DCFEE: A Document-level Chinese Financial Event Extraction System based on Automatically Labeled Training Data

Hang Yang[1], **Yubo Chen**[1], **Kang Liu**[1,2] , **Yang Xiao**[1] and **Jun Zhao**[1,2]
[1] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences, Beijing, 100049, China
{hang.yang, yubo.chen, kliu, yang.xiao, jzhao,}@nlpr.ia.ac.cn

Abstract

We present an event extraction framework to detect event mentions and extract events from the document-level financial news. Up to now, methods based on supervised learning paradigm gain the highest performance in public datasets (such as ACE 2005[1], KBP 2015[2]). These methods heavily depend on the manually labeled training data. However, in particular areas, such as financial, medical and judicial domains, there is no enough labeled data due to the high cost of data labeling process. Moreover, most of the current methods focus on extracting events from one sentence, but an event is usually expressed by multiple sentences in one document. To solve these problems, we propose a Document-level Chinese Financial Event Extraction (DCFEE) system which can automatically generate a large scaled labeled data and extract events from the whole document. Experimental results demonstrate the effectiveness of it.

1 Introduction

Event Extraction (EE), a challenging task in Nature Language Processing (NLP), aims at discovering event mentions[3] and extracting events which contain event triggers[4] and event arguments[5] from texts. For example, in the sentence E1[6] as shown in Figure 1, an EE system is expected to discover an *Equity Freeze* event mention (E1 itself) triggered by **frozen** and extract the corresponding five arguments with different roles: *Nagafu Ruihua* (Role=Shareholder Name), *520,000 shares* (Role=Num of Frozen Stock), *People's Court of Dalian city* (Role=Frozen Institution), *May 5,2017* (Role=Freezing Start Date) and *3 years* (Role=Freezing End Date). Extracting event instances from texts plays a critical role in building NLP applications such as Information Extraction (IE), Question Answer (QA) and Summarization (Ahn, 2006). Recently, researchers have built some English EE systems, such as EventRegistry[7] and Stela[8]. However, in financial domain, there is no such effective EE system, especially in Chinese.

Financial events are able to help users obtain competitors' strategies, predict the stock market and make correct investment decisions. For example, the occurrence of an *Equity Freeze* event will have a bad effect on the company and the shareholders should make correct decisions quickly to avoid the losses. In business domain, official announcements released by companies represent the occurrence of major events, such as *Equity Freeze* events, *Equity Trading* events and so on. So it is valuable to discover event mention and extract events from the announcements. However, there are two challenges in Chinese financial EE.

Lack of data: most of the EE methods usually adopted supervised learning paradigm which relies on elaborate human-annotated data, but there is no labeled corpus for EE in the Chinese financial field.

Document-level EE: most of the current methods of EE are concentrated on the sentence-level

[1] http://projects.ldc.upenn.edu/ace/
[2] https://tac.nist.gov//2015/KBP/
[3] A sentence that mentions an event, including a distinguished trigger and involving arguments.
[4] The word that most clearly expresses the occurrence of an event.
[5] The entities that fill specific roles in the event.
[6] All the examples in this article are translated from Chinese.
[7] http://eventregistry.org/
[8] https://www.nytsyn.com/

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics-System Demonstrations, pages 1–6
Melbourne, Australia, July 15 - 20, 2018.

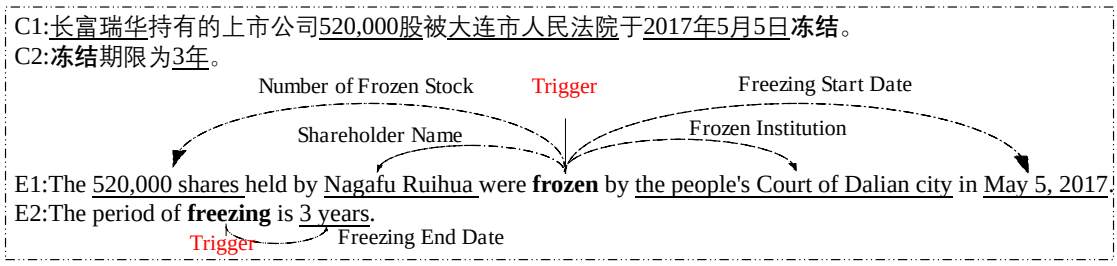

Figure 1: Example of an *Equity Freeze* event triggered by "frozen" and containing five arguments.

text (Chen et al., 2015) (Nguyen et al., 2016). But an event is usually expressed with multiple sentences in a document. In the financial domain data set constructed in this paper, there are 91% of the cases that the event arguments are distributed in the different sentences. For example, as shown in Figure 1, E1 and E2 describe an *Equity Freeze* event together.

