Proceedings of CCIS2019

Event Extraction via Rules and Machine Learning

Xuepan Gao[1], Zhengang Diao[2], Kailin Wei[2], Yuzhao Yang[2], Lei Li[2]

[1]The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China
[2]Beijing University of Posts and Telecommunications, Beijing 100876, China
gaoxuepan@gmail.com, diaozhengang@bupt.edu.cn,weiklemail@163.com

Abstract: Event extraction is a challenging task of Information Extraction(IE). Its target is to extract structured information from the natural language texts, which can be divided into two steps, identifying the type of a event by trigger words and extracting the arguments. This paper proposes a method which combines rule matching and deep learning to implement event extraction. To achieve the best performance, we apply the method of rules to arguments which show a strong regularity in their expressions and apply Bootstrapping to generate rules. For the arguments with rich expressions, we propose a more appropriate machine learning method for each class of arguments and a textual feature representation method which can capture more information by adding location, part-of-speech (POS), named entity recognition (NER) and dependency parsing(DEP) to word embedding (WE). Experiments have proven the validity of our prediction method and the combination feature for each class of arguments.
Keywords: Event extraction; Rule matching; Deep learning; Bootstrapping

1 Introduction

The main function of the event extraction system is to extract specific factual information which can be called argument from the text, for example, person, time, place, mission, etc. At present, there are two main methods for implementing event extraction: one is based on the knowledge engineering, which mainly uses the method of rule matching to implement extraction; the other is based on the statistical learning, which is generally implemented by means of machine learning.

To achieve the best performance, we apply the method of rule to arguments which have a strong regularity in expression to get better accuracy and response time. In order to alleviate the disadvantages of rules, complex and difficult to cover all the language phenomena, we apply Bootstrapping to generate rules. For the arguments with rich expressions, we attempt to explore the best performance from two aspects: features and methods of machine learning.

The main contributions in this paper are:

1) WE is an important feature in event extraction. However, word embedding fails to capture some specific word information of location, POS, NER, DEP and etc. In this paper, we propose a textual feature representation method by adding location, POS, NER and DEP to WE.

978-1-7281-3863-3/19/$31.00 ©2019 IEEE

2) In this paper, we analyze the characteristics of different arguments and propose a more appropriate approach for each class of arguments.

3) Experiments have proved that our prediction method and the combination feature are effective for the military event extraction. And pretty good performance for each argument can be obtained.

2 Related work

The research on the event extraction is mainly published at the MUC [1] and ACE [2] conferences. It is divided into two tasks: event category identification and event element extraction. In order to accomplish these two tasks, the methods selected are roughly divided into two categories:
The first method is the knowledge engineering based on rule matching. Typical event extraction systems based on rule matching are ExDisco, GenPAM, etc. At first, the model was created by manual methods. With the development of statistics and machine learning techniques, various methods of automatic learning acquisition have emerged. For instance, in the aspect of commodity attribute extraction, Wang Hui et al. [3] use rules to extract product attribute information based on Bootstrapping extraction method.
The second method is machine learning. According to different driving sources selected by the model, the current methods can be roughly divided into three categories:

a. Event element driven: Chieu [4] introduces the maximum entropy classifier for event identification in the event argument extraction for the first time. H. Llorens et al. [5] use the CRF model for semantic role labeling and apply it to TimeML event extraction to improve system performance.

b. Trigger-driven: Zhao Yanyan et al. [6] use a method based on trigger word extension and binary classification to identify event categories, which solves the problem of positive and negative examples imbalance and data sparsity. Chen Yadong et al. [7] use the CRF model to treat trigger word recognition as a sequence labeling task, which means that the trigger word is no longer limited to a single word, but may also be a phrase. Zeng et al. [8] use two sequence labeling models consisting of a fusion model of bidirectional long-term memory model (BLSTM) and a conditional random field model (CRF) to obtain key argument label sequences corresponding to each event type and identify the key arguments in the sentence. If a sentence contains all of the key arguments for a particular event type, it is treated as an event of that type. The definition of key arguments remains a question worthy of scrutiny.

c. Event instance driven: Naughton [9] does a lot of research on sentence-level event extraction, use each sentence as a candidate event instance, thus turns the event extraction into a sentence classification problem. Nguyen et al. [10] use entity type features to help event recognition tasks, while Chen et al. [11] use event type features for role classification tasks. In the upper-level feature learning model, the convolutional neural network model is used in the study to automatically extract effective features for event extraction or recognition tasks. Furthermore, Chen et al. [12] add dynamic multi-pooling based on the traditional CNN model, and divide the candidate trigger words and candidate entity locations into three parts, trying to obtain more precise and effective features used for role classification.

