Feature weighting based Semi-supervised Clustering for Extracting Event Argument

ABSTRACT: *In the area of information extraction, event extraction is a challenging task which attracts much attention. This paper focuses on its subtask, event argument extraction, and presents a semi-supervised clustering method for extracting event argument. Considering the particular contributions of different features on clustering analysis, the method weights features by introducing the ReliefF algorithm. Constrained-KMeans, a semi-supervised clustering algorithm, is employed to group event arguments. Compared with normal Constrained-KMeans algorithm, feature weighting obviously improves the F-Measure of system performance. The comprehensive experimental results also demonstrate the outstanding performance of our method.*

Keywords: Feature weighting, Semi-supervised clustering, Event argument Extraction, ReliefF, Constrained-KMeans

Received: 6 July 2009, Revised 16 July 2009, Accepted 25 July 2009

1. Introduction

In the last few years, extensive research has been dedicated to information extraction with quite significant results. However, as a sub-domain of information extraction, the event extraction is still considered as one of the most challenging task. The Automatic Content Extraction (ACE) program, which is funded by National Institute of Standards and Technology (NIST), also defined event extraction as its fundamental task (ACE, 2005). The task mainly involves event recognition, event argument extraction and so on. In this paper, we focus on event argument extraction.

As is defined in ACE (Consortium, 2005), event arguments are often participants, which are the entities involved in an event. In addition to participants, there are two kinds of attributes, Event-Specific Attributes (such as the crime-arg and sentence-arg for justice events) and General Event Attributes (such as place-arg and time-arg), that can be associated with events as arguments. There are eight types and thirty three sub-types of events in ACE corpus. These event types include Life (Be-Born, Marry, Divorce, Injure, and Die), Movement (Transport), Transaction (Transfer-Ownership, Transfer-Money) and so on. Each sub-type of event has its own arguments.

In this paper, a feature weighting based semi-supervised clustering for extracting event argument is presented. The method employs the ReliefF (Robnik-Sikonja and Kononenko, 2003) to weight features and Constrained-KMeans (Basu et al., 2002) to group event arguments. By weighting features, different weights are assigned to different features according to their contribution on clustering analysis. In the Constrained-KMeans algorithm, the labeled objects in the corpus are taken as seed set to aid the clustering of the event arguments. Experimental results show that our method has good performance.

In the next section, we provide an overview of the related work on the event argument extraction. In section 3, we describe our approach for extracting event argument. In section 4, we give the experimental results and the analysis. In section 5, we conclude.

2. Related Work

Several previous studies have attempted to extract event argument. (Ahn, 2006) took the event argument extraction as a classification task. The KNN and ME classifier were employed to identify the event arguments in the task, and achieved the performance of 57.3% F-measure on the ACE English corpus. In (Ahn, 2006), the corpus is unbalanced and the data are sparse when multi-class classifier is trained. (Zhao et al., 2008) saw the general event argument in different event types as one class of event argument, and got the performance of 64.64% F-measure on the ACE Chinese corpus. In addition to the approach of machine learning, the pattern matching was also introduced to the event argument extraction. (Tan et al., 2008) employed multi-level patterns to extract the event arguments. Each level pattern utilized various languages information. The system achieved 48.86% F-measure on the ACE Chinese corpus.

3. Argument Identification

Previous work on the event argument extraction shows that the machine learning method is more effective and flexible than the pattern matching method. However, the disadvantage of sparse data can not be ignored when classifier is trained with the machine learning method. Even in ACE corpus, the largest event annotation corpus by far, 599 annotation documents for English events and 633 documents for Chinese events are still pretty insufficient. Event annotation is hard and time-consuming. However, raw corpus which involves events is easy to get from the Internet and other electronic resource. In this paper, we introduce the semi-supervised clustering algorithm to aid the clustering of event argument in un-annotated documents. By using semi-supervised clustering, the initial annotated event arguments which relevant categories are incomplete can be clustered as well as the initial annotated event arguments which present all the relevant categories.

In traditional semi-supervised clustering algorithm, a hypothesis is implied that all the features have the same contribution to the clustering while the similarity between two objects is computed. However, different features have different importance, ignoring their contributions on clustering may affect system performance. A possible approach is to compute weights for different features. Each feature has a weight associating with it.

3.1 Features for event argument extraction

To extract the event arguments, the following features are selected in our semi-supervised clustering experiments.

