YNU-HPCC at SemEval-2021 Task 10: Using a Transformer-based Source-Free Domain Adaptation Model for Semantic Processing

Zhewen Yu, Jin Wang and Xuejie Zhang
School of Information Science and Engineering
Yunnan University
Kunming, China
zhwyu@mail.ynu.edu.cn, {wangjin, xjzhang}@ynu.edu.cn

Abstract

Data sharing restrictions are common in natural language processing datasets. The aim of this study is to develop a model that is trained in a source domain to make predictions for a target domain with respect to domain data. To address this problem, the organizers provided models that fine-tuned a large number of source domain data on pre-trained models and dev data for participants. However, source domain data were not distributed. This paper describes the model provided for the name entity recognition task and ways to develop the model. Because little data are provided, pre-trained models are suitable for solving cross-domain tasks. The models fine-tuned by a large number of other domains could be effective in the new domain because the task did not change. The code of this paper is available at https://github.com/windforfurture/SemEval-2021-Task10.

1 Introduction

Data sharing constraints are common in natural language processing (NLP) datasets. For example, Twitter policies prohibit the sharing of tweet text, although tweet IDs may be shared. In clinical NLP, the situation is even more prevalent because information on patients' health must be protected. Obtaining annotations about health texts often requires the signing of complex data usage agreements. During the competition, the organizers provided models which fine-tuned on the annotated source domain data, while the source domain data could not be distributed. The organizers also provided labeled data as dev data and unlabeled target domain data as test data.

There were two sub-tasks for SemEval Task 10: 1) to classify clinical event mentions (e.g., diseases, symptoms, procedures, etc.) for whether they are being negated by their context. This is a text-classification task (Yuan et al., 2020). 2) to find time expressions (Laparra et al., 2018) in text, this is a sequence-tagging task. SemEval Task 10 was different from traditional NLP tasks, where training and testing were in the same domain. Predictions could be out of control due to the different target domain. The domain of the dev data is related to the source and target domains, therefore the model can be developed. In SemEval Task 10, it is necessary to develop an existing model along with training the model with labeled data or unlabeled data. Furthermore, the model can be developed by fine-tuning it in different ways.

For certain reasons, only the results for the second subtask were submitted, that is, time expression recognition, which can be considered as a name entity recognition (NER) task. The participants were asked to find time expressions in the text through the task. This is a sequence-tagging task that uses fine-grained time expression annotations that are a component of SemEval 2018 Task 6(Laparra et al., 2018). To deal with this task, deep learning models, such as long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997), bidirectional LSTM with conditional random field (BiLSTM-CRF) (Huang et al., 2015), and bidirectional encoder representation from transformers (BERT) (Dai et al., 2019) have been developed to challenge the NER task. Owing to the lack of a training set, pre-trained models (Qiu et al., 2020) were considered. The NER task can become difficult as the number of classifications increase. In this task, 33 types of entities need to be recognized, and phrases and words may be one of the types. For the base model, B-prefix and I-prefix were used to discriminate the beginning of classification and ensure the accuracy of the tokens; therefore, 65 types of entities were used in the model. In this paper, an ensemble model is proposed for time

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 1289–1294
Bangkok, Thailand (online), August 5–6, 2021.

expression recognition with a hard voting strategy. Each input sample was first tokenized using the matched model tokenizer. Base models, such as BERT (Devlin et al., 2019) and its variants, including RoBERTa (Liu et al., 2019), DistilBERT (Sanh et al., 2019) and ALBERT (Lan et al., 2019), were used to learn hidden representations for each token. Then, a fully connected dense layer with a $Softmax$ function was used for classification. By using a hard voting strategy, the predictions from different base models were merged with the final result. Experimental results show that the proposed ensemble models achieve a competitive result with the official baseline model, which ranked fifth in sub-task 2.

The remainder of this paper is organized as follows. Section 2 describes the overall structure of the proposed ensemble model. The comparative experimental results and discussion are presented in Section 3. Finally, conclusions are presented in Section 4.

2 Ensemble Model for Source-Free Domain Adaptation

The official baseline model is a fine-tuned RoBERTa model: clulab/roberta-timex-semeval. It uses RoBERTa (Liu et al., 2019) for token classification architecture, which is pre-trained using no next-sentence prediction (NSP) and dynamic masking. The RoBERTa model achieved better performance than the BERT model because it was pre-trained by larger volume data, more steps, larger batches, and larger vocabulary than the BERT model. In addition, the provided baseline model was fine-tuned with approximately 25,000 expressions in de-identified clinical notes as well as the development dataset (Sanh et al., 2019). Four other models were also implemented using a hard voting strategy and a hard voting result for a single submission.

