Encoded Classifier Using Knowledge Distillation for Multi-Author Writing Style Analysis

Notebook for PAN at CLEF 2023

Mingjie Huang, Zhaohao Huang, Leilei Kong*

Foshan University, Foshan, Guangdong, China

Abstract

In this paper, we report on our writing style change detection system which is used for the PAN task of Multi-Author Writing Style Analysis. To detect the writing style change within texts, a method based on an encoded classifier using knowledge distillation is proposed. The method proposed in this paper consists of two parts: A neural classifier based on an encoder of a pre-trained language model is used to extract the features of texts and make the writing style change detection. And the knowledge distillation method based on the teacher-student architecture is used for model compression. We evaluate our methods on three tasks with different difficulties on the metrics for F1 score.

Keywords

Multi-author writing style analysis, knowledge distillation, pre-trained language model

1. Introduction

The goal of the style change detection (SCD) task is to identify text positions within a given multi-author document at which the author switches [1]. The SCD task in PAN@CLEF 2023 is defined as for a given text, find all positions of writing style change on the paragraph level [1]. The simultaneous change of authorship and topic is carefully controlled and participants are provided with datasets of three difficulty levels [1]. Multi-level data sets can not only reflect the ability of the same model in different scenarios but also inspire participants to try to use diverse methods to solve SCD problems [1] .

In recent years, PAN@CLEF has held many international competitions on SCD tasks, and the participants also provided a variety of inspiring methods. There are three main categories of participant approaches. The first category is the method of artificially selecting traditional features in the text for similarity discrimination [3]. The second type uses the neural network model to extract the text representation and then calculates the similarity of the text representations [4]. The third category leverages a more complex pre-trained language model, and some participants build a Siamese model or multi-model fusion on this basis [5, 6]. According to the performance rankings of various methods in recent years, it can be observed that models with larger parameters and more complex structures tend to have better performance on SCD tasks.

In this paper, we propose a method of encoded classifier using knowledge distillation [7] for multi-author writing style analysis. The teacher model with huge parameters and complex structure will be trained on the task-specific dataset and in-domain dataset. To allow the uploaded model to run on the evaluation machine, the teacher model will be compressed through the technology of knowledge distillation, where the trained teacher model and student model will be finetuned on task-specific datasets.

[1] CLEF 2023: Conference and Labs of the Evaluation Forum, September 18–21, 2023, Thessaloniki, Greece

EMAIL: mingjiehuang007@163.com (A. 1); zhaohaohuang6@gmail.com (A. 2); kongleilei@fosu.edu.cn (A. 3)
ORCID: 0000-0002-0889-5027 (A. 1); 0000-0003-3278-8472 (A. 2); 0000-0002-4636-3507 (A. 3)