3 Process flow of method

The target of this research is to extract event from the natural language texts of military, and extract argument elements such as subject, receptor, time, quantity, and trigger words. The structure of our system is shown in Figure 1.

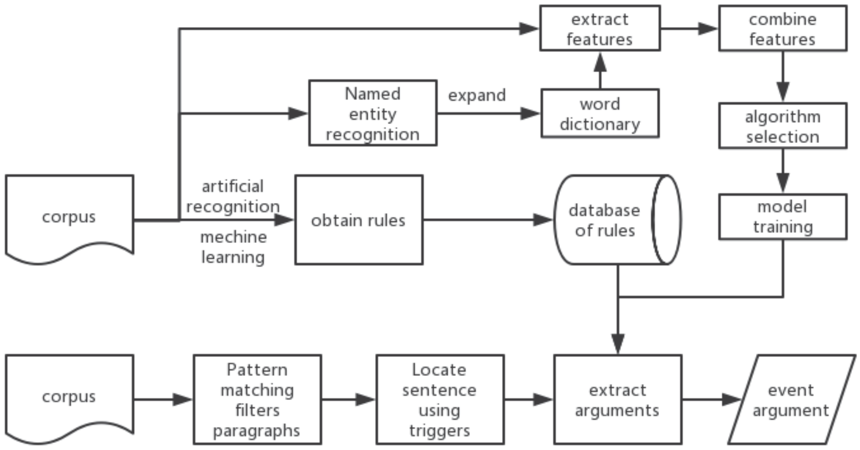

Figure 1 Structure of Our System

3.1 Filter paragraph by pattern matching

The purpose of this sub-module is to extract the event of the military activities of a certain area from the original text. The original text contains a large number of articles and passages on the military activities of other countries. Usually a paragraph has a strong correlation, the description is basically about the military activities of the same country. Through the paragraph pattern matching, relevant content can be retrained as much as possible.

3.2 Triggers

Triggers can locate and identify a certain argument extraction, by using it to narrow the extraction range. When a trigger word of the event appears in a sentence in the document, the extraction task of the corresponding argument can be triggered in the sentence. And the sentence without any trigger word is filtered directly, thereby greatly narrows the extraction range and improves extraction efficiency.

3.3 Text preprocessing

Implement Named Entity Recognition (NER) on the original text, and add the identified related entities to the dictionary, then use the dictionary to assist word segmentation.

Use LTP word segmentation tool to implement segmentation and tag part-of-speech. An example of segmentation and part-of-speech tagging is shown in Figure 2.

某军1架XX型机4月26日拟飞往XX基地
某军:j 1架:m XX:ws 型机:n 4月:nt 26日:nt 拟:v 飞往:v XX:ns 基地:n

Figure 2 Segmentation and Part-Of-Speech Tagging

Carry out the dependency parsing analysis on the words combined with the word-of-speech to prepare for the subsequent requests. An example of dependency parsing analysis is shown in Figure 3.

```
某军 XX型机 ATT
1架 XX型机 ATT
XX型机 飞往 SBV
4月26日 飞往 ADV
拟 飞往 ADV
飞往 。 HED
XX基地 飞往 VOB
。 飞往 WP
```

Figure 3 Dependency Parsing Analysis

3.4 Formulate rule seed and build extraction rules

Although the method of manually formulating rules has a high accuracy in extracting arguments, it is difficult to cover all the language phenomena. Therefore, we adopt Bootstrapping to generate rules.

Bootstrapping is a machine learning algorithm. The core idea of this algorithm is that the same or similar contextual language units such as characters, words, phrases, should have same patterns or features to some extent, which expand the rule seed through such characteristics.

Based on the Bootstrapping algorithm, the following diagram is used to generate the rules through extending the rule seed. The flowchart of Bootstrapping is shown in Figure 4.