2.1 Tokenization

Transforming words to vectors is a necessary step in NLP tasks. Different word representations were used in our implementation, including word2vec (Mikolov et al., 2013; CHURCH, 2017), GloVe (Pennington et al., 2014), ELMo (Peters et al., 2018), and BERT (Devlin et al., 2019). For the RoBERTa model, we used only the matched RoBERTa tokenizer to build word vectors with a length of 514. Given a sentence $\mathbf{x} = [x_1, x_2, \ldots, x_M]$ of length M, different lengths of the sentence will result in different lengths of the representation. Therefore, we considered the maximum sentence length as $N-1$. If the length was less than $N-1$, then it was padded with zero values to make it equal to $N-1$. For each input, two specific tokens were added to the beginning and end of the raw input, that is, <s> and </s> (or [CLS] and [SEP]). The RoBERTa tokenizer uses byte-pair encoding to obtain more relations and meanings. Then, these are converted into a sequence of subwords, which are then mapped into token, position, and segment embeddings, that is, $[E_{[CLS]}, E_1, \ldots, E_{N-1}, E_{[SEP]}]$. In the proposed model shown in Fig.1, the raw inputs were split into one or more 514-dimensional vectors according to the number of sentences contained in the corresponding input.

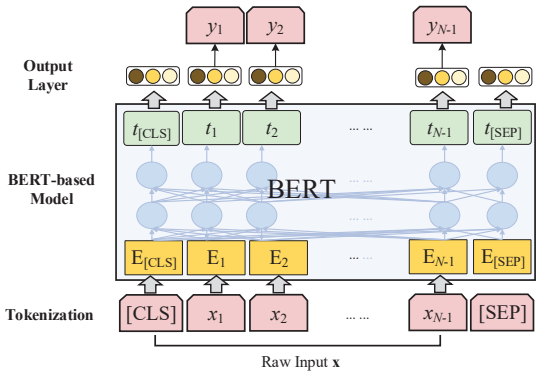

Figure 1: Overall architecture of the base model.

2.2 BERT-based Model

To extract the semantic features, a pre-trained language model(Qiu et al., 2020) was used. It achieved impressive performance in various NLP tasks. It contains multiple layers of bidirectional transformer encoders(Vaswani et al., 2017) and is then pre-trained by using unsupervised learning of either the masked language model (with a masked ratio of 15%) or the NSP.

The aforementioned pre-trained language model contains 12 layers of transformers with a hidden size of 768. Then, the embeddings of both contexts are fed into a BERT model to obtain the semantic representation $T \in \mathbb{R}^{d_t}$, denoted as

$$\begin{aligned} T^F &= [t_{[CLS]}, t_1, \ldots, t_{N-1}, t_{[SEP]}] \\ &= f_{BERT}([E_{[CLS]}, E_1, \ldots, \\ &\qquad E_{N-1}, E_{[SEP]}]; \theta_{BERT}) \quad (1) \end{aligned}$$

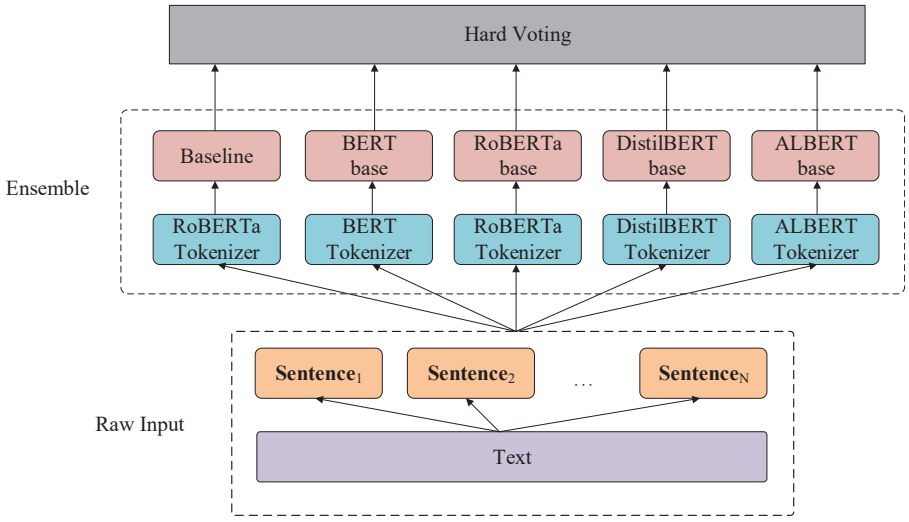

Figure 2: The ensemble of models.

where θ_{BERT} is the trainable parameter of the BERT model, which is then fine-tuned during model training, and $d_t = 768$ is the dimensionality of the hidden representation.