To solve the above two problems, we present a framework named DCFEE which can extract document-level events from announcements based on automatically labeled training data. We make use of Distance Supervision (DS) which has been validated to generate labeled data for EE (Chen et al., 2017) to automatically generate large-scaled annotated data. We use a sequence tagging model to automatically extract sentence-level events. And then, we propose a key-event detection model and an arguments-filling strategy to extract the whole event from the document.

In summary, the contributions of this article are as follows:

- We propose the DCFEE framework which can automatically generate large amounts of labeled data and extract document-level events from the financial announcements.
- We introduce an automatic data labeling method for event extraction and give a series of useful tips for constructing Chinese financial event dataset. We propose a document-level EE system mainly based on a neural sequence tagging model, a key-event detection model, and an arguments-completion strategy. The experimental results show the effectiveness of it.
- The DCFEE system has been successfully built as an online application which can quickly extract events from the financial announcements [9].

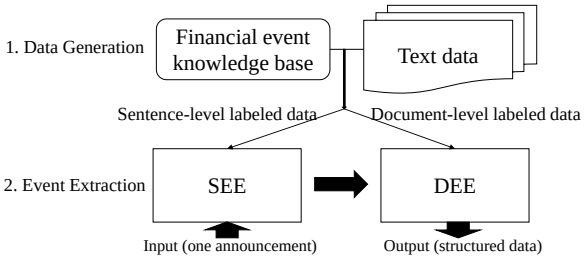

Figure 2: Overview of the DCFEE framework.

2 Methodology

Figure 2 describes the architecture of our proposed DCFEE framework which primarily involves the following two components: (i) Data Generation, which makes use of DS to automatically label event mention from the whole document (document-level data) and annotate triggers and arguments from event mention (sentence-level data); (ii) EE system, which contains Sentence-level Event Extraction (SEE) supported by sentence-level labeled data and Document-level Event Extraction (DEE) supported by document-level labeled data. In the next section, we briefly describe the generation of labeled data and architecture of the EE system.

2.1 Data Generation

Figure 3 describes the process of labeled data generation based on the method of DS. In this section, we first introduce the data sources (structured data and unstructured data) that we use. And then we describe the method of automatically labeling data. Finally, we will introduce some tips that can be used to improve the quality of the labeled data.

[9] http://159.226.21.226/financial_graph/online-extract.html

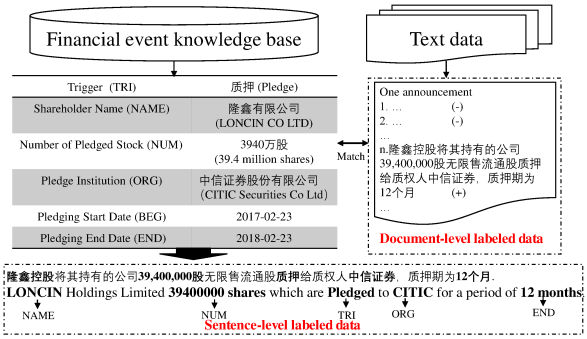

Figure 3: The process of labeled data generation.

Data sources: two types of data resources are required to automatically generate data: a financial event knowledge database containing a lot of structured event data and unstructured text data containing event information. (i) The financial event knowledge database used in this paper is structured data which includes nine common financial event types and is stored in a table format. These structured data which contains key event arguments is summarized from the announcements by financial professionals. An *Equity Pledge* event is taken as an example, as shown on the left of Figure 3, in which key arguments include Shareholder Name (NAME), Pledge Institution (ORG), Number of Pledged Stock (NUM), Pledging Start Date (BEG) and Pledging End Date (END). (ii) The unstructured text data come from official announcements released by the companies which are stored in an unstructured form on the web. We obtain these textual data from Sohu securities net[10].

