

INJECTING EVENT KNOWLEDGE INTO PRE-TRAINED LANGUAGE MODELS FOR EVENT EXTRACTION

Zining Yang¹, Siyu Zhan¹, Mengshu Hou¹,
Xiaoyang Zeng¹ and Hao Zhu²

¹School of Computer Science and Engineering, University of Electronic
Science & Technology of China, Chengdu, China

²Information Center, University of Electronic
Science & Technology of China, China

ABSTRACT

The recent pre-trained language model has made great success in many NLP tasks. In this paper, we propose an event extraction system based on the novel pre-trained language model BERT to extract both event trigger and argument. As a deep-learning-based method, the size of the training dataset has a crucial impact on performance. To address the lacking training data problem for event extraction, we further train the pre-trained language model with a carefully constructed in-domain corpus to inject event knowledge to our event extraction system with minimal efforts. Empirical evaluation on the ACE2005 dataset shows that injecting event knowledge can significantly improve the performance of event extraction.

KEYWORDS

Natural Language Processing, Event Extraction, BERT, Lacking Training Data Problem

1. INTRODUCTION

One Common task of Information Extraction (IE) is event extraction (EE) which aims to detect whether the text has mentioned some real-world events and if so, classifying event types and identifying event arguments. An example sentence and its event annotation in the ACE2005 [1] dataset has been provided in Figure 1. With the increasing amount of text data, EE is becoming an increasingly important component in many natural language processing (NLP) applications for decision making, risk analysis, and system monitoring.

Deep learning has been proven efficient and obtains the state-of-the-art result for event extraction task. As a kind of supervised learning approach, its performance is highly dependent on the quality and quantity of the training data. Generally, to achieve better performance, a neural network involves more parameters and therefore needs more data to converge without over fitting. However, labeling training data is not only time-consuming and laborious but also requires professional domain knowledge, which limits the size of the available corpus. For example, the ACE2005 corpus only has a total of 599 documents which is a very small quantity for the task to extract 33 predefined events and their arguments with 36 predefined roles.

The common idea of current solutions is data expansion technology, which generates more labeled training data from external corpus and uses both original and generated data for model training. We argue that the data generating method is hard for event extraction because events typically have a complex structure: an event can be mentioned by different triggers, different events have different arguments with different roles. To avoid this problem, instead of generating training data explicitly, we directly use the unlabeled corpus to inject event knowledge into our event extraction system by the novel pre-trained language model, which can be regarded as implicitly expand training data.

Concretely, we first build an event extraction system based on the pre-trained language model to extract both event trigger and event argument as our baseline. And then build an unlabeled event training dataset from a large corpus which is then being used to further train the language model to inject the event knowledge to the event extraction system. Compared to the baseline, our method achieves approximately 2% improvement for both trigger and argument classification.

The paper is organized as follows. Section 2 presents related works, along with a special focus on pre-trained language model based on which we build our event extraction system with the help of external event corpus in section 3. The event corpus construction details and evaluation settings are introduced in section 4. Section 5 concludes the paper.

Sentence: <u>Leung</u> was hired by the <u>FBI</u> and paid almost <u>\$2 million</u> over <u>20 years</u> to spy on the <u>Chinese</u> .			
EVENT 0:		EVENT 1:	
Event Type	Personnel: Start-Position	Event Type	Transaction: Transfer-Money
Trigger	hired	Trigger	paid
Arguments	Person: Leung Entity: FBI	Arguments	Giver: FBI Money: \$2 million Recipient: Leung Time: 20 years

Figure 1. An example sentence of ACE2005 dataset, there are two event mentions: Start-Position event triggered by hired and Transfer-Money event triggered by paid. Each event has some entities (underlined words or phrases) as its arguments with specific role.