2. Method

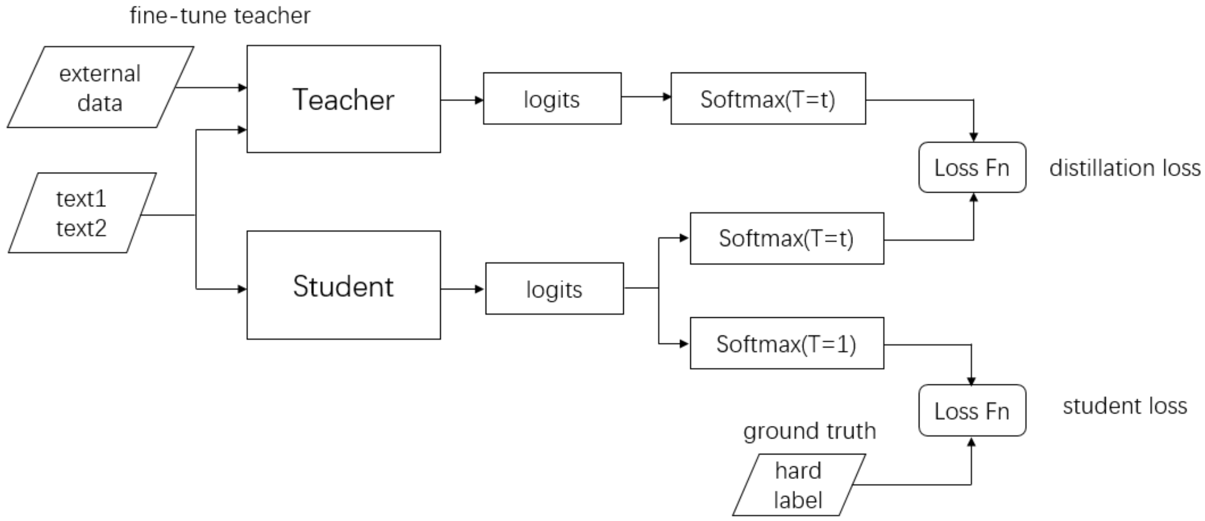

Figure 1: The architecture of the encoded classifier using knowledge distillation.

Our proposed method is shown in Figure 1, Firstly, the teacher model is trained on task-specific datasets and external data. Then the teacher model will be compressed by using knowledge distillation [7] to a smaller student model.

2.1. Data processing

Given a document D, it will be divided into fragments according to natural paragraphs, denoted as $\{p_1, p_2, \dots, p_n\}$. Then each fragment will be recombined with the following fragment to form a new text pair, forming a total of n-1 pairs, denoted as $\{(p_1, p_2), (p_2, p_3) \dots, (p_{n-1}, p_n)\}$. After preprocessing, we converted the multi-label classification problem into a writing style similarity discrimination problem between text pairs. For paragraph pairs that are longer than the maximum sequence size of our models, we first count the token sequence length distribution of paragraphs in the data set, and then select the length that can cover 90% of the sequence as the threshold length for text cutting, and discard the part that exceeds the threshold length.

2.2. Teacher model

The teacher model consists of an encoder based on the Transformer architecture and an MLP, where the encoder converts text into a high-dimensional text representation, and the MLP performs binary classification based on the text representation. The encoder part uses the encoder of the pre-trained language model, so it only needs to be fine-tuned on the downstream tasks.

We train the teacher model using conventional fine-tuning methods on pre-trained language models. In addition to training the teacher model with the dataset specific to style change detection, we also use an additional in-domain dataset as a supplement, expecting the teacher model to learn more relevant writing style information. But this external data set is only used for the training of the teacher model, not for that of the student model, to avoid interference with the student model.

Specifically, we unify the formats of different datasets into the form of text pairs and labels. During training, the dataset is randomly shuffled to ensure an even distribution of different data, and the parameters of both the encoder and MLP will be updated.

2.3. Knowledge distillation

2.3.1. Model training

The student model is a model of the same series as the teacher model while the student model has smaller parameters. The student model is not as powerful as the teacher model in reasoning and classification while the student model has the advantage of smaller computing resource requirements and faster inference speed than the teacher model.

Before knowledge distillation, the teacher model must first be fine-tuned on the task-specific data set again to ensure that the teacher model fits the data adequately. After that, the knowledge of the teacher model will be transferred to the student model through knowledge distillation.

2.3.2. Loss function

To allow our final trained model to run on limited computing resources, Hinton's classic knowledge distillation [7] method is used in our model. The loss function consists of two parts, the distillation loss between the teacher model and the student model and the loss between the student predictions and ground truth labels.

After the task-specific data is fed into the teacher model and the student model, the two models each output a probability distribution, also known as logits. When calculating the loss between the teacher model and the student model, the logits output by the two models will be divided by a temperature coefficient T, then normalized by the softmax function into a probability distribution and finally the KL divergence loss between the two probability distributions can be calculated, which is called distillation loss [7], denoted as L_{kd}.

$$L_{kd} = KL(\frac{\exp(z_i^s)}{\sum_j \exp(z_j^s)}, \frac{\exp(z_i^t)}{\sum_j \exp(z_j^t)}) \quad (1)$$

where z^s is the logits of student model, z^t is the logits of teacher model, $KL()$ is KL divergence function.

In addition to the distillation loss, another part of the total loss is the cross-entropy loss between the probability predicted by the student model and the ground truth label, denoted as L_{ce}. Combining the two losses, with different weights, we get the final total loss function.

$$L_{total} = \alpha L_{kd} + (1 - \alpha) L_{ce} \quad (2)$$

where α is the weight of ground truth loss, and L_{total} is the total loss of the whole knowledge distillation model.

