Knowledge distillation and data augmentation for NLP light pre-trained models

Hanwen Luo1*, Yudong Li2, Xiaodong Wang1 and Yuqing Zhang2

1Department of computer science and technology, National University of Defense Technology, Changsha, Hunan, 410000, China
2Department of computer science and technology, China University of Geosciences (Beijing), Beijing, 100000, China
*Corresponding author’s e-mail: luohanwen@nudt.edu.cn

Abstract. Model lightweight aims to solve the problems of various large models for slow training and high resource requirement. Knowledge distillation can be a good solution to these problems. We built a lightweight model that meet the competition requirements and have prominent NLP capabilities. RoBERTa-tiny-clue was used as our backbone model. We tested the effect of soft labels and hard labels on knowledge distillation, made knowledge distillation, fine-tuned this model to get a lighter model with better performance, and then applied it downstream NLP tasks. We also adopted a series of data augmentation methods to improve the performance of the model on downstream tasks, customized different optimization solutions for four tasks. Based on open-source pre-trained model RoBERTa-tiny-clue and public available datasets, we achieved 15 times smaller and 10 times faster than BERT-base, and 95% of BERT-base performance on downstream NLP tasks. Using suitable data augmentation methods for the trained lightweight model, the performance of the model on various downstream tasks reaches or exceeds BERT-base.

1. Introduction
Based on the Transformer architecture, many pre-trained language models began to emerge in 2018, refreshing many NLP tasks. From BERT to RoBERTa, and the improvement of GPT to GPT2, it has been proved that using more and more comprehensive data can run a more robust and versatile pre-training model. Nvidia, Google, Microsoft, and Open-AI have successively released Megatron LM (8.3 billion parameters), T5 (11 billion), Turing NLG (17 billion) and GPT3 (175 billion), also achieved amazing performance on various NLP tasks. Various improvements based on language models are endless. Although the methods are different, the overall idea is a large-scale corpus pre-training and finetune method.

At present, the research and training of large pre-trained models are still among the leading research goals. But for ordinary companies and scientific research institutions, data support and hardware conditions have become a constraint. Some problems cannot be avoided in large pre-training models. On the one hand, they required lots of computational and storage resources in the training phase and service stage separately. On the other hand, the training model takes a long time, and the reasoning speed is slow. In practical applications, large models require high deployment resources. If the deployment resources are reduced, the company or enterprise may not be able to meet the requirements of QPS and P99. The above two shortcomings limit the application of large models in the industry. Therefore, some
researchers work for model lightweight. This subject aims to achieve an effect close to that of a large model with as few parameters as possible while increasing the speed of training and prediction.

In this paper, the light pre-trained model is analyzed and studied. We propose a solution and optimization scheme from light pre-trained model to downstream tasks. We improved the performance of RoBERTa-tiny-clue[1] in downstream NLP tasks by the knowledge distillation, multi-task learning, same original data augmentation and BERT cloze data augmentation. Through multiple comparison experiments, it is found that these methods are efficient on different tasks. By selecting the best-performing solution in various tasks, the score of the light pre-training model on the CLUE benchmark[2] has improved 9.5% than RoBERTa-tiny-clue baseline. In the article structure, we introduce our work from two aspects of knowledge distillation and data augmentation, and summarize this article in the last part.

2. Related work
The article is a model-lightening study for natural language processing tasks, so it chooses to present both small model pre-training and downstream NLP tasks.

2.1. Small model pre-training
In recent years, language model (LM) pre-training has achieved remarkable success for various natural language processing tasks. Especially since 2018, Various large models constantly refresh the leaderboards of NLP tasks. Conventional pre-trained language models, such as BERT and its variants, use large-scale text corpora to predict words in a given context to learn the contextualized text representation pass additional. The task-specific layers are fine-tuned to suit downstream tasks. However, these models often contain a large number of parameters, which poses challenges for fine-tuning and online services for latency and capacity limitations in real-world applications. Although training a large parameter model can achieve better performance in downstream tasks, it inevitably produces problems such as slow training/inference and high requirements for deployment resources. Therefore, the researchers are committed to reducing the model parameters while maintaining the performance of the model.