1. Lexical features: the current word w_i, the POS tag of w_i
2. Context features: the left word w_{i-1}, the POS tag of w_{i-1}, the right word w_{i+1}, the POS tag of w_{i+1}
3. Dependency features: the dependency relation of w_i, w_{i-1} and w_{i+1} in the sentence
4. Related trigger features: the trigger t, the event type of t, the POS tag of t, the dependency relation of t in the sentence
5. Position feature: the word w_i before or after the trigger t

3.2 Feature weighting

Given a data set $X = \{x_1, x_2, ..., x_n\}$, where n is the number of the objects, x_i is the i th object in X. Feature vector is $F = \{f_1, f_2, ..., f_m\}$, where f_j, is the j th feature. Function $value\ (f_j, x_i)$ gives the weight of x_i on feature f_j. Feature weight vector is $W = \{w_1, w_2, ..., w_m\}$, where $w_j \in \mathrm{N}$, w_j is a weight which assigned to feature f_j. For two objects x_a and x_b, their similarity Sim (x_a, x_b) is defined by:

$$Sim(x_a, x_b) = \frac{\sum_{j=1}^{m} w_j \, E(value(f_j, x_a), value(f_j, x_b))}{\sum_{j=1}^{m} w_j} \tag{1}$$

Where function

$$E(value\ (f_j, x_a), value\ (f_j, x_b)) = \begin{cases} 1, \ if\ value\ (f_j, x_a) = (f_j, x_b) \\ 0, \ otherwise \end{cases}$$

We normalize the similarity by dividing it with $\sum_{j=1}^{m} w_j$ to ensure $Sim\left(x_a, x_b\right) \in [0, 1]$.

As mentioned above, different features have different importance when the similarity between two objects is calculated. An appropriate approach is to compute weights for different features. ReliefF (Robnik-Sikonja and Kononenko, 2003), a well known feature selection algorithm, is employed in our method for computing weights. Suppose we randomly select an object x in data set X, ReliefF searches l nearest neighbors p_r from the same class, and also l nearest neighbors from each of the different classes. For each f in feature vector F, its weightis defined by:

$$w = w \sum_{r=1}^{l} diff\ (f, x, p_r) + \sum_{q_r \notin class(x)} \left[\frac{P(class\ (q_r))}{1 - P(class(x))} \sum_{r=1}^{l} diff\ (f, x, q_r) \right] \tag{2}$$

Where P ($class$ (x)) is the probability of class which x is contained, P ($class$ (x)) is calculated by dividing the number of total objects in $class$ (x) by the number of total objects in data set X. For discrete features, function $diff$ (f, x, p_r) is defined by:

$$diff\ (f, x, p_r) = \begin{cases} 0, & if\ value\ (f, x) = value\ (f, p_r) \\ 1, & otherwise \end{cases} \quad (3)$$

For each feature f, we calculated its weight w by using equation (2) to get weight vector W. The whole process is repeated for t times, where t isa user-defined parameter. In the first step of the repeating process, the w on the right side of the equal sign is initialized to 0. The calculated w on the left side will be assigned to the w on the right side in the next step until the repeating process is terminated.

Note that the ReliefF algorithm is often used in supervised learning algorithm, where the partitions of data set are already known. However, in un-supervised learning, the partitions are unknown. We utilize the KMeans algorithm to getpartitions of the data set X firstly, and then use the ReliefF algorithm to calculate the weight vector W.

3.3 Semi-supervised clustering

In many machine learning tasks, for example, event extraction, there is a large supply of unlabeled data, but insufficient labeled data since it is expensive to generate. Semi-supervised learning combines labeled data and unlabeled data during training to improve performance. Semi-supervised learning is applicable in both classification and clustering. In semi-supervised clustering, some labeled data is used alone with the unlabeled data to obtain a better clustering.

Constrained-KMeans (Basu et al., 2002) is a well known, typical semi-supervised clustering algorithm. In Constrained-KMeans, the labeled data, also called the seed set, is used to initialize the KMeans algorithm. Thus, rather than initializing KMeans with K random means, the mean of the l th cluster is initialized with the mean of the l th partition S_l of the seed set. Besides, in the sequence steps, the cluster memberships in the seed set are not re-computed, and the cluster labels of the seed data are kept unchanged, only the unlabeled data are re-estimated. (Basu et al., 2002) experiments on some data sets which the number of partition K is given. However, in the task of event arguments extraction, we do not know how many categories in the data set, and the number of partition K has a significant impact on system performance. Before analyzing this issue, we would like to give some useful definitions as follows.

