m_y at SemEval-2019 Task 9: Exploring BERT for Suggestion Mining

Masahiro Yamamoto
Sony Corporation, Tokyo, Japan.
Masahiro.A.Yamamoto@sony.com

Toshiyuki Sekiya
Sony Corporation, Tokyo, Japan.
Toshiyuki.Sekiya@sony.com

Abstract

This paper presents our system to the SemEval-2019 Task 9, Suggestion Mining from Online Reviews and Forums. The goal of this task is to extract suggestions such as the expressions of tips, advice, and recommendations. We explore Bidirectional Encoder Representations from Transformers (BERT) focusing on target domain pre-training in Subtask A which provides training and test datasets in the same domain. In Subtask B, the cross domain suggestion mining task, we apply the idea of distant supervision. Our system obtained the third place in Subtask A and the fifth place in Subtask B, which demonstrates its efficacy of our approaches.

1 Introduction

In SemEval2019 Task 9, participants are required to build a model which can classify given sentences into suggestion and non-suggestion classes. We participate in two sub-tasks: domain specific suggestion mining task (Subtask A[1]) and cross domain suggestion mining task (Subtask B[2]). In Subtask A, the test dataset belongs to the same domain as the training and development datasets. These datasets are extracted from the suggestion forum for windows platform. In Subtask B, training and test datasets belong to separate domains. More specifically, the domain of the training dataset is entries from the windows forum and that of the test dataset is hotel reviews. Example sentences used in these tasks are listed in Table 1. For a description of these tasks please refer to (Negi et al., 2019).

Subtask A (windows platform)
P: xbox dev mode companion work ipv6 please...
N: I do not want to convert MSIs.
Subtask B (hotel review)
P: If you can, upgrade to an Ocean Front Room.
N: it doesn't have a very clean look.

Table 1: Example sentences in the test dataset. **P** means a positive sentence, i.e. suggestion sentence and **N** denotes a negative sentence, i.e. non-suggestion sentence.

Subtask A can be viewed as a binary text classification task. Recently, pre-training models, such as OpenAI GPI (Radford et al., 2018) and BERT (Devlin et al., 2018), have gained much attention with their ability to improve a number of downstream tasks. These models are pre-trained using unlabeled corpora and then fine-tuned on labeled datasets. We apply BERT to this task because it has achieved state-of-the-art performance in several text classification tasks. The difference to the original BERT model is that we further pre-train BERT model using an unlabeled corpus related to this domain. More concretely, we extract documents from a windows forum and run additional steps of pre-training using these documents, starting from the pre-trained BERT model.

In Subtask B, we apply the idea of distant supervision which has been firstly proposed by (Mintz et al., 2009). Distant supervision is a weakly supervised learning framework which tries to automatically generate noisy training examples. Specifically, we use the rule based system which is provided by the task organizer for creating a noisy training dataset and train the model based on them.

Our system significantly outperforms baseline methods on two subtasks. These results demonstrate its efficacy of our approaches, target domain pre-training in Subtask A and distant supervision in Subtask B.

[1] https://competitions.codalab.org/competitions/19955
[2] https://competitions.codalab.org/competitions/19956

Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019), pages 888–892
Minneapolis, Minnesota, USA, June 6–7, 2019.

2 System Description

Our model is built using a recent development of pre-training model, BERT. In the following, we describe details of this model first and then explain our systems in Subtask A and Subtask B.

2.1 BERT

BERT, proposed by (Devlin et al., 2018), has been shown to improve several tasks such as sentiment classification, calculating semantic textual similarity task, and recognizing textual entailment task. This model consists of several Transformer models (Vaswani et al., 2017) whose parameters are pre-trained on unlabeled corpora, Wikipedia and BooksCorpus (Zhu et al., 2015). Pre-training consists of two tasks, masked language modeling and next sentence prediction, and trained models of these unsupervised tasks are available.[3]

For the classification task, we added one layer to output predictions, suggestion or non-suggestion. These model parameters were then fine-tuned based on the labeled training dataset.

2.2 Subtask A

Subtask A is a classical document classification task, where training/development/test datasets consist of the same domain. The data has been collected from feedback posts on universal windows platform. We mainly use BERT in this task, however, in the following we describe several techniques to improve the score.

2.2.1 Target Domain Pre-training

The major difference between BERT and our system is the target domain pre-training. Typically, BERT training consists of two parts: pre-training on the general domain corpus and fine-tuning on the target task. On the other hand, our system consists of three training steps: pre-training on the general domain corpus, pre-training on the target domain corpus, and fine-tuning on the target task. More concretely, we further pre-train the model using a task specific unlabeled corpus scraped from the universal windows platform developer feedback site.[4] These documents are split into sentences and the model is then trained on two unsupervised tasks (i.e. masked language modeling and next sentence prediction). We should note here that we use officially provided pre-trained parameters as initial model parameters and further train the model based on target domain documents to obtain better network parameters.