Based on the above considerations, some NLP researchers tried to improve the neural network structure and design a simplified model. For example, ALBERT[3] achieved the purpose of reducing the model by decomposing parametric embedding and cross-layer parameter sharing. Another part of the researchers use the knowledge distillation (KD)[4-5] technology to compress large pre-trained models into small models. Knowledge distillation is a model compression and training method based on the teacher-student network, making the model achieve faster performance while achieving similar performance to the large model. Models like DistillBERT[6], TinyBERT[7], FastBERT[8] and MiniLM[9] are all formed by improving these methods. DistillBERT uses BERT-base as the teacher network. The number of network layer is reduced from the original 12 layers to 6 layers, soft label and hidden layer parameters of the teacher model are used to train the student model. Compared with BERT-base, the model size is reduced by 40%, the inference speed is increased by 60%, and the performance is only reduced by about 3%. Compared to DistillBERT, TinyBERT proposes knowledge distillation for Transformer structure, and knowledge distillation for two stages of pre-training and fine-tuning. FastBERT uses the self-distillation method. Each layer of BERT connects to a classifier, automatically adjusts the calculation amount of each sample through the sample adaptation mechanism. The natural samples can be predicted quickly, and the harder samples need to go through the whole process. Improve the efficiency of the model. MiniLM uses deep self-attention knowledge distillation. Train the student model through attention score information and value relation information of the self-attention layer.

2.2. Downstream NLP tasks and data
General Language Understanding Evaluation (GLUE)[10] is one of the most crucial evaluation systems of the current measurement model in terms of language understanding. But for Chinese natural language processing, there is a lack of similar mature evaluation system, so many experts and scholars in the field
of Chinese NLP have launched a set of evaluation standard system called Chinese Language Understanding Evaluation (CLUE)[2].

NLPCC 2020 Shared Task 1 selected the NLP task in CLUE Benchmark and used the data set published by CLUE. These downstream tasks respectively cover four different aspects, including referential digestion (CLUEWSC2020), paper keyword recognition (CSL), named entity recognition (CLUENER2020)[11] and machine reading comprehension (CMRC2018)[12].

2.2.1. CLUEWSC2020. CLUEWSC2020 is an anaphora resolution task, which requires the model to determine whether the pronouns and nouns in the sentence are co-referential. The task is constructed from a similar data set in English.

Reference is a common linguistic phenomenon, widely existing in various expressions of natural language. In general, there are two kinds of referents: anaphora and co-referential. The anaphora refers to the connection between the current pronoun and the noun (phrase) appearing above. This kind of reference depends on the context and semantics. It may refer to different entities in different contexts. This kind of reference has asymmetric Sexual and non-transitive; co-referential mainly refers to the fact that 2 or more nouns (pronouns, noun phrases) point to the same entity in the real world. This kind of referral is still valid without context.

2.2.2. CSL. This task mainly uses Chinese abstracts and keywords of Chinese core journals. These papers cover multiple fields of natural and social sciences, so the data samples are more diverse. Mix the generated fake keywords with correct keywords, given a summary and some keywords, and determine whether these keywords are the original keywords of a paper. This task mainly evaluates the model's ability to judge whether keywords can summarize documents.

Conventional keyword extraction methods include keyword extraction based on statistics. This method uses word statistical information extraction in documents, such as part-of-speech, word frequency, inverse text frequency, word position, mutual information, and word span. The other is keyword extraction based on the graph model, represented by PageRank and TextRank. The last one is keyword extraction based on the topic model, such as LDA, which mainly uses the nature of topic distribution in the topic model for keyword extraction.