Method of data generation: annotation data consists of two parts: sentence-level data generated by labeling the event trigger and event arguments in the event mention; document-level data generated by labeling the event mention from the document-level announcement. Now the question is, how to find the event triggers. Event arguments and event mention that correspond to the structured event knowledge database are summarized from a mass of announcements. DS has proved its effectiveness in automatically labeling data for Relation Extraction (Zeng et al., 2015) and Event Extraction (Chen et al., 2017). Inspired by D-S, we assume that one sentence contains the most event arguments and driven by a specific trigger is likely to be an event mention in an announcement. And arguments occurring in the event mention are likely to play the corresponding roles in the event. For each type of financial event, we construct a dictionary of event triggers such as **frozen** in *Equity Freeze* event and **pledged** in *Equity Pledge* event. So the trigger word can be automatically marked by querying the pre-defined dictionary from the announcements. through these pretreatments, structured data can be mapped to the event arguments within the announcements. Therefore, we can automatically identify the event mention and label the event trigger and the event arguments contained in it to generate the sentence-level data, as shown at the bottom of Figure 3. Then, the event mention is automatically marked as a positive example and the rest of the sentences in the announcement are marked as negative examples to constitute the document-level data, as shown on the right of Figure 3. The document-level data and the sentence-level data together form the training data required for the EE system.

Tips: in reality, there are some challenges in data labeling: the correspondence of financial announcements and event knowledge base; the ambiguity and abbreviation of event arguments. There are some tips we used to solve these problems, examples are shown in Figure 3.
(i) Decrease the search space: the search space of candidate announcements can be reduced through retrieving key event arguments such as the publish date and the stock code of the announcements.
(ii) Regular expression: more event arguments can be matched to improve the recall of the labeled data through regular expression. for example, *LONCIN CO LTD* (Role=Shareholder Name) in the financial event database, but *LONCIN* in the announcement. We can solve this problem by regular expression and label the *LONCIN* as an event argument
(iii) Rules: some task-driven rules can be used to automatically annotate data. for example, we can mark *12 months* (Role=Pledging End Date) by calculating the date difference between *2017-02-23* (Role=Pledging Start Date) and *2018-02-23* (Role=Pledging End Date).

2.2 Event Extraction (EE)

Figure 4 depicts the overall architecture of the EE system proposed in this paper which primarily involves the following two components: The sentence-level Event Extraction (SEE) purposes to

[10]http://q.stock.sohu.com/cn/000001/gsgg.shtml

extract event arguments and event triggers from one sentence; The document-level Event Extraction (DEE) aims to extract event arguments from the whole document based on a key event detection model and an arguments-completion strategy.

2.2.1 Sentence-level Event Extraction (SEE)

We formulate SEE as a sequence tagging task and the training data supported by sentence-level labeled data. Sentences are represented in the BIO format where each character (event triggers, event arguments and others) is labeled as B-label if the token is the beginning of an event argument, I-label if it is inside an event argument or O-label if it is otherwise. In recent years, neural networks have been used in most NLP tasks because it can automatically learn features from the text representation. And the Bi-LSTM-CRF model can produce state of the art (or close to) accuracy on classic NLP tasks such as part of speech (POS), chunking and NER (Huang et al., 2015). It can effectively use both past and future input features thanks to a Bidirectional Long Short-Term Memory (BiLSTM) component and can also use sentence-level tag information thanks to a Conditional Random Field (CRF) layer.

The specific model implementation of the SEE, as shown on the left of Figure 4, is made up of a Bi-LSTM neural network and a CRF layer. Each Chinese character in a sentence is represented by a vector as the input of the Bi-LSTM layer[11] (Mikolov et al., 2013). The output of the Bi-LSTM layer is projected to score for each character. And a CRF layer is used to overcome the label-bias problem. The SEE eventually returns the result of the sentence-level EE for each sentence in the document.