3. Data

The data on writing style change detection provided by PAN@CLEF consists of three difficulty levels.

1. **Easy**: The paragraphs of a document cover a variety of topics, allowing approaches to make use of topic information to detect authorship changes.
2. **Medium**: The topical variety in a document is small (though still present) forcing the approaches to focus more on style to effectively solve the detection task.
3. **Hard**: All paragraphs in a document are on the same topic.

In the data set given by PAN, the label information available to the participants includes the number of authors in the document and the labels of whether the style of writing changes between paragraphs. We split the documents in the dataset by natural segments and labels and re-counted the number of datasets. The statistical results are shown in the table below.

Table 1
Dataset size of three tasks.

task	easy		medium		hard	
	train	validation	train	validation	train	validation
num of documents	4,200	900	4,200	900	4,200	900
num of text pairs	12,904	2,828	28,216	7,042	19,113	4,112

We choose the training set of the PAN 2020 authorship verification task, a total of 52601 pieces of data, as external data for training. The authorship verification dataset is composed of text pairs and labels used to judge whether they are written by the same author.

4. Experiments

4.1. Experiments setup

In this paper, we choose mT0-xl [8] as the teacher model's encoder with 24-layer, 2048-hidden, 24-heads, and 1.8B parameters. And we choose mT0-large [8] as the student model's encoder with 24-layer, 1024-hidden, 24-heads, and 0.6B parameters. The vocab size is 250,112.

When we finetune the teacher model on all three datasets and external datasets, the training batch size is set to 16, and the maximum length of the encoder is set to 256, which means the total length of text pairs is 512, since most of the text pairs have less than 512 tokens. We use Adafactor optimizer, learning rate set to 1e-6, training epochs set to 10, and dropout layer, the rate set to 0.1, to avoid overfitting during fine-tuning.

When distilling the model, the temperature coefficient T is set to 4, and weight α is set to 0.7. Other hyperparameters are set as shown in Table 2.

Table 2
Hyperparameters of model training.

procedure	learning rate	dropout rate	batch size	epochs
teacher finetuning	1e-6	0.1	16	20
distillation	3e-4	0.1	16	20

We train and distill the model on an A800 80GB GPU, and test it on a virtual machine with 4 CPUs and 40GB of memory. The deep learning framework we use is Pytorch.

4.2. Results

The models are tested on TIRA [9] and evaluated on F1-score for three tasks respectively. The results on the three tasks of validation dataset and test dataset respectively are shown in the table below.

Table 3
The final scores of our model on the three tasks of validation dataset and test dataset respectively

Dataset	task1	task2	task3
validation dataset	0.9691	0.8003	0.7867
test dataset	0.9678	0.8057	0.7900

From the results in the table 3, we can infer that our methods have a certain degree of generalization, since the model that performs well on the validation dataset can also have similar performance on the test dataset.

4.3. Ablation experiments

To demonstrate the validity of a larger model, we trained a Bert-base model with 110M parameters as our baseline. Table 3 shows the performance of a BERT-base [10] model distilled from a BERT-large model and an mT0-large model distilled from an mT0-xl model. For this part, we separate 20% of data from training data to monitor the training progress and test these two models on validation data.

Table 4
F1 score comparison on validation data.

model	task1	task2	task3
BERT-base distilled	0.9645	0.7957	**0.7805**
mT0-large distilled	**0.9649**	**0.8029**	0.7696

The results prove that the model with more parameters can indeed perform better than the small model, although the gap is not large, it may also be because the large model takes longer to fit, and in the experiment, we only fine-tuned 20 epochs.

Then we put 20% of the training data back into the training set to train the model and re-compare the performance. The results are shown in Table 5.

Table 5
F1 score comparison for mT0-large distilled models with an 80% training set, with the complete training set or with the both complete train data and external data.

model	task1	task2	task3
80% train data	0.9649	**0.8029**	0.7696
all train data	0.9650	0.7939	0.7675
with external data	**0.9691**	0.8003	**0.7867**

From the results in the table, we can observe that when a small amount of training data is added, the performance of the model does not necessarily improve. But when we add a large amount of training data to the model, the model can gain some knowledge from it to improve performance on certain tasks. So we can guess that enhancing the data or introducing more in-domain data can add more positive effects to the training of the model.