2.2.3. CLUENER2020. This data is based on the open-source text classification data set THUCTC of Tsinghua University, and selects some data for fine-grained named entity labeling, and subdivides the data into ten label categories. Statistical NER systems typically require a large amount of manually annotated training data. This dataset uses a semi-supervised approach to avoid partial annotation work. Refer to the idea of unsupervised classification method[13].

Named-entity recognition (NER) is a sub-task of information extraction. This task attempts to classify named entities into predefined categories. It is an essential underlying technology for a number of NLP applications. Many NLP tasks are carried out on the premise that named entities can be well classified.

2.2.4. CMRC2018. CMRC2018 is a Chinese machine reading comprehension data set based on span extraction. This data set is from Wikipedia. Over 20,000 questions manually tagged. In CMRC 2018, all sample data consists of context, questions and related answers. The answer is the text span in the context. The proposed Chinese RC data could also be a resource for cross-lingual research purpose when studied along with SQuAD and other similar datasets.

The purpose of Machine Reading Comprehension (MRC) is to understand the context and to answer questions based on that understanding. With the development of the pre-training model, MRC tasks range from a focus on qualifying text to incorporating external knowledge, from a focus on specific pieces to a full understanding of context. There is now an increasing focus on the use of background knowledge and a deep understanding of the text.
3. Knowledge distillation
Advanced large models have achieved very good performance. However, because these models are large and deep, with hundreds of millions of parameters, they are usually slow in the prediction stage. Therefore, when choosing a lightweight pre-trained model, we hope that the model is as small as possible, the prediction speed is as fast as possible, and the accuracy is guaranteed as much as possible. It is a good choice for those less complex tasks to use lightweight model to replace those large and slow BERT models.

We chose CLUE's open-source model RoBERTa-tiny-clue[1] as the light pre-training model. RoBERTa-tiny-clue is based on the RoBERTa model, through the CLUE corpus and vocabulary pre-training[14]. The parameter size of this model is 7.5M, the storage size is 28.3M, the model parameters meet the requirements of Shared Task 1. The configuration of hyper-parameters is keep same as ALBERT-tiny, with hidden size 312 for 4 layers. It is around ten times fast for training and prediction compare to BERT-base. Based on CLUE Benchmark, we use the RoBERTa-tiny-clue model to test the four downstream subtasks: CLUEWSC2020, CSL, CLUENER2020 and CMRC2018.

Table 1. Performance of BERT-base and RoBERTa-tiny-clue on downstream tasks.

|                | WSC (Val. Acc.) | CSL (Val. Acc.) | CLUENER (Val. F1) | CMRC2018 (Val. EM) |
|----------------|-----------------|-----------------|-------------------|-------------------|
| BERT-base      | 67.3            | 82.2            | 64.4              | 51.8              | 74.8              |
| RoBERTa-tiny-clue | 63.5           | 78.2            | 61.8              | 47.6              | 68.7              |

3.1. Soft label
Perform knowledge distillation for reading comprehension tasks is similar with the method of distillation learning described in TextBrewer[15].

For the teacher model, we use RoBERTa-large-wwm, do fine-tune on the target task CMRC2018 to train the teacher model. Distillation is trained on incremental labeled dataset CMRC2018 and DRCD. For student model, we use RoBERTa-tiny-clue. At distillation phrase, initialize the training and distillation configuration, build the distiller, then define the adapter and callback function, and call the distiller's training method.

![Figure 1. Operation process of soft label distillation learning.](image)

Through the method of mark soft label, the output distribution of the teacher model is transferred to the student model, so as to achieve the purpose of improving the effect of the student model on the target task, as shown in Table 2.
Table 2. Use soft label to improve the performance of small models.

| Model                  | Parameters (M) | Val. EM | Val. F1 |
|------------------------|----------------|---------|---------|
| RoBERTa-large-wwm      | 325            | 67.8    | 85.7    |
| RoBERTa-tiny-clue      | 7.5            | 47.6    | 68.7    |
| With soft label distilled | 7.5          | 58.8    | 78.1    |

3.2. Hard label
For CLUENER tasks, we adopt the method based on homologous data annotation to conduct distillation learning.