Labeled Object: the object which is annotated and belongs to a specific category in the annotated corpus is called labeled object. In the task of event argument extraction, the labeled object means the word (or phrase) which is annotated to a specific type of event argument.

Labeled Cluster: after clustering analysis, the cluster which contains labeled objects is called the labeled cluster. Constrained-KMeans algorithm constrains that at most one type of labeled objects can be contained in a cluster. Therefore, we can ensure a one-to-one relationship between labeled object category and labeled cluster.

Unlabeled Object: the object which is un-annotated (because it does not belong to any existing categories in the annotated corpus) in the annotation corpus is called unlabeled object. Figure1 shows an example of unlabeled objects in ACE Annotation Guidelines for Events.

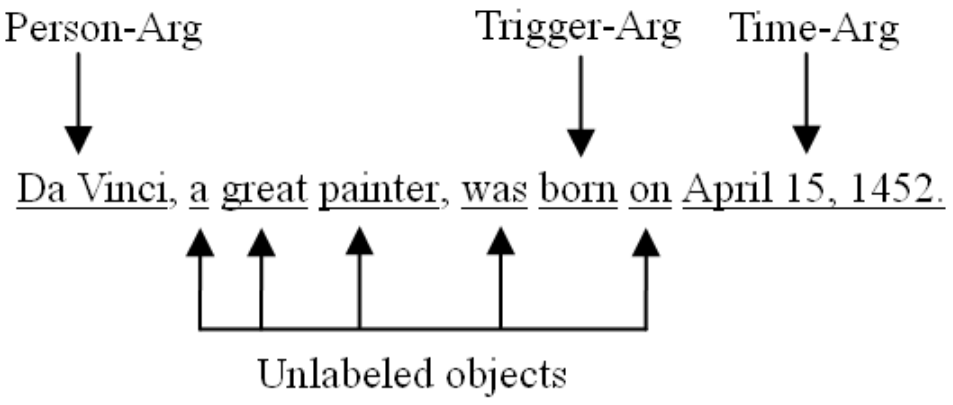

Figure 1. Unlabeled Object in Annotated Corpus

No-Label Object: all the objects in the raw corpus are called no-label objects.

No-Label Cluster: after clustering analysis, the cluster which does not contain any labeled objects is called no-label cluster.

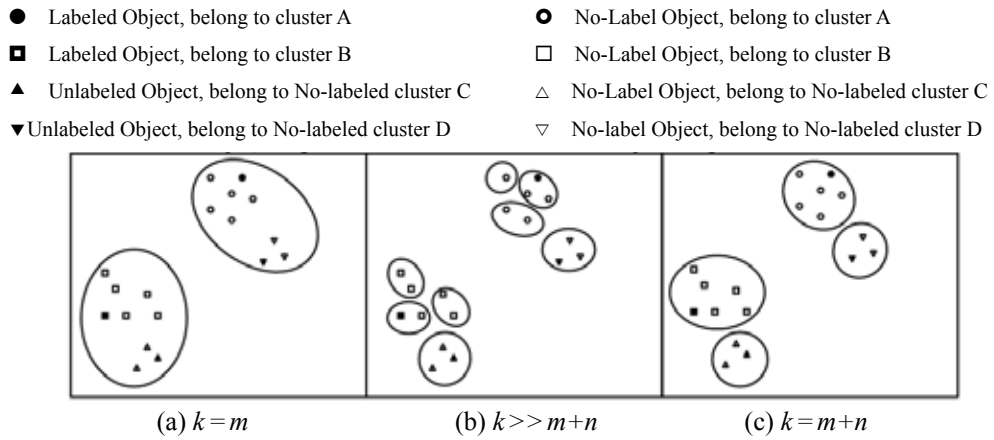

Figure 2. Impact on system performance with different

Given a corpus, in whichis the number of labeled object categories, n is the number of unlabeled object categories. If we let the partition number $k = m$, the unlabeled object will be clustered to a nearest labeled cluster. This will lead to poor precision on system performance. If we let the partition number $k >> m + n$, the no-label objects which should belong to a labeled cluster will be clustered to a no-label cluster with other no-label objects. This will lead to low recall on system performance. Thus, as is shown in figure2, the number of partition k has a significant impact on system performance. Theoretically, the optimal choice is to let $k = m + n$.