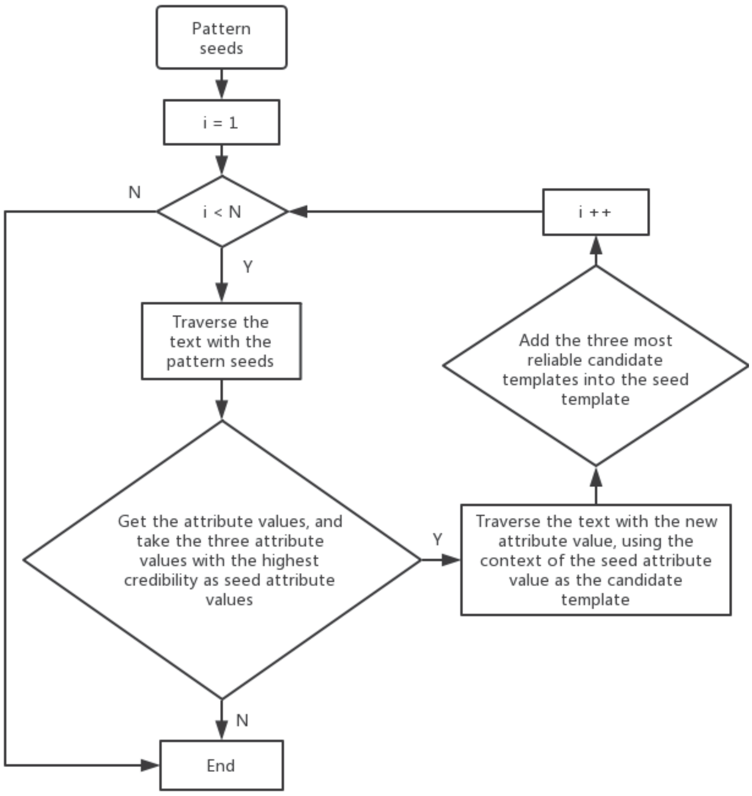

Figure 4 Bootstrapping Algorithm Framework

The Bootstrapping algorithm generation rule mainly includes the following steps.

a. Manually formulate a rule seed as an initial rule set S_{PATTER} and an attribute value set S_{ATTR};

b. Using the rule seed set S_{PATTER}, traverse and match the attribute values in the text to obtain the candidate attribute value set h;

c. Calculate the credibility of each attribute value in the candidate attribute value set h, and add the three attribute values with higher credibility to the seed attribute value set S_{ATTR}. If the S_{ATTR} converges, the algorithm ends. Otherwise go to step d;

d. Using the attribute value set S_{ATTR}, traverse the text and generate a candidate template set h' from the context 2 words of the matched attribute value;

e. Calculate the credibility of each candidate template in the candidate attribute value set h', and add three candidate templates with higher credibility to the seed attribute value set S_{PATTER}. If the S_{PATTER} converges, the algorithm ends. Otherwise, perform step b;

Repeat steps b-e until a certain number of iterations is reached. The attribute value and template credibility are calculated using the following formula.

The credibility of the attribute value is calculated as the formula 1.

$$value(W) = P(W) * \log(P(W)) / ALL\_P \quad (1)$$

P(W) represents the number of attribute values W obtained by the template in the seed template set; ALL_P represents the number of all templates obtained by the attribute value W in the training corpus.

The template credibility is calculated as the formula 2.

$$value(P) = \sum (P(W_i) * \log(P(W_i))) / ALL\_W \quad (2)$$

$P(w_i)$ represents the number of attribute values W_i obtained by the template in the seed template set for each word W_i acquired by the P template; ALL_W represents the number of all words obtained by the template P in the training corpus.

4 Methods based on machine learning

Some arguments in the event, such as subject, receptor etc., have various ways of expression. This paper researches suitable machine learning methods to extract these arguments. The process can be expressed as Figure 5.

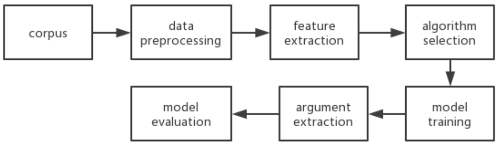

Figure 5 Process of Extraction based on Machine Learning

4.1 Data preprocessing

The model is extracted from the original Chinese documents. Therefore, some preprocessing is required before extracting features from the original text data. It includes the expansion of dictionary with named entity recognition, segmentation, part-of-speech tagging, relevant part-of-speech tagging combination and other steps.