2. RELATED WORK

2.1. Event Extraction

A variety of methods have been used for event extraction task. The pattern matching technique manually constructing event patterns with the help of professional knowledge. [2] and [3] are very early and typical pattern-based extraction system. Traditional feature-based machine learning algorithms are also widely used for event extraction task. These approach first extract feature from training text to train classifiers, then applying the classifiers for new text. [4] formulate the event extraction as a structured learning problem, and proposed a joint extraction algorithm integrating local and global features into a structured perceptron model to predict triggers and arguments simultaneously. [5] proposed a cross-entity event extraction model that exploited utilize global information as global features together with sentence-level features to train classifier. Recently, neural based deep learning method is becoming mainstream for event extraction. Deep learning can help to reduce the difficulties of feature engineering. Benefit from the well-designed network structure and the depth of network layers, it can typically achieve better performance than traditional machine learning algorithms. DMCNN [6] utilize a variant of convolution neural network called dynamic multi-pooling CNN to extract features and event

automatically. JRNN [7] adopts bidirectional recurrent neural network (RNN) to jointly extract event trigger and arguments. JMEE [8] propose an event extraction framework that extract features using bidirectional long short-term memory (LSTM) networks, and capture the global relationship by graph convolutional network (GCN) with attention mechanism.

A large and growing body of literature has investigated how to improve the extraction accuracy from a small set of labeled dataset. Utilize the bootstrapping [9] and active learning strategy [10] is challenging for event extraction as it is hard to evaluate the classification confidence for the generated event structure. Some methods expand data from knowledge bases (KBs, such as FrameNet [11][12][13], WordNet [14]) based on a set of hypotheses which is complicated and hard to cover the many different types of events.

2.2. Pretrained Language Model

Pre-trained language models have made great success in recent years and been a standard part of many NLP tasks. It adopts a two stages strategy: pre-trained on the massive unlabeled corpus to learn general contextualized representations with linguistic information of language and then fine-tune on a specific downstream task. For downstream tasks, pre-trained language model can be regarded as an encoder that encodes each token of the original text into a vector with contextual and semantic information which has been proved to be very effective and helpful to the downstream task. The Generative Pre-trained Transformer (GPT) [15] by OpenAI builds a unidirectional language model (LM) based on the transformer and firstly introduces the fine-tuning approach. Bidirectional Encoder Representations from Transformers (BERT) [16] overcome the unidirectionality constraint through a new training object called mask language model (MLM) and introduce the next sentence prediction (NSP) training object to obtain sentence representation.

The BERT language model is pretrained using the general English corpus, while the downstream tasks usually require some task-specific knowledge. However, very little research has been done to solve this domain mismatch problem. BioBERT [17] and SciBERT [18] shows pre-training with in-domain data are very efficient for biomedical and science domain tasks. [19] uses product knowledge to further training BERT for Review Reading Compression (RRC) task. [19] and [20] use in-domain data to improve the performance of Aspect-Target Sentiment Classification (ATSC) task. In [21], physiology, government and psychology knowledge are used to further train BERT to improve the Short Answer Grading task. Inspired by the aforementioned work, we leverage in domain event knowledge to improve the event extraction performance.

3. METHODOLOGY

This section describes how we build the event extraction system and inject event knowledge based on the BERT pretrained language model.

We extract event trigger and argument in a pipelines mode though two BERT fine-tune strategy respectively: token classification and sentence pair classification.

3.1. Event Trigger Extraction through Token Classification

Given a sentence and a set of predefined event types, trigger extraction aims to find the phrase in the sentence that most clearly express an event occurrence, and identify the event subtypes. This can be seen as a simple sequence labeling task. We encode the input by BERT as a single sentence and feed the contextual representation (BERT's last hidden layer) of each token to a

classifier to assign an event type. Besides 33 event subtypes defined by ACE2005, we use an extra “None” label to denote that a token does not trigger any event so that we can identify and classify triggers at the same time. We adopt the IO tagging because a trigger may across more than one token and two triggers hardly appear in adjacent positions.

3.2. Argument Extraction Through Sentence Pair Classification

Argument extraction is relatively more complicated. Following [4] and [8], we directly use the gold annotations for entities. In a sentence consist of words $\{w_1, w_2, \dots, w_n\}$, some of the words are trigger words $T: \{w_{t1}, w_{te}, \dots, w_{tk}\}$ with corresponding event type and some of the words are entity mention $E: \{w_{e1}, w_{e2}, \dots, w_{ej}\}$ as argument candidates, argument extraction aims to identify if the candidate entity is an argument of event triggered by the trigger words, and if so, recognize its role.

