Author Identification Base on Media Contents

Liming Zhang, Zhenyu Xuan, Rui Bao, Xiaoting Chen, Yingying Fang, YunFeng Zeng

Abstract—With the development of automatic writing technology, nowadays many materials in social media are automatically generated by machine learning models, which make readers indistinguishable between high or low quality of articles. Therefore, it is important to make effective identification of whether articles are produced by human or robot. In this study, we explore different deep neural networks for author identification. We proposed different models combined with character representations and word representations and ensemble them with a voting mechanism. Experiments show the effectiveness of our proposed method on the SMP EUPT 2018 dataset and we achieve the state-of-the-art performance.

Keywords—Author Identification, Social Media, User Profiles

1 Introduction

ARTIFICIAL intelligence technologies in automatic writing enable articles to be generated via a certain of algorithms. Nowadays, social media is flooded with plenty of articles every day, and some of them are made by robot. These articles without manual intervention may disturb readers by introducing wrong information or touchy subjects. Therefore, identification of authors (human or robot) is of great importance to publishers and reviewers.

Modeling articles for author identification can be viewed as text classification, but it could not be as simple as news categorization. For example, a news article mentioning "Basketball" and NBA" may be categorized into the Sport news with high probability. However, there are few keywords or phrases that can distinguish articles from human or robot. Therefore, common probabilistic topic modelling approaches may fail. Instead, compositions, hierarchies, structures of texts are needed to gained more attention. For example, in those articles created by automatic summary algorithm, the sequence order of words is often not as fluent as human writing. Some phrase expressed by machine translations may be rough and even incorrect. Herein, deep learning network models show their advantages, such as CNN, RNN. CNN captures n-gram features which can distinguish wrong expressions made by robot, while RNN detects the sequence order of word to make a distinction between human and robot. Additionally, from our observation, there are many subtle differences between human writing and robot writing, not only in word level, but also character level. Hence, we employ deep learning models both in word embeddings and character embedding, joining them together to get the whole article representation.

We totally create 12 different deep neural network architectures, combining word-level and character-level representations. Besides, we devise extra sophisticated features helping to analyze differences between human articles and robot articles. By unifying all these models and features with a lightGBM classifier, we rank the first in validation dataset. Finally, we only pick out 9 of pre-trained models and ensemble them with a unweighted voting mechanism, to get the highest score in the final test dataset. Codes for contest are publicly released at https://github.com/Quincy1994/smp2018.

2 Models

In this section, we firstly describe the differences between human writing and robot writing. Then, we introduce our proposed deep neural network models, which are varieties of CNN and RNN, as well as their combinations. In addition, we elaborate extra salient features that are helpful for classifying articles.

2.1 Word embeddings vs. Char embeddings

Both word embeddings layer and char embeddings layers are usually considered as the basic input of deep learning models. Compared to char embeddings, word embeddings effectively retain the sematic information of words, but they are greatly influenced by word segment methods. In contrast, char embeddings alleviate the deficiency of word segment, and show the effectiveness of solving the OOV problems, especially in short text. Take one sentence as an example, which is made by automatic summary algorithm:

"恭城 各乡、横山 新村、杜 山新村、北 洞 源 新村"

This sentence is processed by Hanlp segment tool, from which we observe a wrong word segment in "杜 山新村". But in fact, "新村" should be an independent unit. Based on this, together with a series of conducted experiment, we find that word embeddings perform inferior to char embeddings when articles only consist of less than 100

- *Liming Zhang is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China, CO 510006. Email: zhangliming134@foxmail.com.*
- *Zhenyu Xuan is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China, CO 510006. Email: 771640993@qq.com*
- *Rui Bao is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China, CO 510006. Email: zengyunfeng233@outlook.com.*
- *Xiaoting Chen is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China, CO 510006. Email: 954872212@qq.com.*
- *Yingying Fang is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China, CO 510006. Email: 1030768676@qq.com.*
- *YunFeng Zeng is with the School of Information Science and Technology, Guangdong University of Foreign Studies, Guangzhou, China, CO 510006. Email: 13500129650@163.com.*

words. This explains why it is hardest to categorize articles between human writing and automatic summary.