4.2 Feature extraction

Feature selection plays a crucial role in event extraction. We summarize the work of our predecessors and combine the current popular deep learning results so that we put forward a number of features. In the field of machine learning, a feature is a description used to reflect the performances or characteristics of an event or object in a certain aspect. Different characteristics of the same event or object are different representations of the event or object. We can describe the same thing from many different dimensions, select the relevant features, exclude the irrelevant features, grasp the essence of things, and correctly describe it in the language of machine learning algorithm, finally has an important influence on the performance of the machine learning.

By using some mature language analysis techniques and relevant literature research, this paper selects four features for the implementation of event argument extraction. These four features are entity recognition information of word embedding, part-of-speech tagging and dependency syntax analysis.

Table Ⅰ Examples of Features other than Word Embedding

word	**X军**	**1架**	**XX型机**	**4月26日**	**拟**	**飞往**	**XX基地**	**。**
loc	1	2	3	4	5	6	7	8
POS	j	M	N	nt	v	v	N	wp
NER	army	O	Equipment	O	O	O	Target	O
parse	ATT	ATT	SBV	ADV	ADV	HED	VOB	WP

The word as the smallest unit containing complete semantic information, we believe that if it can be converted into a proper form as a feature that contains semantic information of the article, it will provide important information for machine learning algorithms.

Let $S = [w_1, w_2, \dots, w_n]$ be a sentence where n is the sentence length. $W^{wrd} \in R^{d^w|V|}$ is the embedded matrix, where d^w is the size of the user-defined word embedding, and V is the size of vocabulary. For each word in the sentence, we can get the word embedding by the formula 3.

$$w^e = W^{wrd} v^i \tag{3}$$

v^i is the one-hot representation of the current word in the corresponding column of W^{wrd}.

In addition, w^{kj} is a vector representation of features such as POS, NER, and DEP, where j represents the $j-th$ feature. The POS, NER and DEP are generated from the results of LTP tools.

Thus, for the sentence $S = [w_1, w_2, \dots, w_n]$, the feature representation of the word w_n can be expressed as formula 4:

$$w_n = [w_i^w, w_i^{k1}, w_i^{k2}, w_i^{k3}] \tag{4}$$

5 Experimental results and analysis

5.1 Dataset and model evaluation

In this paper, real military data about aircraft and ship activities and public data set provided by Microsoft research Asia are used for experiment, however the real military data is not convenient to open because of the confidentiality. Consistent with the evaluation methods of most event extraction studies, we used accuracy (P), recall rate(R) and F1-score values to evaluate the experimental results.

5.2 Experimental results

For different class of arguments with different characteristics, different machine learning methods should be adopted to find the most proper method for each class of arguments. LSTM, CRF, SVM and BILSTM+CRF are used for each class of arguments respectively.

5.2.1 Experiments on real military data

Experiments on subjects.

Table II Performance of different methods and features combinations under small data sets for subjects

Subjects	**CRF**	**BILSTM +CRF**	**B-LSTM**	**F-LSTM**	**BILSTM**
vec	**0.9410**	**0.9499**	**0.9645**	0.8872	**0.9097**
vec+POS	0.9303	0.9404	0.9061	**0.9390**	0.902
vec+POS +NER	0.9296	0.9483	0.9106	0.9293	0.8929
vec+POS +NER+ parse	0.9303	0.9404	0.9028	0.8909	0.8888
All	0.9284	0.9319	0.9602	0.8786	0.8992

The result of B-LSTM with vec works best for subject under small data sets. In real military texts, many commonly expressions do not conform to the regular grammar rules, so the addition of dependency parsing restricts the recognition of subject. Subjects are the types of aircraft and ships which are expressed in name or serial numbers, so NER and POS cannot bring good gain. More information will limit the performance and result only with word embedding can be better. Subjects are usually in the front of sentence, so it is more effective to consider the future state of the current moment for Chinese military text than the state before the current moment for arguments. The result of backward LSTM is a slightly better than that of forward LSTM.