2.3 Output Layer

The token classification in the BERT model architecture is considered as a series of binary classifications. For each token, the classification was a one-layer MLP with a $Softmax$ function. The loss function is a categorical cross-entropy, defined as:

$$\hat{y}_m^c = softmax(W_t t_m + b_t) \quad (2)$$

$$L = -\sum_{k=1}^{K}\sum_{c=1}^{C}\sum_{m=1}^{M} y_m \log(\hat{y}_m^c) \quad (3)$$

where y_m and $\hat{y}_m^c$ denote the gold label and the predicted probability of samples k, respectively, K and C are the number of training samples and candidate categories, respectively; W_t and b_t are the weight and bias, respectively, which are associated with the fully connected layer. The entire network was trained by back-propagation (Rumelhart et al., 1986) while the BERT model was fine-tuned with the provided labeled data in the training phase.

2.4 Ensemble Learning

To output the final result, we used a hard voting strategy to integrate the results from different base models, including the official baseline, BERT, RoBERTa, DistilBERT, and ALBERT models, as shown in Fig. 2. In hard voting, every individual classifier votes for a class and the majority wins. In statistical terms, the predicted target label of the ensemble is the mode of the distribution of the individually predicted labels.

3 Experimental Results

3.1 Datasets

The organization provided 99 news articles and matched 99 annotated data files for subtask 2. There were different types in these articles, such as ABC, APW, and CNN. The dev data provided were related to news, and the source data were related to medical data. The test data were related to the food security. The cross-domain task consisted of three domains. Each raw input was divided into sentences. The annotated data files were formed as XML files. Every XML file contained a few entities that consisted of ids, spans, and types. Span contains the start position, and the end positions and types are classified. The labels should be transformed because of the offset that were used as the input data.

3.2 Evaluation Metrics

Subtask 2 was evaluated using standard precision, recall, and mainly $the\ F_1$-score. The F_1-score is often used to evaluate unbalanced data, and is defined as follows:

$$F_1\text{-score} = 2 * \frac{P * R}{(P + R)} \quad (4)$$

where P and R denote the precision and recall, respectively. A higher F_1-score indicates better model prediction performance.

Model	Vocab Size	Position Embeddings	Attention Heads	Hidden Layers
clulab/roberta-timex-semeval	50265	514	12	12
BERT base	30522	512	12	12
RoBERTa base	50265	514	12	12
DistilBERT base	30522	512	12	6
ALBERT base	30000	512	12	12

Table 1: Optimal parameter settings of the base models

Model	R1	R2	R3	R4	R5	Avg.
clulab/roberta-timex-semeval	**0.864**	**0.910**	0.883	0.857	0.859	**0.875**
BERT base	0.844	0.880	0.895	0.847	0.837	0.861
RoBERTa base	0.817	0.850	0.843	0.809	0.823	0.828
DistilBERT base	0.828	0.868	0.874	0.837	0.839	0.849
ALBERT base	0.809	0.811	0.752	0.792	0.774	0.788
Hard Voting	0.856	0.889	**0.900**	**0.866**	**0.860**	0.874

Table 2: Empirical results of the ensemble model and the base models

3.3 Implementation Details

We fine-tuned the official baseline model (RoBERTa) and four other models: the BERT model (bert-base-uncased), RoBERTa model (roberta-base), DistilBERT model (distilbert-base-uncased), and ALBERT model (albert-base-v2). For each model, 5-fold cross-validation was performed. For each run, the datasets were split into a ratio of 8:2 for the training and dev sets.

We trained the new model by adding the same label2id and id2lable JSON as the provided model to the original "config.json" We ran codes using the downloaded model and the modified "config.json." After training, the models were used for prediction, and the results were recorded. After comparing the results, we observed that the model provided the best performance. Then, we performed hard voting that picked out one result, if three or more results of these models were the same.