[22] explores constructing an auxiliary sentence as extra BERT input for Aspect-Based Sentiment Analysis (ABSA) task: predict sentiment polarity of each target’s aspects in a sentence which is similar to our argument extraction task. Their experiment demonstrates that converting a single sentence classification task to several sentence pair classification tasks can significantly improve the performance for the ABSA task. They discuss that their method can be seen as exponentially expanding the corpus. Inspired by their work, we also adopt this method to our system for argument extraction.

We treat the argument extraction task for a sentence as several multiclass classification problems: given a sentence s , events triggered by T and candidates entities E , predict the role over the full set of trigger-entity pairs. Table 1 shows the examples used to extract arguments for the example sentence in Figure 1. There are 37 roles in total. ACE2005 defines 36 different argument roles (e.g. place, person). We use an extra ‘None’ label to indicate that the entity is not the argument of a given event so that we can identify and classify arguments simultaneously). For each trigger-entity pair, we first build a simple auxiliary pseudo-sentence. For example, the generated sentence for the trigger-entity pair (paid, FBI) is “paid - FBI”. We use the sentence pair (the original English sentence and the generated auxiliary sentence) as BERT input. Follow the BERT convention, one special classification token “[CLS]” is added as the first token, and two “[SEP]” tokens are inserted between two sentences and appended to the end respectively. The final BERT input tokens s for this example is “[CLS] Leung was hired by the FBI and paid almost \$2 million over 20 years to spy on the Chinese. [SEP] paid - FBI [SEP]”. We use BERT to encode the constructed input sentence and get the last hidden layer $h \in \mathbb{R}^{L \times H}$ (H is the hidden size of BERT and L is the sequence length) as the contextual embedding:

$$h = \mathbf{BERT}(s) \quad (1)$$

We use the “[CLS]” token’s embedding in last hidden layer (denoted as $h_{[CLS]} \in \mathbb{R}^H$) to predict the argument role. The predicted argument role distribution is defined as:

$$z = \mathbf{softmax}(W_e h_{[CLS]} + b_e) \quad (2)$$

Where $W_e \in \mathbb{R}^K \times H$, $b_e \in \mathbb{R}^K$ are weights and bias for event type e . As different event type has a different set of arguments, we use separate argument classifiers for each event type so that the argument classifier can utilize the event type information.

For each sentence, the argument classification error is defined as the average of all the cross-entropy between the gold and our predicted arguments role distribution:

$$\mathcal{L}_{arg} = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K z_{n,k} \log(\hat{z}_{n,k}) \quad (3)$$

N is the total number of the trigger-entity pairs in the sentence. K is the total number of argument roles. $z_{n,k} \in \{0,1\}$ denote the gold role for the entity of the event, $\hat{z}_{n,k}$ is our model output.

Table 1. multiclass classification problems for arguments extraction

Trigger	Event Type	Entity	Role (Label)
hired	Start-Position	Leung	Person
hired	Start-Position	FBI	Entity
hired	Start-Position	\$2 million	None
hired	Start-Position	20 years	None
hired	Start-Position	Chinese	None
paid	Transfer-Money	Leung	Recipient
paid	Transfer-Money	FBI	Giver
paid	Transfer-Money	\$2 million	Money
paid	Transfer-Money	20 years	Time
paid	Transfer-Money	Chinese	None

3.3. Inject Event Knowledge by Further Pretrain BERT

To inject event knowledge to the BERT model, starting from the original BERT checkpoint which is trained on general English corpus (BooksCorpus and Wikipedia), we further pre-train it by in-domain corpus as an intermediate step before fine-tuning it for our event extract system described in 3.1 and 3.2.

Two training objects are used to further pretrain the BERT model: Mask Language Model (MLM) and Next Sentence Prediction (NSP).

For MLM task, 15% random tokens in the original sentence is masked (80% of which is replaced by special token “[mask]”, another 10% of which is replaced by a random token and the remind 10% is unchanged). The model is trained to predict masked tokens.

