A trainable method for extracting Chinese entity names and their relations

Yimin Zhang & Zhou Joe F

Intel China Research Center
Kerry Center 6F1
No. 1 Guanghua Road, Chaoyang District
Beijing 100020, P.R. China

Abstract

In this paper we propose a trainable method for extracting Chinese entity names and their relations. We view the entire problem as series of classification problems and employ memory-based learning (MBL) to resolve them. Preliminary results show that this method is efficient, flexible and promising to achieve better performance than other existing methods.

1. Introduction

Entity names and their relations form the main content of a document. By grasping the entity names and their relations from a document, we will be able to understand the document to some extent.

In the field of information extraction, much work has been done to automatically learn patterns [1] from training corpus in order to extract entity names and their relations from English documents. But for Chinese, researchers primarily use man-made rules and keyword sets to identify entity names [2]. Building rules is often complex, error-prone and time-consuming, and usually requires detailed knowledge of system internals. Some researchers also use statistical method [3], but the training needs lots of human annotated data and only local context information can be used. With this in the view, we have sought a more efficient and flexible trainable method to resolve this problem.

Section 2 first gives a general outline of the trainable method we have defined to extract Chinese entity names and their relations, then describes person name extraction, entity name classification and relation extraction in detail. Section 3 describes the preliminary experimental results of our method. Section 4 contains our remarks and discussion on possible extensions of the proposed work.

2. General Outline of the Method

We view the problem of Chinese entity names and their relations as a series of classification problems, such as person name boundary classification, entity name classification, noun phrase boundary classification and relation classification. If those classification problems are clarified, then we will be able to resolve the related extraction problem. For example, if we can correctly classify (or identify) the beginning or the ending boundaries of a person name appeared in a document, we will be able to extract the person name.

The process can be divided into two stages. The first stage is the learning process in which several classifiers are built from the training data. The second stage is the extracting process in which Chinese entity names and their relations are extracted using the classifiers learned. The learning algorithm used in the learning process is memory-based learning (MBL) [4]. MBL entails a classification based supervised learning approach. The approach has been named differently in a variety of contexts, such as similarity-based, example-based, analogical, case-based, instance-based, and lazy learning. We selected MBL as our learning algorithm because it suits well the domains with a large number of features

from heterogeneous sources and can remember exceptional and low frequency cases that are useful to extrapolate from [5]. In addition, we can customize the learner using different weighting functions according to linguistic bias.

The main steps for the learning process are:

Step 1: Prepare training data in which all noun phrases, entity names and their relations are manually annotated.

Step 2: Segmenting, tagging, and partial parsing the training data.

Step 3: Extract the training sets (instance base) from the parsed training data. Four training sets are extracted for different tasks with each related to Chinese person names, entity names, noun phrases, or relations between entity names in the training data. The main features used in an example can be either local context features, e.g. dependency relation feature, or global context features, e.g. the features of a word in the whole document, or surface linguistic features, e.g. character feature and word feature, or deep linguistic features like semantic feature.

Step 4: Use MBL algorithm to obtain IG-Tree [4] for the four training sets. IG-Tree is a compressed representation of the training set that can be processed quickly in classification process. In our case, the resulted IG-Trees are PersonName-IG-Tree, EntityName-IG-Tree, NP-IG-Tree, and Relation-IG-Tree.

The main steps for extracting process are:

Step 1: Segmenting, tagging and partial parsing the input Chinese documents.

Step 2: Identify Chinese people names using PersonName-IG-Tree.

Step 3: Identify Chinese organization names using the same method as described in [2].

Step 4: Identify other entity names (location, time, number) using the same method as described in [2].

Step 5: Identify Chinese noun phrases (NP chunking) using NP-IG-Tree.

Step 6: Use entity names and noun phrases extracted to perform partial parsing again to fix the parsing errors.

Step 7: Use EntityName-IG-tree to classify the noun phrases extracted. This step will identify entity names that are missed in the previous steps.

Step 8: Use Relation-IG-Tree to identify relations between the extracted entity names.

For a better understanding of the algorithm, we will describe in detail the person name extraction, the entity name classification, and the relation extraction in the next subsection. Please note that we are not going to discuss NP chunking further since it is beyond the main theme of this paper.

2.1 Person Name extraction

Chinese person names can be divided into two categories, local Chinese person names that consist of Chinese surnames and given names and transliterated person names that are sound translations of foreign names. The length of a local person name ranges from 2 to 6 characters, while the length of a transliterated person name is unrestricted.