To obtain additional results, we split the provided data to obtain different dev data and test data five times, and the test data were different every time. In every round, the initial models were used, and five rounds were tested. Hard voting gave the best results, followed by the provided model. Finally, we fine-tuned these initial models with the test data in the evaluation phase and chose the results of the provided model and the hard voting for submissions.

3.4 Fine-tuning the Parameters

To train the proposed model, the Adam optimizer applied a warmup strategy with a weight decay of 0.01 during training. The learning rates of all the base models were 5e-6. For the official baseline model, the default parameters were used in our experiments because the initial parameters usually performed well in other experiments. For other models, we fine-tuned the hyperparameters using a grid-search strategy. Once the optimal settings of the parameters were obtained, they were used for classification on the test sets. The details of the hyperparameters are summarized in Table 1.

3.5 Comparative Results

Table 2 shows the results of the base models on different folds, that is, R1–R5. The results showed that the hard-voting ensemble model outperformed the official baseline model with three rounds, but the average F_1-score of the ensemble model was less than the average F_1-score of the official baseline model. Considering that the evaluation data is in a new domain, we finally submitted the result of the official baseline model, which was fine-tuned by feeding numerous annotated data in the source domain and the result of the hard voting ensemble model. For the test set, the newly released official baseline model, that is, **Organizers (new)**, outperformed the previously released baseline model, that is, **Organizers (previous)**, which was only pretrained on the source data and achieved an F_1-score of 0.794(see Table 3). Owing to the low amount of

Team Name	F_1-score
BLCUFIGHT-1	0.815
Self-Adapter-1	0.811
BLCUFIGHT-2	0.810
Baseline-2	0.804
YNU-HPCC-2	**0.803**
Self-Adapter-2	0.797
PTST-UoM-1	0.796
UArizona-1	0.795
UArizona-2	0.795
Boom-1	0.795
Baseline-1	0.794
KISNLP-1	0.793
KISNLP-2	0.781
YNU-HPCC-1	**0.748**

Table 3: All results on leaderboard for time expression recognition.

data used for fine-tuning, the performance of the proposed model is slightly lower than that of the new official baseline model. However, its performance is still competitive and finally ranked fifth on the leaderboard.

4 Conclusions

The SemEval-2021 Task 10 framework requests participants to develop semantic annotation systems in the face of data sharing constraints. In this study, we fine-tuned the official baseline model and then combined it with four other pre-trained models with a hard voting strategy for time expression recognition. Experimental results showed that the proposed model outperformed the previously released baseline model, achieved a competitive result with the newly released official baseline model, and finally ranked fifth on the leaderboard. Future work will attempt to improve the performance of cross-domain NER tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants Nos. 61702443, 61966038 and 61762091.

References

KENNETH WARD CHURCH. 2017. Word2Vec. *Nat. Lang. Eng.*, 23(1).

Zhenjin Dai, Xutao Wang, Pin Ni, Yuming Li, Gangmin Li, and Xuming Bai. 2019. Named Entity Recognition Using BERT BiLSTM CRF for Chinese Electronic Health Records. *Proc. - 2019 12th Int. Congr. Image Signal Process. Biomed. Eng. Informatics, CISP-BMEI 2019*, (October 2020).

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, 1(Mlm):4171–4186.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Comput.*, 9(8).

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging.

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*, pages 1–17.

Egoitz Laparra, Dongfang Xu, Ahmed Elsayed, Steven Bethard, and Martha Palmer. 2018. SemEval 2018 Task 6: Parsing Time Normalizations. pages 88–96.

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*, (1).

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations ofwords and phrases and their compositionality. *Adv. Neural Inf. Process. Syst.*, pages 1–9.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global vectors for word representation. In *EMNLP 2014 - 2014 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf.*

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *NAACL HLT 2018 - 2018 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. - Proc. Conf.*, 1:2227–2237.

Xi Peng Qiu, Tian Xiang Sun, Yi Ge Xu, Yun Fan Shao, Ning Dai, and Xuan Jing Huang. 2020. Pre-trained models for natural language processing: A survey. *Sci. China Technol. Sci.*, 63(10):1872–1897.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088).

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv*, pages 2–6.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Adv. Neural Inf. Process. Syst.*, 2017-Decem(Nips):5999–6009.

Li Yuan, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. 2020. Graph Attention Network with Memory Fusion for Aspect-level Sentiment Analysis. *Proc. 1st Conf. Asia-Pacific Chapter Assoc. Comput. Linguist. 10th Int. Jt. Conf. Nat. Lang. Process.*, pages 27–36.