Table Ⅲ Performance of BACKWARD LSTM under small data sets for subjects

Method	Tag	Precision	Recall	F1-score
B- LSTM	O	0.9761	0.9946	0.9852
	B	1.0000	0.5333	0.6957
	I	0.6364	0.6364	0.6364
	Avg/total	0.9675	0.9671	0.9645

Best performance for subjects as shown in Table 3.

Experiments on receptors.

Table IV Performance of different methods and features combinations under small data sets for receptors

Receptor	CRF	BILSTM+CRF	B-LSTM	F-LSTM	BILSTM
vec	0.9000	0.9417	0.9394	0.9097	0.919
vec+POS	0.9160	0.9587	**0.9422**	0.9692	0.9224
vec+POS+NER	0.9263	0.9634	0.9053	0.9157	0.9246
vec+POS+NER+parse	0.9326	0.9704	0.9097	0.9098	0.9122
All	**0.9346**	**0.9708**	0.9008	**0.9617**	**0.9662**

The class of receptor is different from subjects, receptors are usually a military base or a region which have a more fixed expression. The addition of NER, dependency parsing, POS and location can bring some improvements for the recognition of receptors. The receptor is usually at the end of the sentence so the addition of the state of before current moment can also play a positive role. As we can see in table IV, the result of F-LSTM is better than that of B-LSTM, and the result of BILSTM and BILSTM+CRF is slight better than that of F-LSTM.

Table V Performance of BILSTM+CRF under small data sets for receptors

Method	Tag	Precision	Recall	F1-score
BILSTM+CRF	O	0.9813	0.9919	0.9866
	B	0.9000	0.7500	0.8182
	I	0.7000	0.5833	0.6364
	Avg/total	0.9703	0.9722	0.9708

Best performance for receptors as shown in Table **VI**.

Table VI Performance of SVM under small data sets

	subjects	receptors
vec	0.9412	0.9128
vec+POS	0.9469	0.9109
vec+POS+NER	0.9469	0.9109
vec+POS+NER+parse	0.9389	0.9109
All	0.9365	0.9109

As we can see in table **VI**, adding other features to word embedding can get a noticeable effect for SVM. And the result of SVM is not better than that of deep learning.

Table VII Performance of SVM, CRF and BILSTM+CRF under big data sets for subjects

Method	Tag	Precision	Recall	F1-socre
SVM	O	0.9852	0.9934	0.9893
	B	0.8852	0.6129	0.7243
	I	0.8592	0.8821	0.8705
	Avg/total	0.9758	0.9765	0.9754
CRF	O	0.9833	0.9976	0.9904
	B	0.9387	0.7928	0.8596
	I	0.9601	0.7961	0.8704
	Avg/total	0.9808	0.9812	0.9803
BILSTM+CRF	O	0.9879	0.9979	0.9928
	B	0.9677	0.8483	0.9041
	I	0.9692	0.8622	0.9125
	Avg/total	0.9866	0.9867	0.9864

The performances of CRF, SVM and BILSTM+CRF are greatly improved under big data sets. Compared with small data sets, model can be more fully trained with sufficient corpus, so better results can be obtained. And BILSTM+CRF can get the best performance.

Table VIII Performance of SVM, CRF and BILSTM+CRF under big data sets for receptors

Method	Tag	precision	Recall	F1-score
SVM	O	0.9704	0.9804	0.9754
	B	0.7449	0.7302	0.7375
	I	0.6022	0.4409	0.5091
	Avg/total	0.9482	0.9515	0.9494
CRF	O	0.9464	0.9923	0.9688
	B	0.7927	0.4552	0.5783
	I	0.6432	0.1455	0.2373
	Avg/total	0.9396	0.9401	0.9272
BILSTM+CRF	O	0.9563	0.994	0.9748
	B	0.8872	0.5773	0.6994
	I	0.7887	0.2875	0.4182
	Avg/total	0.9470	0.952	0.9942

The performance of SVM, CRF and BILSTM+CRF in big data sets for receptors is similar to the result of subjects. And the BILSTM+CRF has the best performance.

Arguments with a relatively single expression method and strong regularity, such as time and quantity, are suitable for rule-based method. And it can use Bootstrapping to generate rules.