After segmentation, person names are usually divided into several words. The task is to extract the word sequences that are person name components. With this in view, we convert the person name extraction problem to an equivalent classification problem, i.e. classifying word sequences existing in the results of segmentation into two classes, namely Person-Name and Not-Person-Name.

To classify a word sequence we need to use a number of features.

(1) Word features: the beginning word/tag of the sequence, the ending word/tag of the sequence.

(2) Local context features: the n-th (n<=3) word/tag before/after the sequence; the verb before/after the sequence.

(3) Context dependency features: the dependency relations of the word sequence and the dependency relations of the first

word before/after the sequence. The main dependency relations include verb-object (the relation between a verb and its noun object), subject-verb (the relation between a verb and its subject), subject-adj (the relation between an adjective and its subject), adv-verb (the relation between a verb and its adverbial modifier), adv-adj (the relation between an adjective and its adverbial modifier), modifier-head (the relation between a noun and its modifier. In Chinese, the modifier can be adjective, noun, verb or other phrases).

The word features and local context features can be directly extracted from the parsing results of the training data. The extraction of dependency features needs more explanation. We employ the collocation information obtained from a large corpus [6] to help the Chinese partial parser do the parsing. In most cases, dependency relations can be taken directly from the parsing results. But, there are some instances that the parser does not function well resulting in flat parsing trees. Under these circumstances, we resort to some simple heuristics such as linear order for dependency relations, thus making our method robust enough to extract most of the dependency relations.

To make the learning process more efficient, we use Boolean features in the training set, so every feature described above is translated into several Boolean features. For example, for the feature 1th-Next-Word (the first word after the word sequence), its value is the top 500 words (ordered by frequency) that can appear after a person name. We translate it into 500 features with every feature name like 1th-Next-Word-XX in which XX is one of the 500 words. The feature value 1 means that the XX appear next to the word sequence in the instance. The translated examples have about several thousands of Boolean features, which will be a big challenge for machine learning algorithms like C4.5 and CN2, but for MBL this is not a big problem. Furthermore, we use sparse array representation to make the storage requirement much lower.

For every word sequence in the training data that meets with one of the following three requirements, we extract that word sequence, including its class and all its features described above:

(1) Begin with a surname, plus 1 or 2 characters.

(2) Begin with two surnames, plus 1 or 2 characters.

(3) Begin with a character included in the first character set of transliterated person names (extracted from training data), plus several characters. The name may not surpass a normal word (that is included in a list of 5000 most frequently used Chinese words), because these normal words rarely occurred in a transliterated name.

For example, if in the training data, three words "W1 W2 W3" are annotated as a person name, then we will extract a Person-Name "W1 W2 W3" and Not-Person-Name "W1 W2".

After all the examples are extracted from the training data, they are fed to MBL Learner to get the PersonName-IG-Tree.

In the extracting process, we do the same as in the learning process to extract all examples, but the class of every example is unknown to us in advance. With the PersonName-IG-Tree, we can derive the class of every example, and then all word sequences classified as PersonName are extracted.

When a person name appears more than once in a document, we can rely on cache mechanism, similar to those described in [2], to solve ambiguous cases. For example, a person name "李文" appears more than once in a document. We first erroneously extract "李文亲" from a sentence "集团总裁李文亲率一个大型代表团造访中国", then correctly extract "李文" from another sentence "李文来华考察". Now "李文亲" and "李文" are both in the cache, the cache mechanism will be able to correct the first

error based on the heuristics that, if an extracted person name is a substring of another extracted person name and they do not appear in one sentence, then only the substring is the correct person name.

To better understand the process of extracting person names, we describe a few intuitive examples below.

(1) Sample 1

The sentence: "总经理李少华认为"

The segmentation result: "总经理 李 少 华 认为".

Here both "李少华" and "李少" are person name candidates for extraction. Because no dependency relations are found for the next word "华" and most training examples with this feature are classified as Not-Person-Name in training data, so "李少" is also classified as Not-Person-Name. Both the previous word "总经理" and the next word "认为" are positive evidences that make it certain that ""李少华" should be classified as a person name, therefore our algorithm correctly extracted it from this sentence.

(2) Sample 2

The sentence: "记者看到一张通知"

The segmentation result: "记者 看到 一 张 通知".

Our algorithm correctly classified "张通知" as not a person name. The reason is that in the training data all word sequences whose previous word is "一" is a not a person name.

These examples show that our method performs disambiguation well thanks to the MBL learner's capability of catching exceptions.