For NSP task, given a sentence pair (A, B), the model is trained to determine whether they are adjacent (sentence B is the actual next sentence that follows sentence A).

4. EXPERIMENT

4.1. Data Set and metric

We utilize the ACE2005 dataset to evaluate our event extraction system. Following previous data split convention [4][5], we use 40 newswire documents as testset, 30 randomly documents as development set, and remaining 529 documents as training set. We also adopt the following criteria to evaluate the extraction performance as previous work [4][6][7][8][12]:

A trigger is correct if its event subtype and offsets match those of a reference trigger.

An argument is correctly identified if its event subtype and offsets match those of any of the reference argument mentions.

An argument is correctly identified and classified if its event subtype, offsets, and argument role match those of any of the reference argument mentions.

We report individual micro precision, recall and f1 score on the test set for trigger/arguments identification/classification. The precision (Equation 4) is the ratio between correct predictions for all events and all predictions reported by the model. The recall (Equation 5) is the ratio between correct predictions for all events and all trigger/arguments that should be identified/classified. The f1 score (Equation 6) is the harmonic mean between the precision and the recall.

$$\mathbf{precision} = \frac{\mathit{true\ positives}}{\mathit{true\ positives} + \mathit{false\ positives}} \quad (4)$$

$$\mathbf{recall} = \frac{\mathit{true\ positives}}{\mathit{true\ positives} + \mathit{false\ negatives}} \quad (5)$$

$$\mathbf{F_1} = 2 * \frac{\mathit{precision} * \mathit{recall}}{\mathit{precision} + \mathit{recall}} \quad (6)$$

4.2. Hyperparameters and Details of Fine-Tune

We utilize the BERT-base to build our baseline model. Fine-tuning is performed on a single GPU with batch size 32. We set the maximum BERT sequence length to 256. Shorter sequences are padded and no sequences exceed this limit. We train the model using Adam optimizer at learning rate $2e-5$ with weight decay 0.01 until converge.

4.3. Event Corpus

In this section, we describe how we build the event corpus for further pre-training BERT. We notice that almost half of the original data in ACE2005 comes from newswire and broadcast news. And as an event extraction data set, it contains a wide range of topics and event types. Therefore, to cover all the ACE2005 events, we utilize the New York Times Annotated Corpus [23] which contains over 1.8 million articles written and published by the New York Times to build our event corpus. NYT is a very large dataset, pretraining with all the data requires a lot of computing resources which can be very expensive. On the other hand, not all articles in NYT involves useful topics that can help improve the performance of ACE2005 task (for example, many articles are related to company report, biographical information, eta) Therefore, we preprocess the NYT corpus by manually selecting articles related to ACE-defined event types. Concretely, each article in NYT corpus is released with metadata and the “descriptors” field specifies a list of descriptive terms corresponding to subjects mentioned in the article, many subjects in NYT corpus have a very strong relation with the ACE predefined event subtypes. We screened the news documents with the most similar topics to each event type to form our corpus, see Table 2 for details.

We ended up with 290409 articles, including 150M words in total as our event corpus. Notice that the total article number is slightly smaller than the sum of all subjects (321356) because some articles may have several different subjects.

Table 2. Components of event corpus

ACE2005 Data Set		NYT Corpus	
Event Type	Event Subtype	Selected Subjects	#article
Life	Be-Born, Marry, Divorce 、 Injure, Die	weddings and engagements	43848
		deaths	24486
		murders and attempted murders	12804
		accidents and safety	10690
Movement	Transport	armament, defense and military forces	11309
Transaction	Transfer-Ownership, Transfer-Money	finances	26342
Personnel	Start-Position, End-Position 、 Elect, Nominate	suspensions, dismissals and resignations	16392
		appointments and executive changes	25227
		elections	23668
Contact	Meet, Phone-Write	united states international relations	20390
Conflict	Demonstrate, Attack	civil war and guerrilla warfare	15657
		bombs and explosives	5583
		demonstrations and riots	7750
Justice	Acquit, Charge-Indict, Arrest-Jail, Release-Parole 、 Sue, Convict, Appeal, Sentence, Trial-Hearing, Fine, Execute, Extradite, Pardon	suits and litigation	23808
		decisions and verdicts	5188
		trials	5381
Business	Merge-Org, Start-Org, Declare-Bankruptcy, End- Org	mergers, acquisitions and divestitures	32903
		reform and reorganization	9930