This target domain pre-training can be considered as a similar framework of Universal Language Model Fine-Tuning (ULMFiT), proposed by (Howard and Ruder, 2018). In their framework, the model is firstly trained on general domain corpus to capture general features. In addition, the model is further trained on target task data to learn task specific features, and fine-tuned on the target task. This procedure is similar to our system. In other words, our work can be viewed as the extension of BERT model by using the idea of ULMFiT.

2.2.2 Model Averaging

(Reimers and Gurevych, 2017) showed that training deep neural network is sensitive to the initial weights and (Che et al., 2018) showed the effectiveness of ensembling models trained with different initialization. Furthermore, (Devlin et al., 2018) empirically showed that ensembling BERT models trained with different pre-training checkpoint leads to performance improvement. We follow this work and train three models with different pre-training checkpoint, then ensemble these models by simply averaging output scores.

2.3 Subtask B

Subtask B is cross-domain suggestion mining, where train and development datasets belong to windows platform domain, while the test dataset belongs to the hotel review domain. In this task, we do not use the datasets provided in Subtask A, because we find that models trained on these datasets tend to have poor performance (see also Sec. 3.2.2). We instead apply the paradigm of distant supervision.

Distant supervision is a framework to generate noisy annotated data automatically and use them as a training dataset. This idea has been firstly proposed by (Mintz et al., 2009) and well studied in the field of relation extraction (Riedel et al., 2010; Hoffmann et al., 2011).

2.3.1 Rule Based Labeling

For distant supervision, the initial labeling method is needed. In this work, we make use of the officially provided baseline method as an initial labeling tool. This system is based on a rule based

[3] https://github.com/google-research/bert

[4] https://wpdev.uservoice.com/forums/110705-universal-windows-platform

	Subtask A			Subtask B		
	Train	Dev	Test	Train	Dev	Test
Suggestions	2,085	296	87	-	404	348
Non Suggestions	6,415	296	746	-	404	476
All	8,500	592	833	-	808	824

Table 2: Statistics of datasets.

method. For example, if a specific word such as "suggest" is in the sentence, then the sentence is predicted as a suggestion sentence. For more details, please see the code in the official repository.[5]

2.3.2 Model Training Procedure

After the labeled corpus is generated, we trained a model using this corpus. Here, we do not label all of the unlabeled sentences via the rule based system. Instead, we split the sentences into several pieces and label one of them by using the rule based system. A model is trained on the noisy labeled dataset and we label another piece through the trained model. We apply this procedure iteratively until the model is trained on the last piece. More specifically, our training procedure can be summarized as follows:

1. We prepare the unlabeled hotel review corpus and split it into N pieces (corpus C_1, C_2, C_3, ..., C_N).
2. We apply the baseline method which is provided by the task organizer to the corpus C_1 and treat predicted labels as true labels.
3. The model is then trained using the labeled corpus C_1'.
4. We apply the trained model to the corpus C_2 and treat predicted labels as true labels.
5. A new model is trained using the labeled corpus C_2' and labels of sentences in C_3 are predicted by this model.
6. We iteratively apply the above procedure until the model is trained on the corpus C_N.

3 Experiments

Table 2 shows the statistics of datasets which are used in Subtask A and Subtask B. These datasets are available at the official repository.[6] For evaluating systems, F1 score for the positive class, i.e. the suggestion class, is employed.

[5]https://github.com/Semeval2019Task9/Subtask-B/blob/master/semeval-task9-baseline.py

[6]https://github.com/Semeval2019Task9

System	Dev F1	Test F1
Baseline	0.721	0.267
BERT BASE (Single)	0.845	0.731
BERT BASE (Sgl. + Tgt.)	0.866	0.755
BERT LARGE (Single)	0.867	0.737
BERT LARGE (Sgl. + Tgt.)	0.882	0.759
BERT LARGE (Ens. + Tgt.)	0.890	0.776

Table 3: Results of Subtask A. **Single** or **Sgl.** denotes the single model and **Ens.** means the ensembled models. **Tgt.** denotes the pre-training on the target domain corpus.

3.1 Subtask A

3.1.1 Settings

We employed the official BERT model. We used both BASE model which has 12 Transformer layers, 12 self-attention heads, and 768 hidden size, and LARGE model which has 24 Transformer layers, 16 self-attention heads, and 1024 hidden size. There are 110 million parameters in total in BASE model and 340 million parameters in LARGE model.

