A General Approach to Extracting Full Names and Abbreviations for Chinese Entities from the Web

Guang Jiang, Cao Cungen, Sui Yuefei, Han Lu, Shi Wang

A General Approach to Extracting Full Names and Abbreviations for Chinese Entities from the Web

Guang Jiang[1,2], Cao Cungen[1], Sui Yuefei[1], Han Lu[1,2], Shi Wang[1]

[1] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, No.6 Kexueyuan South Road, Zhongguancun, Beijing 100190, China
[2] Graduate University of Chinese Academy of Sciences, No.19 Yuquan Road, Shi Jing Shan Distinct, Beijing 100049, China

Abstract. Identifying Full names/abbreviations for entities is a challenging problem in many applications, e.g. question answering and information retrieval. In this paper, we propose a general extraction method of extracting full names/abbreviations from Chinese Web corpora. For a given entity, we construct forward and backward query items and commit them to a search engine (e.g. Google), and utilize search results to extract full names and abbreviations for the entity. To verify the results, filtering and marking methods are used to sort all the results. Experiments show that our method achieves precision of 84.7% for abbreviations, and 77.0% for full names.

1 Introduction

Named Entity Recognition (NER) is a basic task in text mining; it is significant for information extraction, machine translation etc. in nature language processing (NLP). In 1998, MUC (Message Understanding Conference) defined seven categories of named entity task belong to three subtasks: entity names (organizations, persons, locations), times (dates, times), and quantities (monetary values, percentages). Named Entity Recognition is a challenging topic in NLP because named entities are huge and increase as time goes on.

Among the seven categories above, organizations, persons and locations are three most import types, therefore identification of these three categories become hot points of research (Maynard et al., 2000; Luo et al., 2003; Wang et al., 2007). In view of full names and abbreviations, there are two kinds of entities: full name entity and abbreviation entity. Identification of full name and abbreviation for person names and locations are easier than which of organizations, because they embody more regular laws: person name begins with a family name, e.g. (Manager Zhang), while location names are finite and their abbreviations usually are fixed usually. E.g., (Jing) is an abbreviation of (Beijing). However, organizations are more difficult to identify because they are often long, loose and flexible or sometimes contain locations.

Identification of named entities is prepared for extracting relation of different entities, which is significant in several fields such as question answering (QA), multi-word expression (MWE) and information retrieval (IR). Automatic content extraction (ACE) [3] evaluation plan organized by the National institute of standards and technology (NIST) had defined seven types of entity relation: ART (artifact), GEN-AFF (Gen-affiliation), METONYMY, ORG-AFF (Org-affiliation), PART-WHOLE, PRE-SOC (person-social) and PHYS (physical). They also can be divided into much more subtypes.

2 Related Research

Many researchers had been focused on the identification of named entity relations (Li et al., 2007; Liang et al., 2006). A general route is to convert the problem into classification, which means relations of entities are categorized as several types and a classifier is constructed. Different machine learning algorithms are applied to solve this question (Che et al.,2005; Miller et al., 2000; Kambhalta et al., 2004). As to Chinese entity relation, (Che et al.,2005) attempted to extract relations of entities automatically using Winnow and SVM, with F-score 73.08% and 73.27% respectively. (Liu et al.,2007) used HowNet to acquire semantic knowledge, and combine both semantic sequence kernel function and KNN classifier to extract relation, the accuracy achieve about 88%. (Dong et al.,2007) divided entity relation into categories: embedding relations and non-embedding relations, and construct different syntactic features respectively. Their experiment proved the method can improve performance of Chinese entity relation extraction task.

In this paper, we investigate to extract Chinese full names and abbreviations for given entities. Entities discussed in this paper are not limited to named entities, more precisely, are lexical nominal concepts, which are discriminative and exist independently; they can be either physical (e.g. Peking University) or abstract (e.g. scientific technique). In the following, we denote an entity with a Chinese full name as full name entity, and that with an abbreviated Chinese name as abbreviation entity. Different from other foreign languages, Chinese words and Chinese characters are different, we will refer to word as Chinese word and Chinese character as character itself.

In our paper, we don't study document-specific abbreviation, which means the abbreviation is just used in a specific document or context. E.g. In sentence ..., the abbreviation (Wuhan subway) appears just as a concise anaphora in the document, but not a common fixed abbreviation.

[3] The nist ace evaluation website: http://www.nist.gov/speech/tests/ace/ace07/.