Training a large model with excellent performance on the CLUENER task and use it to tag task-homologous corpus. Here, MSRA NER and Toutiao News (TNEWS) datasets are used as corpus. Tagged data is used to train a lightweight model that meets the requirements of this task.

We use RoBERTa-large-wwm as teacher model and student model is distilled on MSRA corpus. The method we used in this part is different from the previous part. It does not directly label the corpus but learns the teacher model's output distribution, that is the difference between soft labeling and hard labeling.

Figure 2. Operation process of hard label distillation learning.

Table 3. Use hard label to improve the performance of small models.

| Model                  | Parameters (M) | Val. EM | Val. recall | Val. F1 |
|------------------------|----------------|---------|-------------|---------|
| RoBERTa-large-wwm      | 325            | 76.9    | 79.3        | 78.1    |
| RoBERTa-tiny-clue      | 7.5            | 61.4    | 62.1        | 61.8    |
| With hard label distilled | 7.5          | 65.4    | 68.4        | 66.9    |

4. Data augmentation
In the field of natural language processing, data augmentation is a very effective way to expand the sample size of data. The use of large scale and high-quality data in training models allows for better generalization of models.

Easy data augmentation (EDA)[16] includes synonym replacement, random insertion, random exchange, and random deletion. Synonym Replacement: Randomly select \( n \) words that do not belong to
the stopword set from the sentence, and randomly choose their synonyms to replace them. Random Insertion: Randomly find a word in the sentence that does not belong to the stop word set, and calculate a random synonym for the word, and then insert the synonym into a random position in the sentence. Repeat the above steps \( n \) times. Random Swap: Randomly select two words in the sentence and exchange their positions. Repeat the above steps \( n \) times. Random Deletion: Remove each word in the sentence randomly with probability \( p \). The selection of the above \( n \) and \( p \) values determines the difference between the enhanced sample and the original data, also determines the effect of data augmentation.

Back translation can also achieve data augmentation. We use machine translation to translate a Chinese text into another language, and then translate back to Chinese. This method not only has the function of synonym replacement, but also can add or remove words and reorganize sentences while maintaining the original intent.

Contextual augmentation is a context-based data augmentation method. Randomly replace words in a sentence with other words that are predicted by a bi-directional language model at the word positions. In this way, the words in the sentence are replaced with other words with a paradigmatic relationship.

4.1. Multi-task learning

The goal of multi-task learning (MLT) is to use the useful information contained in multiple learning tasks to help each task learn and get a more accurate learner. MTL helps alleviate data sparsity by using useful information from other related learning tasks, and achieves the effect of introducing external data sources to enhance the data.

For CMRC2018, We perform data preprocessing on WebQA[17] and Laisi Cup data (a machine reading comprehension task for the military field), arrange their format to be consistent with the format of CMRC2018, as enhanced data.

WebQA's data is collected from a large community QA website Baidu Zhidao. These questions are all put forward by users in daily life. They cover all kinds of problems in life and involve a lot of fields. Relatively speaking, the Laisi Cup data is focused on the military field. We mixed WebQA data and Laisi data at different ratios to test the augmentation effect.

WebQA is a large dataset with different content length, we divide it into subsets with maximum lengths of 32, 64, 125, 256, 512 respectively. There are 6 augmented datasets were built. We first compared the scores of training on each dataset and evaluate on CMRC validation set, thereby observing how relevant datasets are to the CMRC task. Then, we mixed CMRC training data and augmented data to tell if the augmented data is helpful for the CMRC task.