In addition, since the mT0-xl model is relatively large compared to popular pre-trained models such as BERT-base, which only has 110M parameters, we tried many hyperparameters before selecting the final parameter settings. We mainly test hyperparameters on the data set of task 1. The performance corresponding to hyperparameters is shown in the following table.

Table 6
F1 score of student model with different hyperparameter sets for both teacher model and student model.

batch size	dropout rate	epochs	teacher lr	task1 F1
8	0.1	10	2e-4	0.9592
16	0.2	10	2e-4	0.9641
16	0.1	10	3e-4	0.9688
16	**0.1**	**20**	**1e-6**	**0.9691**

For the mT0-xl with great amounts of parameters, choosing a conventional learning rate of 3e-4 may make it difficult to find a better global optimal point, and choosing a smaller learning rate to fit more for more epochs can expect to get better performance.

5. Conclusions

In this paper, we proposed a method based on an encoded classifier using knowledge distillation for writing style change detection. We first train a large model as the teacher model on datasets of all three tasks and external datasets, then distill the teacher model into a smaller student model. With this approach, we obtain a student model that can run with fewer resources and possesses capabilities close to that of the teacher model. In the follow-up wortion tunning skills to leverage the emergence ability of LLM for SCD tasks.

6. Acknowledgments

This work is supported by the Natural Science Foundation of Guangdong Province, China (No.2022A1515011544)

7. References

[1] J. Bevendorff, I. Borrego-Obrador, M. Chinea-Rios, et al., Overview of PAN 2023: Authorship Verification, Multi-Author Writing Style Analysis, Profiling Cryptocurrency Influencers, and Trigger Detection, in: A. Arampatzis, E. Kanoulas, T. Tsikrika (Eds.), CLEF 2023 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2023.

[2] E. Zangerle, M. Mayerl, M. Potthast, B. Stein, Overview of the Multi-Author Writing Style Analysis Task at PAN 2023, in: M. Aliannejadi, G. Faggioli, N. Ferro, M. Vlachos, CLEF 2023 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2023.

[3] Castro-Castro, D., Rodríguez-Lozada, C.A., noz, R.M.: Mixed Style Feature Representation and B-maximal Clustering for Style Change Detection. In: Cappellato, L., Ferro, N., Névéol, A., Eickhoff, C. (eds.) CLEF 2020 Labs and Workshops, Notebook Papers, CEUR-WS.org (Sep 2020)

[4] R. Deibel, D. Löfflad, Style Change Detection on Real-World Data using LSTM-powered Attribution Algorithm, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2021.

[5] Z. Zhang, X. Miao, Z. Peng, J. Zeng, H. Cao, J. Zhang, Z. Xiao, X. Peng, Z. Chen, Using Single BERT For Three Tasks Of Style Change Detection, in: G. Faggioli, N. Ferro, A. Joly, M. Maistro, F. Piroi (Eds.), CLEF 2021 Labs andWorkshops, Notebook Papers, CEUR-WS.org, 2021.

[6] X. Jiang, H. Qi, Z. Zhang, Style Change Detection: Method Based On Pre-trained Model And Similarity Recognition, in: CLEF 2022 Labs and Workshops, Notebook Papers, CEUR-WS.org, 2022.

[7] G. Hinton, O. Vinyals and J. Dean, 2015. Distilling the knowledge in a neural network. Machine Learning. arXiv:1503.02531. 2015.

[8] N. Muennighoff, T. Wang, L, Sutawika, A, Roberts, S. Biderman and et al., Crosslingual Generalization through Multitask Finetuning. arXiv: 2211.01786. 2022

[9] M. Frobe, M. Wiegmann, N. Kolyada, et al., Continuous Integration for Reproducible Shared Tasks with TIRA.io, in: J. Kamps, L. Goeuriot, F. Crestani (Eds.), Advances in Information Retrieval. 45th European Conference on {IR} Research. Lecture Notes in Computer Science, 2023.

[10] J. Devlin, M.W. Chang, K. Lee, et al. Bert: Pre-training of deep bidirectional transformers for language understanding//Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, 1: 4171-4186