2.2 Entity Name Classification

The task of entity name classification is to classify the given noun phrases into several categories, such as organization name, product name, location, etc., as well as person names that are missed in the previous extraction.

In addition to the features used in person name extraction, more features are needed. Some of these features are equivalent to the features used in Crystal [8], such as subject-auxiliary-noun, e.g. the relation between "联想公司" and "公司" in the sentence "联想公司是一家著名的计算机公司". Some features are specific to Chinese. Semantic features are also included to make the learned classifier more powerful. The semantic features of a word can be taken from a widely used Chinese thesaurus [7] that classifies Chinese words into 12 broad categories, 94 middle categories, and 1428 small categories. There are about 70 thousands words in the thesaurus.

Unlike other inductive learning systems in the field of information extraction, such as Crystal, we use a general machine learning algorithm to do the learning. The most relevant earlier work is the experiment described in [8] using the machine learning algorithm C4.5. Though their experiment showed that the performance of C4.5 based method was comparable to Crystal, they abandoned this method due to the time complexity of C4.5 when dealing with large number of features. MBL is similar to C4.5 in that both are general machine learning algorithms. Their differences lie in that MBL is a lazy learning algorithm that keeps all training data in memory and only abstracts at classification time by extrapolating a class from the most similar items in memory, therefore, its time complexity is much lower than C4.5, especially when training data contains large number of features and examples. Actually, in Soderland's analysis [8], MBL's time complexity is only slightly greater than that of Crystal. Though the instance-based algorithm like MBL may require large memory, the advanced hardware technology available today can overcome this problem. A sparse vector representation will also lower the memory requirement. Taken all these into consideration, MBL is well suited for entity name extraction and relation extraction. Furthermore, the simple instance representation and weight function make

MBL-based method more flexible and extensible. Any useful features can be added to the system without any modification to the algorithm. In Crystal, however, adding more features may affect the correctness of weight function used in finding similar examples. Our method can employ global features, i.e. features beyond the sentence level. We treat a NP and all its occurrences (including its anaphorical references) in one text as one single example and all context words that are in some dependency relations to this NP as this example's features. Thus, we can resolve more complicated cases than Crystal. For example, if a NP is in subj-verb relation with verb "说", it can be a person name or an organization name. But, if we know all verbs that have subj-verb relations with this NP, then we will know the exact class this NP belongs to.

The steps for entity name classification are similar to the steps in person name extraction. Our method is quite impressive in that it can learn a lot of context features to classify the entity names, e.g. it correctly classifies "启明" in "启明的父亲" (Qiming's father) as a person name. Such a person name cannot be recognized in person name extraction because it does not begin with a surname or first character of transliterated person names.

2.3 Relation extraction

This task is to identify relation classes between entity names. Our current classes include employee-of, location-of, product-of, and no-relation. The relations we can extract are by no means restricted to this set. We can expand the set if training data are provided.

The features for this task include features used in Soderland's experiment [8]. These features are equivalent to the syntactic-lexical or syntactic-semantic constraints used in Crystal. The feature name begins with the name of the syntax position (SUBJ, OBJ, PP-OBJ etc.), followed by the name of the constraint and the actual term or class name. For example, "联想总裁" in the subject position would include the features:

SUBJ-Terms-联想
SUBJ-Terms-总裁
SUBJ-Mod-Terms-联想 // the terms in the modifier of the subject
SUBJ-Head-Terms-总裁
SUB-Classes-Employee // the semantic categories of the subject
SUB-Mod-Classes-Organization
SUB-Head-Classes-Organization

More features are introduced in our method, such as the linear order of entity names, the word(s) between the entity names, the relative position of the entity names (in one sentence or in neighboring sentences), etc. These features will make our method more robust than Crystal.

For every two related entity names in the training data, we identify a training example and extract it. After all the examples are extracted from the training data, they are fed to MBL Learner to get the Relation-IG-Tree.

In the extracting process, we do the same as in the learning process to extract all pairs of entity names. Then using the Relation-IG-Tree, we can derive the relation between every pair of entity names.

To better understand the process of relation extraction, we describe a couple of examples below.

(1) Sample 1

The input text: 浪潮集团作为国内著名的IT 硬件设备制造商，…

In the entity extraction, we have extracted "浪潮集团" as a company name and "IT硬件设备" as a product name. In the training data, some training examples have similar sentence patterns, e.g. "Company Name (作为/是) ...Product Name 制造商", and most of the time there are product-of relation between the two entity names. Based on this evidence, a product-of relation can be identified between "浪潮集团" and "IT硬件设备".