To address this, we unify word embeddings and char embeddings. Concretely, we first adopt them separately in a same neural network architecture to get two different forms of presentations in document level, concatenating them as the input of the final fully-connected layer.

2.2 CNN Network Architecture

We implement two kinds of CNN network architectures. A simple implement of CNN is inspired by [1], involving a convolutional layer and 1-max pooling layer, shown in Figure 1.

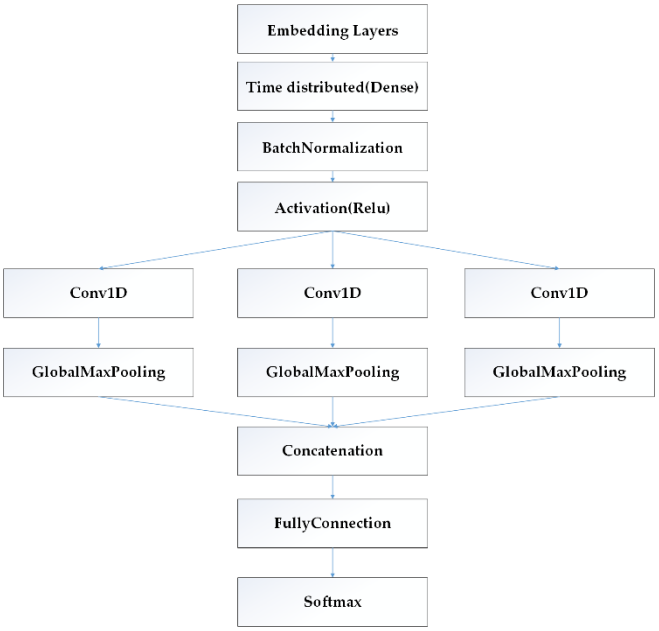
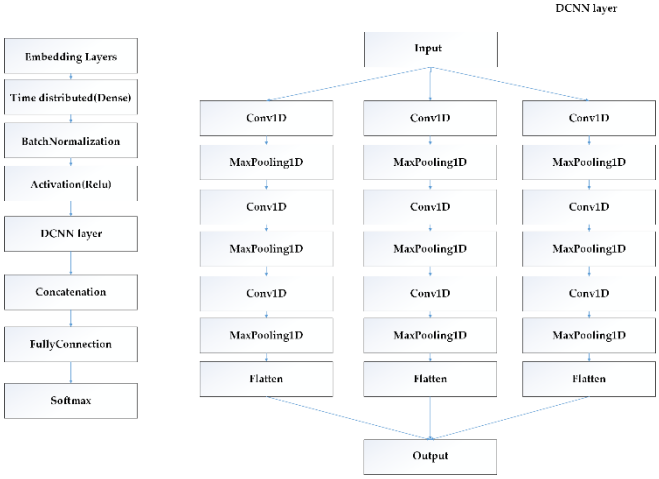


Fig. 1.an implement of CNN architecture

Another network becomes deeper, called DCNN, which utilizes 3 convolutional layers inspired by [2], shown in Figure 2. Instead of using a global max-pooling operation, we use max-pooling operation and send them into another



convolutional layer to get higher semantic information. After 3 layers of convolutional and max-pooling operation, we make flatten operation to get 1-dimension features and concatenate them together for modeling articles.

Fig. 2. an implement of DCNN architecture

2.3 RNN Network Architecture

RNN plays a prominent part in capture the sequence order of words or chars, reflecting the logic of semantic information. Herein, a bidirectional GRU layer is adopted and we use a global max-pooling and a global average-pooling operation to get document representations, shown in Figure 3.