2.2.2 Document-level Event Extraction(DEE)

The DEE is composed of two parts: a key event detection model which aims to discover the event mention in the document and an arguments-completion strategy which aims to pad the missing event arguments.

Key event detection: as shown on the right of figure 4, the input of the event detection is made up of two parts: one is the representation of the event arguments and event trigger come from the output of SEE (blue), and the other is the vector representation of the current sentence(red). The

[11]Word vectors are trained with a version of word2vec on Chinese WiKi corpus

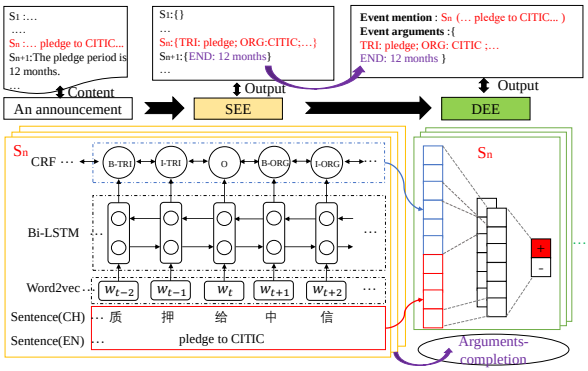

Figure 4: The architecture of event extraction.

two parts are concatenated as the input feature of the Convolutional Neural Networks(CNN) layer. And then the current sentence is classified into two categories: a key event or not.

Arguments-completion strategy: We have obtained the key event which contains most of the event arguments by the DEE, and the event extraction results for each sentence in a document by the SEE. For obtaining complete event information, we use arguments-completion strategy which can automatically pad the missing event arguments from the surrounding sentences. As shown in figure 4, an integrated *Pledge* event contains event arguments in event mention S_n and filled event argument *12 months* obtained from the sentence S_{n+1}.

3 Evaluation

3.1 Dataset

We carry out experiments on four types of financial events: *Equity Freeze (EF)* event, *Equity Pledge (EP)* event, *Equity Repurchase (ER)* event and *Equity Overweight (EO)* event. A total of 2976 announcements have been labeled by automatically generating data. We divided the labeled data into three subsets: the training set (80% of the total number of announcements), development set (10%) and test set (10%). Table 1 shows the statistics of the dataset. NO.ANN represents the number of announcements can be labeled automatically for each event type. NO.POS represents the total number of positive case sentences (event mentions). On the contrary, NO.NEG represents the number of negative case sentences. The positive and negative case sentences constitute the document-level data as the training data for the DEE. The positive sentences which contain event

trigger and a series of event arguments, are labeled as sentence-level training data for the SEE.

Dataset	NO.ANN	NO.POS	NO.NEG
EF	526	544	2960
EP	752	775	6392
EB	1178	1192	11590
EI	520	533	11994
Total	2976	3044	32936

Table 1: Statistics of automatically labeled data.

We randomly select 200 samples (contain 862 event arguments) to manually evaluate the precision of the automatically labeled data. The average precision is shown in Table 2 which demonstrates that our automatically labeled data is of high quality.

Stage	Mention labeling	Arguments Labeling
Number	200	862
Average Precision	94.50	94.08

Table 2: Manual Evaluation Results.

3.2 Performance of the System

We use the $precision(P)$, $recall(R)$ and (F_1) to evaluate the DCFEE system. Table 3 shows the performance of the pattern-based method[12] and the DCFEE in the extraction of the *Equity Freeze* event. The experimental results show that the performance of the DCFEE is better than that of the pattern-based method in most event arguments extraction.

Method	Pattern-based			DCFEE		
Type	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
ORG	79.44	72.22	**75.66**	88.41	61.62	72.62
NUM	57.14	54.55	55.81	59.20	52.02	**56.38**
NAME	63.84	57.07	60.27	89.02	73.74	**80.66**
BEG	65.79	63.13	64.43	81.88	61.62	**70.42**
END	67.62	35.86	46.86	85.00	68.00	**75.56**

Table 3: P, R, F_1 of pattern-based and DCFEE on the *Equity Freeze* event.

