULTI-LABEL SO BERT CLASSIFIERS VS GOEMOTION AND EMOTXT FOR EACH EMOTION CATEGORY

[9] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, and S. Ravi, “Goemotions: A dataset of fine-grained emotions,” *arXiv preprint arXiv:2005.00547*, 2020.

[10] N. Novielli, F. Calefato, and F. Lanubile, “A gold standard for emotion annotation in stack overflow,” in *2018 IEEE/ACM 15th International Conference on Mining Software Repositories (MSR)*. IEEE, 2018, pp. 14–17.

[11] G. Parrot, “Emotions in social psychology–psychology press,” 2000.

[12] R. Jongeling, S. Datta, and A. Serebrenik, “Choosing your weapons: On sentiment analysis tools for software engineering research,” in *2015 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, 2015, pp. 531–535.

[13] P. Tourani, Y. Jiang, and B. Adams, “Monitoring sentiment in open source mailing lists: exploratory study on the apache ecosystem.” in *CASCON*, vol. 14, 2014, pp. 34–44.

[14] B. Lin, F. Zampetti, G. Bavota, M. Di Penta, M. Lanza, and R. Oliveto, “Sentiment analysis for software engineering: How far can we go?” in *Proceedings of the 40th International Conference on Software Engineering*, 2018, pp. 94–104.

[15] N. Novielli, F. Calefato, F. Lanubile, and A. Serebrenik, “Assessment of se-specific sentiment analysis tools: An extended replication study,” *arXiv preprint arXiv:2010.10172*, 2020.

[16] M. R. Islam and M. F. Zibran, “Leveraging automated sentiment analysis in software engineering,” in *2017 IEEE/ACM 14th International Conference on Mining Software Repositories (MSR)*. IEEE, 2017, pp. 203–214.

[17] G. Uddin and F. Khomh, “Automatic mining of opinions expressed about apis in stack overflow,” *IEEE Transactions on Software Engineering*, 2019.

[18] T. Ahmed, A. Bosu, A. Iqbal, and S. Rahimi, “Senticr: A customized sentiment analysis tool for code review interactions,” in *2017 32nd IEEE/ACM International Conference on Automated Software Engineering (ASE)*, 2017, pp. 106–111.

[19] T. Zhang, B. Xu, F. Thung, S. A. Haryono, D. Lo, and L. Jiang, “Sentiment analysis for software engineering: How far can pre-trained transformer models go?” in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 70–80.

[20] F. Calefato, F. Lanubile, F. Maiorano, and N. Novielli, “Sentiment polarity detection for software development,” *Empirical Software Engineering*, vol. 23, no. 3, pp. 1352–1382, 2018.

[21] H. Batra, N. S. Punn, S. K. Sonbhadra, and S. Agarwal, “Bert based sentiment analysis: A software engineering perspective,” *arXiv preprint arXiv:2106.02581*, 2021.

[22] E. Biswas, M. E. Karabulut, L. Pollock, and K. Vijay-Shanker, “Achieving reliable sentiment analysis in the software engineering domain using bert,” in *2020 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. IEEE, 2020, pp. 162–173.

[23] A. Murgia, M. Ortu, P. Tourani, B. Adams, and S. Demeyer, “An exploratory qualitative and quantitative analysis of emotions in issue report comments of open source systems,” *Empirical Software Engineering*, vol. 23, no. 1, pp. 521–564, 2018.

[24] M. R. Islam and M. F. Zibran, “Deva: sensing emotions in the valence arousal space in software engineering text,” in *Proceedings of the 33rd annual ACM symposium on applied computing*, 2018, pp. 1536–1543.

[25] M. R. Islam, M. K. Ahmmed, and M. F. Zibran, “Marvalous: machine learning based detection of emotions in the valence-arousal space in software engineering text,” in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, 2019, pp. 1786–1793.

[26] F. Calefato, F. Lanubile, and N. Novielli, “Emotxt: a toolkit for emotion recognition from text,” in *2017 seventh international conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW)*. IEEE, 2017, pp. 79–80.

[27] I. Beltagy, K. Lo, and A. Cohan, “Scibert: A pretrained language model for scientific text,” *arXiv preprint arXiv:1903.10676*, 2019.

[28] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, and J. Kang, “Biobert: a pre-trained biomedical language representation model for biomedical text mining,” *Bioinformatics*, vol. 36, no. 4, pp. 1234–1240, 2020.

[29] W. Tai, H. Kung, X. L. Dong, M. Comiter, and C.-F. Kuo, “exbert: Extending pre-trained models with domain-specific vocabulary under constrained training resources,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2020, pp. 1433–1439.

[30] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, and N. Smith, “Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping,” *arXiv preprint arXiv:2002.06305*, 2020.