| Dataset     | Data Num. | Val. F1 | Val. EM |
|-------------|-----------|---------|---------|
| Laisi       | 23913     | 67.0    | 42.5    |
| WebQA-32    | 110923    | 17.4    | 7.7     |
| WebQA-64    | 114765    | 25.6    | 11.7    |
| WebQA-128   | 114089    | 27.4    | 13.1    |
| WebQA-256   | 67643     | 29.2    | 12.4    |
| WebQA-512   | 49605     | 19.6    | 7.7     |

Table 5. Performance of mixed augmented data and CMRC training data on validation set.

| Dataset          | Data Num. | Val. F1 | Val. EM |
|------------------|-----------|---------|---------|
| CMRC_Laisi       | 34055     | 78.1    | 55.9    |
| CMRC_WebQA-32    | 121064    | 75.7    | 54.6    |
| CMRC_WebQA-64    | 124902    | 75.7    | 53.9    |
It can be seen from table 4, performance of mixed training data are better than that of pure CMRC training data (F1:68.6/EM:47.6). The final augmented data is a mixture of all the above data.

In the fine-tuning phase, two different fine-tuning methods were tested. One method is to fine-tune with augmented data first, and then fine-tune with CMRC training data (One-stage fine tuning). The other is to fine-tune the two types of data together (Two-stage fine tuning). Two.

Table 6. Performance of fine-tuning method on CMRC tasks.

| Dataset            | Val. Avg. | Val. F1 | Val. EM |
|--------------------|-----------|---------|---------|
| Baseline           | 58.1      | 68.6    | 47.6    |
| Augmented CMRC One-stage | 69.2      | 79.5    | 58.8    |
| Augmented CMRC Two-stage | 70.0      | 80.8    | 59.2    |

4.2. Same original data augmentation
For CSL, we collected same original data as the CSL dataset which is the abstract part and key words of CNKI Chinese core papers. We used TF-IDF and TextRank to extract keywords and found a union to construct a set of added keywords. Different from select n correct keywords and n false keywords to constitute positive and negative samples, we enriched the types of negative samples in order to improve the discrimination of the data samples to the model. We select m false keywords and n-m correct keywords to constitute a new negative sample. We separately compared the TF-IDF and textrank methods, and their hybrid strategies to construct fake keywords.

Table 7. Performance of augmented CSL data.

| Dataset              | Val. precision | Val. recall | Val. F1 | Val. Acc. |
|----------------------|----------------|-------------|---------|-----------|
| Baseline             | 76.2           | 82.3        | 79.1    | 78.2      |
| TF-IDF (One-stage)   | 76.7           | 82.2        | 79.4    | 78.6      |
| TF-IDF (Two-stage)   | **77.5**       | **86.0**    | **81.5**| **80.5**  |
| TextRank (One-stage) | 74.2           | **86.3**    | 79.8    | 78.1      |
| TextRank (Two-stage) | 76.4           | 84.3        | 80.1    | 79.1      |
| Mixed (One-stage)    | 75.6           | 84.3        | 79.7    | 78.5      |
| Mixed (Two-stage)    | 75.8           | 86.2        | 80.7    | 79.3      |

4.3. BERT cloze data augmentation
BERT is a masked language model, which is inherently suitable for cloze tasks. We use the BERT model to replace the words in the sentence with words of similar meaning to achieve data augmentation. With an input sentence, we use RoBERTa-large-wwm model to mask and predict top N candidate words at each position. Then, randomly select M words to replace one of the candidate characters, in which the pronoun and the noun will not be replaced. We compare the experimental results of different choices of N and M.