Table 4 shows the P, R, F_1 of SEE and DEE on the different event types. It is noteworthy that the golden data used in SEE stage is the automatically generated data and the golden data used in DEE stage comes from the financial event knowledge base. The experimental results show that the effectiveness of SEE and DEE, the acceptable precision and expansibility of the DCFEE system presented in this paper.

Stage	SEE			DEE		
Type	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
EF	90.00	90.41	90.21	80.70	63.40	71.01
EP	93.31	94.36	93.84	80.36	65.91	72.30
ER	92.79	93.80	93.29	88.79	82.02	85.26
EO	88.76	91.88	90.25	80.77	45.93	58.56

Table 4: P, R, F_1 of SEE, DEE on the different event types.

In conclusion, the experiments show that the method based on DS can automatically generate high-quality labeled data to avoid manually labeling. It also validates the DCFEE proposed in this paper, which can effectively extract events from the document-level view.

4 Application of the DCFEE

The application of the DCFEE system: an online EE service for Chinese financial texts. It can help financial professionals quickly get the event information from the financial announcements. Figure 5 shows a screenshot of the online DCFEE system. Different color words represent different event arguments' types, underlined sentences represent the event mention in the document. As shown in figure 5, we can obtain a complete *Equity Freeze* event from unstructured text (an announcement about *Equity Freeze*).

Figure 5: A screen shot of the online DCFEE system[9].

5 Related Work

The current EE approaches can be mainly classified into statistical methods, pattern-based method and hybrid method (Hogenboom et al., 2016).

[12]Example of a pattern for a freeze event: (*Frozen institution(ORG)+, Trigger word(TRI)+, Shareholder names(NAME)+,time*)

Statistical method can be divided into two categories: traditional machine learning algorithm based on feature extraction engineering (Ahn, 2006), (Ji and Grishman, 2008), (Liao and Grishman, 2010), (Reichart and Barzilay, 2012) and neural network algorithm based on automatic feature extraction (Chen et al., 2015), (Nguyen et al., 2016), (Liu et al., 2017). The pattern method is usually used in industry because it can achieve higher accuracy, but meanwhile a lower recall. In order to improve recall, there are two main research directions: build relatively complete pattern library and use a semi-automatic method to build trigger dictionary (Chen et al., 2017), (Gu et al., 2016). Hybrid event-extraction methods combine statistical methods and pattern-based methods together (Jungermann and Morik, 2008), (Bjorne et al., 2010). To our best knowledge, there is no system that automatically generates labeled data, and extracts document-level events automatically from announcements in Chinese financial field.

6 Conclusion

We present DCFEE, a framework which is able to extract document-level events from Chinese financial announcements based on automatically labeled data. The experimental results show the effectiveness of the system. We successfully put the system online and users can quickly get event information from the financial announcements through it9.

7 Acknowledgements

This work was supported by the Natural Science Foundation of China (No. 61533018). And this work is also supported by a grant from Ant Financial Services Group.

References

David Ahn. 2006. The stages of event extraction. In *The Workshop on Annotating and Reasoning about Time and Events*, pages 1–8.

Jari Bjorne, Filip Ginter, Sampo Pyysalo, Jun'Ichi Tsujii, and Tapio Salakoski. 2010. Complex event extraction at pubmed scale. *Bioinformatics*, 26(12):i382–i390.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the ACL*, pages 409–419.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the ACL*.

Jiatao Gu, Zhengdong Lu, Hang Li, and O.K. Victor Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the ACL*, pages 1631–1640.

Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, Franciska De Jong, and Emiel Caron. 2016. A survey of event extraction methods from text for decision support systems. *Decision Support Systems*, 85(C):12–22.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *Computer Science*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of the ACL*, pages 254–262.

Felix Jungermann and Katharina Morik. 2008. *Enhanced Services for Targeted Information Retrieval by Event Extraction and Data Mining*. Springer Berlin Heidelberg.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the ACL*, pages 789–797.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the ACL*, pages 1789–1798.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Computer Science*.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the ACL*, pages 300–309.

Roi Reichart and Regina Barzilay. 2012. Multi event extraction guided by global constraints. In *Proceedings of the NAACL*, pages 70–79.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the EMNLP*, pages 1753–1762.