4.4. Hyperparameters and Details of Further Pre-Training

We create training examples using our event corpus with dupe factor 5, each example consists of a pair of sentences with some tokens masked for MLM and NSP object. The maximum sequence length is 256 which is consistent with the fine-tuning stage. Start from the original BERT checkpoint, the model is further pre-trained on a cloud TPU for 200k steps of batch size 384 at learning rate $2e-5$.

4.5. Effect of Event Knowledge

Table 3 shows the effect of event knowledge. The event extraction system based on original pre-trained BERT already achieves a fairly considerable score (73.3% f1 score on trigger classification and 58.4% f1 score on argument classification). After further updating the model through the event corpus, we observed the model (denoted by **Event BERT**) achieve better performance over all metrics on both trigger and argument identification/classification task. It gains 1.8% f1 score improvement on trigger classification and 2.3% f1 score improvement on argument classification which shows the benefits of having in-domain event knowledge.

Table 3. Effect of event knowledge.

Models	Trigger Identification			Trigger Classification			Argument Identification			Argument Classification		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
BERT	78.0	75.7	76.8	74.5	72.3	73.3	60.7	64.1	62.4	56.7	60.1	58.4
EventBERT	78.1	78.0	78.1	75.2	75.0	75.1	62.6	64.6	63.6	59.7	61.8	60.7

5. CONCLUSION AND FUTURE WORK

In this paper, we propose an event extraction system based on the pre-trained language model for both event trigger and argument extraction. We explore a new way of using external corpus. An elaborately constructed event corpus is built to improve the ACE2005 event extraction task by further pretraining the BERT language model. Experimental results show that our method is very effective and achieve around 2% improvement while avoiding designing complex event generation processes and rules.

We believe the idea of injecting in-domain knowledge by further pretraining the BERT can be helpful to other different NLP tasks especially for which generating extra training data is hard and painful. However, one major limitation is that a corpus that contain specific in-domain knowledge is required for each different task. For ACE2005 event extraction task, building such a corpus is easy as the ACE2005 dataset involves just common topic. But this is not the case for many other tasks that involves specialized fields knowledge or lacks relative resources.

Therefore, one possible direction for future work is to minimize the cost of constructing the knowledge corpus when applying our method to other tasks. One way to achieve it would be to transfer knowledge from one task to another so that we can reuse the knowledge corpus. It is to be verified that our model can also improve some similar task like KBP event extraction task.

ACKNOWLEDGMENTS

This research is supported by the grants from the National Key Research and Development Program of China (No. 2019YFB1705601).

REFERENCES

- [1] Walker, C., Strassel, S., Medero, J., & Maeda, K. (2006). ACE 2005 multilingual training corpus. Linguistic Data Consortium, Philadelphia.
- [2] Riloff, E. (1993). Automatically Constructing a Dictionary for Information Extraction Tasks. Proceedings of the Eleventh National Conference on Artificial Intelligence (pp. 811–816).
- [3] Cao, K., Li, X., Ma, W., & Grishman, R. (2018). Including New Patterns to Improve Event Extraction Systems. FLAIRS Conference.
- [4] Li, Q., Ji, H., & Huang, L. (2013, 8). Joint Event Extraction via Structured Prediction with Global Features. Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 73–82).
- [5] Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., & Zhu, Q. (2011, 6). Using Cross-Entity Inference to Improve Event Extraction. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies (pp. 1127–1136)
- [6] Chen, Y., Xu, L., Liu, K., Zeng, D., & Zhao, J. (2015, 7). Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).