3 Extraction of full name entity and abbreviation entity from the Web

3.1 Comparison & analysis of full name entity and abbreviation entity

There are many common architectural features for a full name entity and its corresponding abbreviation entities. We sum up several points as follows:

1. Full name entity is consisted of several words, and its abbreviation entity is composed by one Chinese character from each word. E.g. (Women's Federation) and (Fulian).
2. Full name entity is consisted of several morphemes, delete some of them and constitute its abbreviation entity. E.g. (Tsinghua University) and (Tsinghua).
3. Full name entity is consisted of coordinating morphemes; first Chinese character of each morpheme and common morphemes constitute its abbreviation entity. E.g. (agriculture and industry) and (Gongnongye).
4. Full name entity is consisted of coordinating morphemes; numeral and the common morphemes constitute its abbreviation entity. E.g. (modernization of agriculture, industry, national defense and science and technology) and (the four modernizations).
5. There are no special laws for full name entities and abbreviation entities of countries, provinces, and transliterations etc. E.g. (Shanxi province) and (Jin); (Kentucky Fried Chicken) and KFC

In the following, we introduce our method of extracting full name entity as example, as extracting abbreviation entity is analogous. Fig 1 illustrates our overall framework.

3.2 Obtaining corpus

A big problem of extracting relation of entities is OOV (out of vocabulary), which is hard to identify. However, we found that full name entity and according abbreviation entity co-exist in a sentence in many cases: AB(B is short for A) and BA(A is full name of B). Inspired by this phenomenon, for a given entity, we can construct query items to search from the Web and obtain relevant corpus.

There are two kinds of query items: forward query item *Ent*, and backward query item **Ent*, in which *Ent* represents an entity. The query item contains double quotation marks to assure *Ent* and "/" could appear in the query result consecutively. The queries are committed to Google search engine[4] , we get only the summary of search results, not all the web pages which the result links to. In our experiment we collect top 200 search result summaries for each query item and delete the HTML tags.

[4] http://www.google.cn

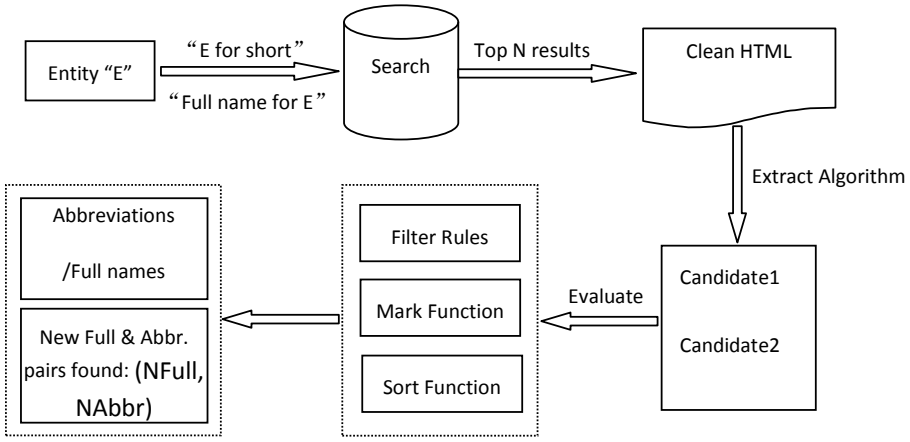

Fig. 1. Extract Abbreviations/Full names of Entity E from the Web

3.3 Relevant sentence analysis

For the corpora obtained above, we use two patterns for the two query items respectively to extract sentences which contain the query items

Pattern 1 *Ent*(common or bracket or . . .)(colon or bracket or is or . . .) We call the sentence matching pattern 1 as a relevant sentence, denoted as S, it can be represented as $Pre{+}Ent{+}Pun{+}{+}Cand{+}Pun$, in which Pre represent prefix words preceding entity Ent, Pun represent punctuation such as comma, and $Cand$ represent preliminary candidate.

Pattern 2 (common or bracket or . . .)(colon or bracket or is or . . .). . . *Ent*
Likely, relevant sentence S matching pattern 2 can be represented as $Cand{+}Pun{+}{+}Pre{+}Pun{+}Ent$. Different strategies are designed according to Pre(show in table 1). We list some examples in table 2.

Table 1. Different actions according to Pre

	Relevant Sentence	Next Action	
		Pre= NULL	$Pre\neq$ NULL
First pattern	$Pre{+}Ent{+}Pun{+}{+}Cand{+}Pun$	$Cand$ is a candidate. Extract Algorithm (**FAEA**)	(1) Extract Algorithm (**FAEA**) (2) New Full name and abbreviation may exist.
Second pattern	$Cand{+}Pun{+}{+}Pre{+}Pun{+}Ent$		

Table 2. Some examples of relevant sentence

Entity	Query Item	Relevant Sentence	$Pre=$ NULL	$Pre\neq$ NULL
(Peking)		""...	√	
(attached primary school)		...		√
(four modernizations)	*	1963129, ""	√	
(primary school)	*			√

3.4 Extracting candidates

Pattern 1 and Pattern 2 are analogous, so we describe our method aiming at pattern 1 in the following. We segment[5] relevant sentence $Pre{+}Ent\ {+}Pun\ {+}{+}\ Cand{+}\ Pun$ as $P_1P_2 \dots P_mEntPunC_1C_2 \dots C_nPun$. So our question converted into how to find two boundaries i, j, $1 \leq i \leq m$, $1 \leq j \leq n$, the full name entity candidate is $C_{j+1}C_{j+2} \dots C_n$, meanwhile, new possible full name & abbreviation entity pair found is ($P_{i+1}P_{i+2} \dots P_m$, $C_1C_2 \dots C_j$).