5.2.2 Experiments on public data set provided by Microsoft research Asia

We have tested F-LSTM, B-LSTM, BILSTM, BILSTM+CRF and SVM on public data set provided by Microsoft research Asia. BILSTM+CRF can get the best performance for all arguments, the result is similar to that of the big data set of real military data's result.

Then we test BILSTM+CRF's performance with different features.

Table IX Performance of different features combinations using BILSTM+CRF method

Feature	Precision	Recall	F1-score
vec	0.9826	0.9828	0.9827
vec+POS	0.9896	0.9897	0.9897
vec+POS+NER	0.9898	0.9899	0.9899
vec+POS+NER+parse	0.9912	0.9912	0.9912
All	**0.9920**	**0.9920**	**0.9920**

As we can see in table **IX**, all features added to WE can get the best performance, and the best performance is as shown in table **X**.

Table X Performance of all features added to WE using BILSTM+CRF method

Tag	Precision	Recall	F1-score
O	0.9958	0.9978	0.9968
B-PER	0.9818	0.9784	0.9801
I-PER	0.9830	0.9858	0.9844
B-LOC	0.9666	0.9628	0.9647
I-LOC	0.9536	0.9477	0.9506
B-ORG	0.9479	0.9048	0.9259
I-ORG	0.9580	0.9311	0.9444
Avg/total	0.9920	0.9920	0.9920

6 Conclusions

According to our experiments, we can conclude that different classes of arguments should adopt different features and methods. For the class of arguments with rich type of expressions, more information will limit performance and only word vectors with deep learning work best in small data sets. For the location and part-of-speech in the sentence are relatively fixed arguments, BILSTM+CRF with the features of vec, POS, NER, parse and location can get the best performance under small data sets. When corpus is enough, BILSTM+CRF has the best performance for both classes of subjects and receptors. Arguments with a relatively single expression methods and strong regularity are suitable for the rules-based methods and using Bootstrapping to generate rules.

Acknowledgements

This work is supported in part by the Beijing Municipal Commission of Science and Technology under Grant Z181100001018035; National Social Science Foundation of China under Grant 16ZDA055; National Natural Science Foundation of China under Grant 91546121; Engineering Research Center of Information Networks, Ministry of Education.

References

[1] Ji H,Grishman R.Refining Event Extraction through Cross Document Inference.ACL,2008:254-262.

[2] Doddington G R,Mitchell A,Przybocki M A,et al,The Automatic Content Extraction(ACE)Program-Tasks, Data, and Evaluation.LREC,2004.

[3] Wang Hui. Research on product attribute extraction technology based on Bootstrapping. Beijing University of Posts and Telecommunications, 2015.

[4] Chieu H L,Ng H T.A maximum entropy approach to information extraction from semi-structured and free text. AAAI/IAAI,2002,2002:786-791.

[5] Llorens H, Saquete E, Navarro-Colorado B. TimeML Events Recognition and Classification: Learning CRF Models with Semantic Roles[C] International Conference on Computational Linguistics. Association for Computational Linguistics, 2010:725-733.

[6] Zhao Yanyan, Qin Bing, Che Wanxiang and Liu Ting. Research on Chinese Event Extraction Technology. Chinese Journal of Information Science, 2008 (01): 3-8.

[7] Chen Yadong, Hong Yu, Wang Yibin, et al. Event cue detection based on multi-model and high confidence dictionary. Journal of Peking University: Natural Science, 2017, 53(3): 412-420.

[8] Zeng Y, Feng Y, Ma R, et al. Scale Up Event Extraction Learning via Automatic Training Data Generation. 2017.

[9] Naughton M, Stokes N, Carthy J.Sentence-level event classification in unstructured texts .Information retrieval, 2010, 13 (2):132-156.

[10] Nguyen MT, Nguyen T T. Extraction of disese events for a real-time monitoring system. Symposium on Information and Communication Technology, 2013:139-147.

[11] Chen Y, Xu L, Liu K, et al. Event Extraction via Dynamic Multi-Pooling Convolutional Neural Networks. The Meeting of the Association for Computational Linguistics. 2015.

[12] Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng and Jun Zhao. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 167–176, Beijing, China, July 26-31, 2015.