It's easy to know the value of m which is the number of the annotated object categories in an annotated corpus. Thus, determining how to find the optimal k is changed to how to find the optimal n. Though the value of n is unknown, if a corpus and a classification criteria are given, the value of n is fixed, that is to say the number of unlabeled object categories is fixed. We can get the approximation of n by experiments.

4. Experiments and Analyses

There are three steps in our experimental process: (1) calculating the feature weight vector; (2) experiment on the number of unlabeled object categories n in semi-supervised clustering; (3) comparing the system performance of our method with the normal Constrained-KMeans and KNN for event argument extraction.

We collected 800 documents from varies websites. Four types and eight sub-types of events about emergencies were involved in the collection. Each sub-type event has its own arguments. The information about the corpus is shown in table1.

Event type	Event sub-type	Number of args	Number of docs
Natural disaster	Typhoons	3	100
	Earthquake	3	100
Accidents	Traffic	5	100
	Explosion	3	100
Public sanitation	Food Poisoning	5	100
	Avian Influenza	4	100
Social security	Riot	3	100
	Air Raid	4	100

Table 1. Information of the corpus in the experiments

The information of word POS and dependency relation are obtained by using the DeParser system developed by IR-Lab in HIT[1]. We evaluated our system with precision, recall and F-Measure.

4.1 Results of feature weight vector

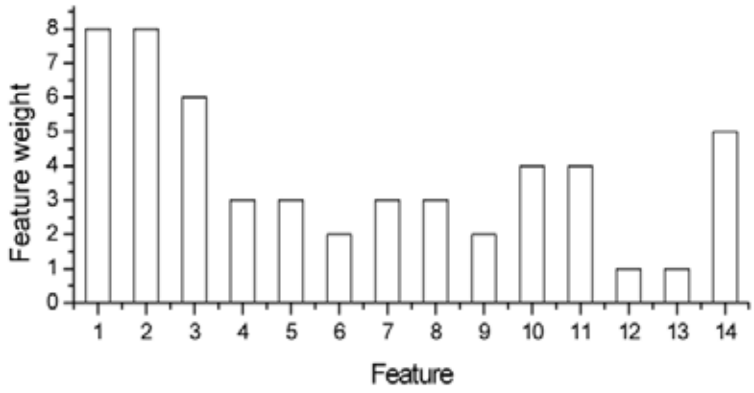

Figure 3. Results of the feature weight vector

In step1, we employ the ReliefF algorithm to compute feature weight vector. The results are shown in figure3, in which feature1 is the current word w_i; feature2 is the POS tag of w_i; feature3 is the dependency relation of w_i in the sentence.

The columns 4 to 6 are features of word w_{i-1}; 7 to 9 are features of w_{i+1}; 10 to 13 are features of trigger t; the feature14 is position relation between the current word w_i and the trigger t. From figure3, we can come to the conclusion that the information of current word is more important than its context in event argument extraction.

4.2 Approximate value of unlabeled object categories n

The results of feature weight vector are used in the step2. In this step, we study the effect of the number of unlabeled object categories n in semi-supervised clustering. The number of n is a fixed value in a given corpus, no matter how many seeds we assigned. So we simply experiment on 60% of the data set as seed set in this stage. The value of n in semi-supervised clustering is assigned varying from 20 to 60 in steps of 10.

Experimental results are shown in table2. From the table we can see that when $n = 20$, system achieves a highest recall, but the precision is the lowest. The reason for those result is that the actual value of n is larger than 20, assigning $n = 20$ leads to that some unlabeled objects are clustered to a nearest labeled cluster. When $n = 60$, system achieves the highest precision, but the recall is the lowest. The reason for this phenomenon is that the actual value of n is smaller than 60, assigning $n = 60$ leads to that some no-label objects, which should belong to a labeled cluster, are clustered to a no-label cluster. When $n = 40$, system achieves the highest F-Measure. From this step, we can assign an approximate value of $n = 40$ in our corpus.

categories(n)	Precision	Recall	F-Measure
20	0.392	0.791	0.524
30	0.547	0.723	0.623
40	**0.646**	**0.690**	**0.667**
50	0.723	0.532	0.613
60	0.785	0.458	0.578

Table 2. System performance on different number of categories, seed rate=60%

[1]http://ir.hit.edu.cn/

4.3 Comparison of system performance

Finally, we compare the system performance of our method with the normal Constrained-KMeans and KNN for event argument extraction. The same features mentioned in section 3.1 are selected in the three methods. The seed set, which is used to guide the clustering of Constrained-KMeans at various seed rate, is also taken as the training set in the KNN learner. In our method, the feature weight vector calculated by step1 is applied to the next Constrained-KMeans.