(2) Sample 2

The input text: **吴士宏再度成为媒体关注的焦点。不过，这次她是以TCL集团副总裁兼信息产业公司总经理的身份。**

In the entity extraction, we have extracted "吴士宏" as a person name and "TCL集团" as a company name. Now we want to test if these two entity names have an employee-of relation. As can be seen in the training data, if a person name and a company name appear in neighboring sentences, and no other person names and company names are found in between, they tend to have a employee-of relation. Based on this evidence, an employee-of relation can be identified between "吴士宏" and "TCL集团". Current systems, such as Crystal, would find it difficult to resolve because these two entity names appear in different sentences.

3. System Evaluation

To test our method we prepare a manually annotated corpus comprised of about 200 business news. All the entity names (about 500 person names and 300 organization names), noun phrases, and relations (i.e. employee-of, product-of, location-of) in the corpus were manually annotated. Ten pairs of training set and testing set were randomly selected from the corpus with each set equivalent to half size of the entire corpus. We ran our learning and extracting processes on all the data sets and calculated the mean recall and precision rates. The results are showed in Table. 1.

Table 1: Evaluation for extracting Chinese entity names and their relations

	Recall	Precision
Person Name	86.3%	83.2%
Organization Name	73.4%	89.3%
Employee-Of	75.6%	92.3%
Product-Of	56.2%	87.1%
Location-Of	67.2%	75.6%

As can been seen, our performance in person name and organization name extraction is comparable to other systems [2,3] considering the relatively small size of the training corpus. Based on our survey, our work on extracting entity relations is unprecedented for Chinese, therefore we are unable to establish a benchmark. But, the extraction of emloyee-of relation looks quite good. Detailed analysis reveals that our method can handle well some instances where co-reference resolution is needed because we introduced cross-sentence features. The method did poorly on product-of relation extraction due to the errors in noun phrases chunking. With a better NP chunking module, the performance can be improved.

4. Conclusion

In this paper we presented a trainable method for extracting Chinese entity names and their relations. The method provides a unified framework based on MBL. Our preliminary experiment demonstrates that this trainable method is efficient and flexible. Any linguistic features, either surface or deep, can be easily added into the system. Preliminary experiments have shown that our performance is comparable to or better than other existing trainable methods, such as HMM and Crystal. Our work, however, is still in its preliminary stage. More thorough evaluation is required using larger testing corpora. Some algorithmic extensions are also expected so as to improve the performance, including automatic feature selection, coreference resolution, etc.

Reference

[1] S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. CRYSTAL: Inducing a conceptual dictionary. In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, Montreal, Canada, August 1995.

[2] H.-H. Chen, Y.-W. Ding, S.-C. Tsai, G.-W. Bian, Description of the NTU System used for MET-2, Message Understanding Conference

Proceedings(MUC-7), Washington, DC. Available at http://www.muc.saic.com /proceedings/muc_7_proceedings/ntumet2.pdf. 1998.

[3] Shihong Yu, Shuanhu Bai and Paul Wu , Description of the Kent Ridge Digital Labs System Used for MUC-7, Message Understanding Conference Proceedings, Washington, DC. Available at http://www.muc.saic.com/proceedings/muc_7_proceedings/kent_ridge.pdf. 1998.

[4] Walter Daelemans, Jakub Zavrel, Ko van der Sloot, and Antal van den Bosch. TiMBL: Tilburg Memory Based Learner, version 3.0, Reference Guide Reference: ILK Technical Report 00-01. Available at http://ilk.kub.nl/~ilk/papers/ilk0001.ps.gz., 2000.

[5] Walter Daelemans, Antal van den Bosch, Jakub Zavrel, Jorn Veenstra, Sabine Buchholz, and Bertjan Busser. Rapid development of NLP modules with memory-based learning, In Proceedings of ELSNET in Wonderland, pp. 105-113. Utrecht: ELSNET, 1998.

[6] D. Lin. Extracting Collocations from Text Corpora. First Workshop on Computational Terminology, Montreal, Canada, August, 1998.

[7] Mei Jiaju,《Tong Yi Ci Ci Lin》(Chinese), Shanghai Dictionary Publishing Press, Shanghai, 1983.

[8] S. Soderland. "CRYSTAL: Learning Domain-specific Text Analysis Rules", Technical Report, Center for Intelligent Information Retrieval, University of Massachusetts, Available from http://www-nlp.cs.umass.edu/ciir-pubs/te-43.pdf, 1996.