- [7] Nguyen, T. H., Cho, K., & Grishman, R. (2016, 6). Joint Event Extraction via Recurrent Neural Networks. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 300–309).
- [8] Liu, X., Luo, Z., & Huang, H. (2018, 10). Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation. Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (pp. 1247–1256).
- [9] Abney, S. (2002). Bootstrapping. Proceedings of the 40th annual meeting of the association for computational linguistics, (pp. 360–367).
- [10] Liao, S., & Grishman, R. (2011, 11). Using Prediction from Sentential Scope to Build a Pseudo Co-Testing Learner for Event Extraction. Proceedings of 5th International Joint Conference on Natural Language Processing (pp. 714–722).
- [11] Li, W., Cheng, D., He, L., Wang, Y., & Jin, X. (2019). Joint Event Extraction Based on Hierarchical Event Schemas From FrameNet. *IEEE Access*, 7, 25001-25015.
- [12] Liu, S., Chen, Y., He, S., Liu, K., & Zhao, J. (2016, 8). Leveraging FrameNet to Improve Automatic Event Detection. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 2134–2143).
- [13] Chen, Y., Liu, S., Zhang, X., Liu, K., & Zhao, J. (2017, 7). Automatically Labeled Data Generation for Large Scale Event Extraction. Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 409–419).
- [14] Araki, J., & Mitamura, T. (2018, 8). Open-Domain Event Detection using Distant Supervision. Proceedings of the 27th International Conference on Computational Linguistics (pp. 878–891).
- [15] Radford, A., Narasimhan, K., Salimans, T., & Sutskever, I. (2018). Improving language understanding by generative pre-training. Improving language understanding by generative pre-training.
- [16] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019, 6). 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, Volume 1 (Long and Short Papers) (pp. 4171–4186).
- [17] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., & Kang, J. (2019, 9). BioBERT: a pre-trained biomedical language representation model for biomedical text mining. (J. Wren, Ed.) *Bioinformatics*.
- [18] Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. SciBERT: A Pretrained Language Model for Scientific Text.
- [19] Xu, H., Liu, B., Shu, L., & Yu, P. (2019, 6). BERT Post-Training for Review Reading Comprehension and Aspect-based Sentiment Analysis. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 2324–2335).
- [20] Rietzler, A., Stabinger, S., Opitz, P., & Engl, S. (2020, 5). Adapt or Get Left Behind: Domain Adaptation through BERT Language Model Finetuning for Aspect-Target Sentiment Classification. Proceedings of The 12th Language Resources and Evaluation Conference (pp. 4933–4941).
- [21] Sung, C., Dhamecha, T., Saha, S., Ma, T., Reddy, V., & Arora, R. (2019, 11). Pre-Training BERT on Domain Resources for Short Answer Grading. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP) (pp. 6071–6075).
- [22] Sun, C., Huang, L., & Qiu, X. (2019, 6). Utilizing BERT for Aspect-Based Sentiment Analysis via Constructing Auxiliary Sentence. Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (pp. 380–385).
- [23] Sandhaus, E. (2008). The new york times annotated corpus. Linguistic Data Consortium, Philadelphia, 6, e26752.

AUTHORS

Zining Yang is a postgraduate student at the School of Computer Science and Engineering at University of Electronic Science & Technology of China (UESTC), Chengdu, China. He received his B.S degree also at UESTC in 2015. His main research interests include natural language processing, data storage, and data mining.



Siyu Zhan is currently an associate professor at the School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC). He was a visiting scholar at the Electrical and Computer Engineering Department at Virginia Polytechnic Institute and State University (Virginia Tech) on 2007 and at Computer Science Department at Wayne State University on 2017. His interests include distributed computer system, machine learning, wireless communications, networking and software engineering.



Mengshu Hou is a professor in the School of Computer science &Engineering at the University of Electronic Science and Technology of China (UESTC). He received the M.S and Ph.D. degrees in 2002 and 2005 respectively from the UESTC.



Xiaoyang Zeng is currently a Ph.D. at the Department of computer science and Engineering, University of Electronic Science and Technology (UESTC), Chengdu, China. He received the B.S. degrees in Southwest Petroleum University in 2018, and passed the successive master-doctor program and is studying in UESTC. His research interests focus on natural language processing and text mining.



Hao Zhu is an engineer in the Information Center at the University of Electronic Science and Technology of China (UESTC). He received the B.S and M.S degrees in 2002 and 2006 respectively from the UESTC. His current research interests include management informatization, data visualization, and big data analysis.