From figure4, it can be seen that with the decreasing of the seed rate, the performance of normal Constrained-KMeans and our method (Constrained-KMeans plus Feature weighting) is declined correspondingly. However, our method performs better than normal Constrained-KMeans at any seed rate. Compared with normal Constrained-KMeans algorithm, feature weighting obviously improves the system performance. Note that when seed rate declines to 0%, the Constrained-KMeans is degraded to traditional KMeans.

It also can be seen that our method performs better then KNN when seed rate varies from 90% to 0%. Particularly, our method is more effective when seed rate varies from 50% to 0%. When the labeled objects are insufficient, the classifier isn't trained well in the KNN, which leads to dramatically decreasing of system performance. By introducing feature weighting and semi-supervised clustering, our method can group data well with insufficient labeled objects.

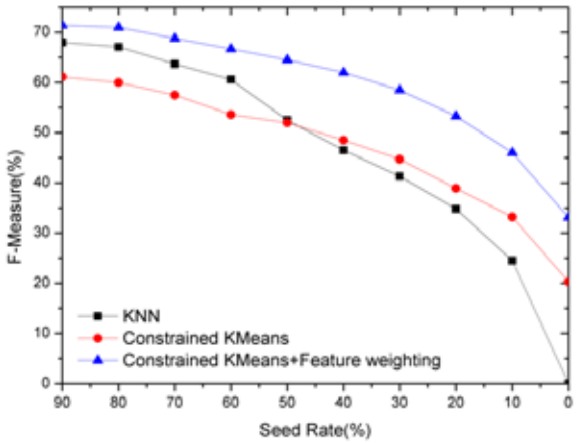

Figure 4. Comparison of system performance on different seed rate

5. Conclusion

In this paper, we presented a novel method for event argument extraction. The ReliefF algorithm, a feature selection algorithm often used in supervised learning, is introduced to our method. By using ReliefF algorithm, different weights are assigned to different features according to their importance of contribution on clustering. Then the calculated feature weight vector is applied to Constrained-KMeans algorithm for extracting event argument. Besides, we studied the impact of the number of unlabeled objects categories *n* on system performance. And we get an approximate value of *n* by experiment. Though the value of *n* is various in different dataset, our method is still appropriate. Compared with normal Constrained-KMeans algorithm, feature weighting obviously improves the F-Measure of identification. The comprehensive experimental results also demonstrate the outstanding performance of our method.

6. Acknowledgement

This work is supported by the projects of National Science Foundation of China (NSFC No.60575035, No.60975033), Shanghai Leading Academic Discipline Project (J50103) and Postgraduate Innovation Fund of Shanghai University (SHUCX091041).

References

[1] ACE (2005). The ACE 2005 (ACE05) evaluation plan. http://www.nist.gov/speech/tests/ace/ace05/doc/ace05-evalplan.v3.pdf.

[2] Ahn, D. (2006). The stages of event extraction. *In: Proceedings of the COLING- ACL 2006 Workshop on Annotating and Reasoning about Time and Events,* Sydney, July 23, Association for Computational Linguistics, p. 1–8.

[3] Basu, S., Banerjee, A., Mooney, R. (2002). *Semi-supervised Clustering by Seeding.* Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, p. 27-34.

[4] Consortium, L. D. (2005). ACE (Automatic Content Extraction) English Annotation Guidelines for Events. http://www.ldc.upenn.edu/Projects/ACE/.

[5] Robnik-Sikonja, M., Kononenko, I. (2003). *Theoretical and Empirical Analysis of ReliefF and RReliefF.* Machine Learning, 53 : 23-69.

[6] Tan, H., Zhao, T., Zheng, J. (2008). Identification of Chinese Event and Their Argument Roles. *IEEE 8th International Conference on Computer and Information Technology Workshops,* Sydney, Australia, IEEE computer society, p. 14-19.

[7] Zhao, Y., Qin, B., Che, W., Liu, T. (2008). Research on Chinese Event Extraction. *JOURNAL OF CHINESE INFORMATION PROCESSING,* 22 : 3-8.