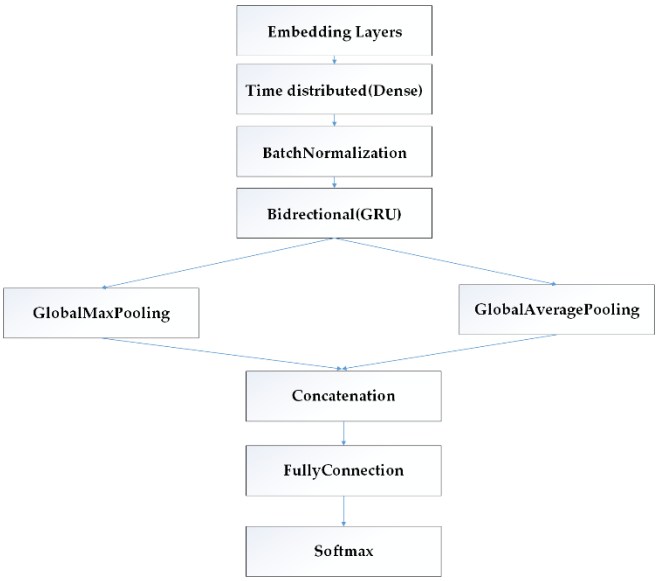


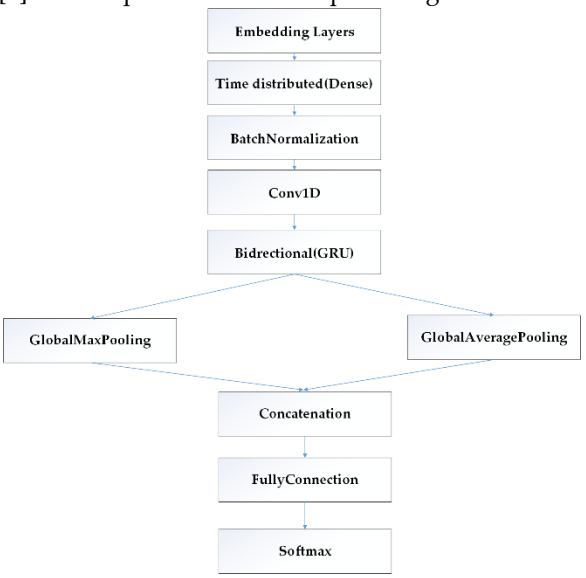
Fig. 3.an implement of RNN architecture

We do not use an attention mechanism out of reason that the information of sequence order of words is more important than semantic information of words for detecting author types.

2.4 Combinations of RNN and CNN Network

Both the architectures of RNN and CNN units are powerful tools for text classification, and we join them together to generate different forms of combinations. There are two popular network structures: CNN-RNN, RNN-CNN.

A combination form of CNN-RNN we devise comes from [3]. We adopt CNN units to capture N-grams features



and feed them into RNN units to get final document representation, as shown in Figure 4. We name it as CGRU.

Fig. 4.an implement of CGRU architecture

As for RNN-CNN architecture, we follow the model mentioned in [4], which employs RNN units to capture the sequential information, taking them as the input of the CNN layer, shown in Figure 5. We name it as RCNN.

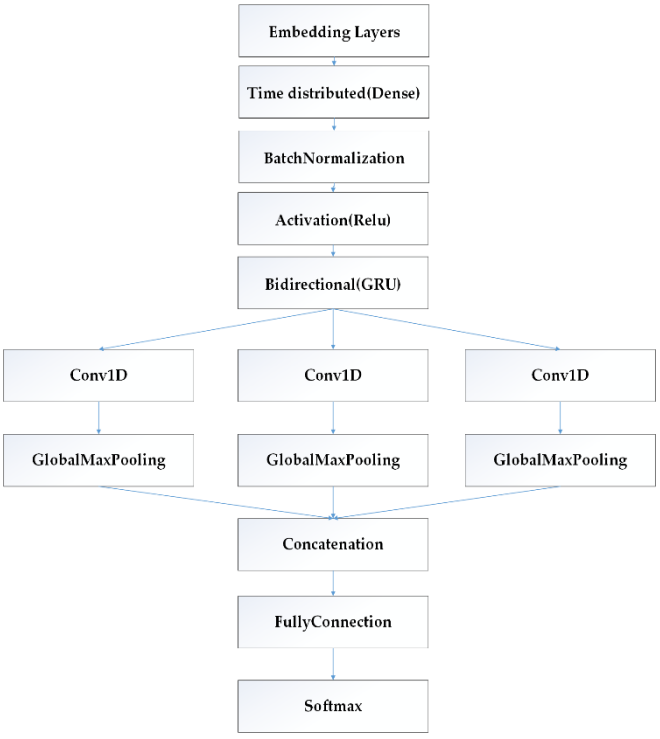


Fig. 5.an implement of RCNN architecture

2.5 Extra Features

Apart from deep learning models, we elaborate 4 types of extra features, mainly focusing on the following three parts: language perplexity, topic perplexity, statistic of sentences, use of punctuation.