As for the target domain pre-training, we obtained the windows review corpus and split it into sentences using NLTK.[7] The number of scraped documents is 2,325 and we used these documents as the unlabeled corpus for further pre-training the BERT model.

3.1.2 Results and Discussions

Table 3 shows the results of the experiment. Here, the baseline method is a rule based method as explained in Sec. 2.3.1. From Table 3, we can conclude the following four things:

First, BERT LARGE model outperformed BERT BASE model despite of the small size of the dataset. In general, it has been known that increasing the model size leads to an improvement on large scale tasks such as machine translation, and this does not be applied to small scale tasks

[7]https://www.nltk.org/

except for (Devlin et al., 2018). They showed a similar tendency in another small scale task. These results demonstrate that large size models improve results not only on large scale tasks but also on small scale tasks, if the model has been well pre-trained.

Second, the effect of the target domain pre-training is not trivial. The improvement can be observed in both BASE and LARGE model. In BASE model, the F1 score is pushed from 0.845 to 0.866 in the development dataset and from 0.731 to 0.755 in the test dataset. In LARGE model, the score is improved from 0.867 to 0.882 in the development dataset and from 0.737 to 0.759 in the test dataset. These results show that better models are produced by target domain pre-training even if we do not have large, domain-specific documents.

Third, ensembling models leads to further performance improvement.

Fourth, Test F1 score is much lower than Dev F1 score. This has also been observed by other teams. In concrete, many teams have achieved over 0.850 F1 score on the development dataset, however, no team has achieved over 0.800 F1 score on the test dataset. It might have something to do with the small size of the development dataset size or the difference in the label distribution.

3.2 Subtask B

3.2.1 Settings

In Subtask B, we got unlabeled hotel review documents provided by (Wachsmuth et al., 2014).[8] We split documents into sentences using the NLTK package and randomly extracted 500,000 sentences. These sentences were further split into 5 pieces.

We firstly ran the baseline method and automatically labeled 100,000 sentences. These sentences were used as a training dataset to train the machine learning model. We used the BERT BASE model and set the default hyperparameters except for the training epochs. To avoid over-fitting, we trained the model for only one epoch.

As described in Sec. 2.3, another 100,000 sentences were automatically labeled using the trained model and we trained a new model using this labeled data. We iteratively applied this procedure and submitted results predicted by the final model.

[8] http://argumentation.bplaced.net/arguana/data

System	Dev F1	Test F1
Baseline	0.774	0.732
BERT BASE (windows)	0.308	0.419
BERT BASE (1st model)	0.800	0.785
BERT BASE (5th model)	0.817	0.793

Table 4: Results of Subtask B. **BERT BASE (windows)** is the model trained on windows corpus provided in Subtask A.

3.2.2 Results and Discussions

Table 4 shows the results of the experiment. As you can see in Table 4, BERT BASE (windows), the model trained on the other domain corpus, shows poor performance while the baseline method has achieved a much better score. This motivated us to apply the idea of distant supervision.

The model based on distant supervision significantly outperformed the baseline model. This result shows that the distant supervision idea can be applied successfully in the cross domain suggestion mining task. Furthermore, we can see that our iterative approach leads to more performance improvement (from 0.800 to 0.817 in the development dataset and 0.785 to 0.793 in the test dataset). We currently do not know why the F1 score has been improved, however, one interpretation could be that our iterative framework avoids over-fitting to the one model and learns more general decision boundaries. A detailed study of this effect is future work.

4 Conclusion

This paper explains our submission to SemEval2019 Task 9, Subtask A and Subtask B. We explored BERT models focusing on the target domain pre-training in Subtask A and the idea of the distant supervision in Subtask B. Our approach obtained the third place in Subtask A and the fifth place in Subtask B.

In the future, we will further investigate the effect of these approaches in other tasks.

References

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. *arXiv preprint arXiv:1807.03121*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 328–339.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Sapna Negi, Tobias Daudert, and Paul Buitelaar. 2019. Semeval-2019 task 9: Suggestion mining from online reviews and forums. In *Proceedings of the 13th International Workshop on Semantic Evaluation (SemEval-2019)*.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. 2018. Improving language understanding with unsupervised learning. Technical report, OpenAI.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. *arXiv preprint arXiv:1707.09861*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Henning Wachsmuth, Martin Trenkmann, Benno Stein, Gregor Engels, and Tsvetomira Palakarska. 2014. A review corpus for argumentation analysis. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 115–127. Springer.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.