Table 8. Performance of augmented WSC data.

| Dataset | Val. precision | Val. recall | Val. F1 | Val. Acc. |
|---------|----------------|-------------|---------|-----------|
| Baseline| 50.4           | 51.4        | 50.9    | 63.8      |
Table 9. Performance of final model on downstream tasks.

|                | WSC (Val. Acc.) | CSL (Val. Acc.) | CLUENER (Val. F1) | CMRC2018 (Val. EM) |
|----------------|----------------|----------------|-------------------|--------------------|
| BERT-base      | 67.3           | 82.2           | 64.4              | 51.8               |
| RoBERTa-tiny-clue | 63.5         | 78.2           | 61.8              | 47.6               |
| Final model    | 66.1           | 80.5           | 66.9              | 55.9               |

5. Conclusion

According to the fact that large models are slow to train in engineering applications and require high deployment resources. This paper tested the impact of different knowledge distillation methods on model performance, trained a more competitive model with RoBERTa-tiny-clue as the backbone model. We also performed a series of data augmentation methods for specific tasks. In our experiments, it was found that two-stage fine-tuning is more effective for small models. This experience can provide a reference for engineering applications.

Combining various optimization strategies, we achieved a high-performance small model based on open-source pre-trained model and public available datasets. It is 15 times smaller and 12 times faster than BERT-base on average, and 95% of BERT-base performance on downstream tasks. Using suitable data augmentation methods for the trained lightweight model, the performance of the model on various downstream tasks reaches or exceeds BERT-base.

References

[1] Xu, L., Zhang, X., & Dong, Q. (2020) CLUECorpus2020: A Large-scale Chinese Corpus for Pre-training Language Model. arXiv preprint arXiv:2003.01355.
[2] Xu, L., Zhang, X., Li, L., Hu, H., Cao, C., Liu, W., ... & Cui, Y. (2020) CLUE: A Chinese Language Understanding Evaluation Benchmark. arXiv preprint arXiv:2004.05986.
[3] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2019) ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. In International Conference on Learning Representations.
[4] Buciluă, C., Caruana, R., & Niculescu-Mizil, A. (2006) Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. New York. (pp. 535-541).
[5] Hinton, G., Vinyals, O., & Dean, J. (2015) Distilling the Knowledge in a Neural Network. stat, 1050, 9.
[6] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019) DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108.
[7] Jiao, X., Yin, Y., Shang, L., Jiang, X., Chen, X., Li, L., ... & Liu, Q. (2019) Tinybert: Distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351.
[8] Liu, W., Zhou, P., Zhao, Z., Wang, Z., Deng, H., & Ju, Q. (2020) FastBERT: a Self-distilling BERT with Adaptive Inference Time. arXiv preprint arXiv:2004.02178.
[9] Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., & Zhou, M. (2020) Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. arXiv preprint arXiv:2002.10957.
[10] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2018) GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP (pp. 353-355).
[11] Xu, L., Dong, Q., Yu, C., Tian, Y., Liu, W., Li, L., & Zhang, X. (2020) CLUENER2020: Fine-grained Name Entity Recognition for Chinese. arXiv preprint arXiv:2001.04351.
[12] Cui, Y., Liu, T., Che, W., Xiao, L., Chen, Z., Ma, W., ... & Hu, G. (2019) A Span-Extraction Dataset for Chinese Machine Reading Comprehension. 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). Hong Kong. (pp. 5886-5891).
[13] Zhang, D., Li, D., Guo, L., & Tan, K. L. (2020) Unsupervised Entity Resolution with Blocking and Graph Algorithms. IEEE Transactions on Knowledge and Data Engineering.
[14] Peng, B., Jin, X., Liu, J., Zhou, S., Wu, Y., & Liu, Y., et al. (2019) Correlation congruence for knowledge distillation.
[15] Yang, Z., Cui, Y., Chen, Z., Che, W., Liu, T., Wang, S., & Hu, G. (2020) TextBrewer: An Open-Source Knowledge Distillation Toolkit for Natural Language Processing. arXiv preprint arXiv:2002.12620.
[16] Wei, J., & Zou, K. (2019) EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Hong Kong. (pp. 6383-6389).
[17] Li, P., Li, W., He, Z., Wang, X., Cao, Y., Zhou, J., & Xu, W. (2016) Dataset and neural recurrent sequence labeling model for open-domain factoid question answering. arXiv preprint arXiv:1607.06275.