Language perplexity considers the perplexity of N-gram language model. We count language perplexity for each sentence by using SRILM[1]. Among sentences, we calculate the highest scores, the average of top 3 highest scores, the lowest score, the range and variance of language perplexity.

Topic perplexity considers the perplexity of topic distribution in article. We count topic perplexity for the whole article by using Familia[2]. We calculate the number of topics with a frequency higher then 0.05, information entropy to topics.

Statistic of sentences include the number of sentences and the average length of sentences.

Use of punctuation detects those wrong punctuations appeared in articles, such as "[。。]", "[,。]", "[。,]" and ">>>>".

3 Ensemble and Results

To improve the accuracy of author identification, we totally construct 12 different neural network models based on Section 2.1 – 2.4. These models are combined with word embeddings and char embeddings. Details are shown in Table 1.

TABLE 1
MODERS OF DEEP NEURAL NETWORK

Model	Model
word_char_cnn	word_rcnn_char_rnn
word_char_rnn	word_rnn_char_rcnn
word_char_rcnn	word_cgru_char_rcnn
word_char_cgru	word_rcnn_char_cgru
deep_word_char_cnn	word_cgru_char_rnn
word_rnn_char_cnn	word_rnn_char_cgru

We use each model to get the probabilities of author types (*automatic summary, machine translation, robot writer, human writer*) respectively and join these probabilities together to form new presentations added with extra features. We adopt a LightGBM classifier to make the final prediction. This method shows the state-of-the-art performance in validation dataset. However, owing to the limitation of time, we only pick out 9 of deep neural network models for the final test dataset, using an unweighted voting mechanism without extra features which is trained in LightGBM. Table 2 shows the performance of different models, Table 3 shows the results of our methods on validation dataset and final dataset, with the metric of average F1 on four type of author identities.

TABLE 2
PERFORMANCE OF DIFFERENT MODELS

model	Off-line trainset	Validation dataset	Whether is used in test dataset
word_char_cnn	0.9888	0.9849	Y
word_char_rnn	0.9894	0.9863	Y
deep_word_char_cnn	0.9887	0.9828	Y
word_rcnn_char_rnn	0.9899	0.9879	Y
word_rnn_char_rcnn	0.9902	0.9872	Y
word_char_cgru	0.9896	0.9861	N
word_cgru_char_rcnn	0.9904	untested	Y
word_rcnn_char_cgru	0.9910	0.9882	Y
word_cgru_char_rnn	0.9887	untested	N
word_rnn_char_cgru	0.9899	untested	N
word_rnn_char_cnn	0.9897	0.9862	Y

[1] http://www.speech.sri.com/projects/srilm

[2] https://github.com/baidu/Familia

word_char_rcnn	0.9894	0.9884	Y

TABLE 3
RESULTS OF OUR PROPOSED METHODS

Model	scores
Validation dataset (lightgmb + 12 models + extra features)	0.99053
Test dataset (unweighted voting + 9 models)	0.98933

ACKNOWLEDGMENT

We would like to show our deepest gratitude to my supervisor, Professor Shengyi Jiang, who has provided us with great support and valuable guidance during the contest. We also extend our thanks to anonymous member in Data Mining Lab in GDUFS for their valuable assistance and suggestions.

Reference:

[1] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *Eprint Arxiv*, 2014.

[2] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," *arXiv1404.2188 [cs]*, 2014.

[3] C. Zhou, C. Sun, Z. Liu, and F. C. M. Lau, "A C-LSTM Neural Network for Text Classification," 2015.

[4] C. Wang, F. Jiang, and H. Yang, "A Hybrid Framework for Text Modeling with Convolutional RNN," *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '17*, pp. 2061–2069, 2017.