For a full name entity with segmentation $F = f_1f_2 \dots f_m$, $f_j(1 \leq j \leq m)$ is a word, and an abbreviation entity $A = a_1a_2 \dots a_n$, $a_i(1 \leq i \leq n)$ is a Chinese character. We define a similarity mark, which is not a strict measure, but used to decide the boundary of candidate from a sentence.

$$\text{SimMark}(A, F) = \sum_{\text{i=1}}^{\text{n}} \sum_{j=1}^{m} j * issub(a_i, f_j) \tag{1}$$

,in which $issub(a_i, f_j) = \left\{ \begin{array}{ll} 1 & if\ a_i\ appears\ in\ f_j \\ 0 & else \end{array} \right\}$

For example, we want to get the full names of entity (USTC), we extract a sentence (with segment and Pos-tagging) :

/n /n /j /w /d /v /a /a /n /w /v /w /u /n /w

(Central South University Patriotic Health Campaign Committee (full name of "Ai Wei Hui") issue notice about ...)

We found that $SimMark(\ ) \leq SimMark(,) \leq SimMark(,)$. Therefore, is more possible become a candidate than . The descriptions of algorithm are as follows.

[5] We use Chinese segment tool ICTCLAS. http://www.ictclas.org/

Input: Sentence S(with segment and Pos-tagging): $P_1P_2 \dots P_m EntPun$ $C_1C_2 \dots C_nPun$ Entity: Ent; Relation: ""(Full name)
Output: Full name Candidate of Ent: $EntFull$; New Full name-abbreviation pair ($NewFull$, $NewAbbr$);

1 Initialization: i=m, j=n;
2 If $P_1P_2 \dots P_i$ has at least a common character with $C_1C_2 \dots C_n$, then i=i-1; Else go to (4);
3 If P_i is an auxiliary or preposition word, then go to (4); Else i=i-1;
4 If P_i is a geographical entity name (discerning from Pos-tagging) or $SimMark(P_iP_{i+1} \dots P_mEnt, C_1C_2 \dots C_n) > SimMark(P_{i+1}P_{i+2} \dots P_mEnt, C_1C_2 \dots C_n)$, then i=i-1, go to (2);
5 If $C_jC_{j+1} \dots C_n$ has no common character with Ent , then j=j-1;
6 If $SimMark(Ent, C_jC_{j+1} \dots C_n) + SimMark(P_{i+1}P_{i+2} \dots P_m, C_1C_2 \dots C_{j-1}) > SimMark(Ent, C_{j+1}C_{j+2} \dots C_n) + SimMark(P_{i+1}P_{i+2} \dots P_m, C_1C_2 \dots C_j)$, then j=j-1; Else go to (7);
7 $EntFull$=$C_{j+1}C_{j+2} \dots C_n$, $NewFull$=$P_{i+1}P_{i+2} \dots P_m$, $NewAbbr$=$C_1C_2 \dots C_j$; Return.

Algorithm 1: Full name/ Abbreviation Extract Algorithm (**FAEA**)

3.5 Filtering & sorting candidates

After using algorithm **FAEA**, we may get several candidates; however, the Web is so open that some sentences may be irrelevant or even wrong. Therefore, it is necessary to validate and sort all the candidates.

We summarize some heuristic rules according to the commonalities between the abbreviation and full name of an entity, and they are used to filter out the candidates as follows.

Denote the set of candidates as $Candidates = \{Cand_1, Cand_2, \dots, Cand_n\}$, $Cand \in Candidates$, five filtering rules are defined as follows: (if one of the rules is satisfied, then $Cand$ is thought as an error, and thus be filtered out.)

1. $Cand$is a single Chinese character;
2. $len(Ent) \geq len(Cand)$, where $len(Ent)$ and $len(Cand)$ are the numbers of characters in $Cand$ and Ent respectively;
3. There are no common Chinese characters between $Cand$ and Ent;
4. $segnum(Cand) > segnum(Ent)$+3, where $segnum(Cand)$ and $segnum(Ent)$ are the numbers of words of $Cand$ and Ent with segmentation respectively;
5. $Cand$ contains some meaningless interrogatives words;

The first four rules are straightforward, and we explain the fifth one a little. We may obtain some interrogative sentences which interrogate for the full name or abbreviation of an entity, but which do not end with any question mark, such as (What is the full name of Tsinghua). In this case, $Cand$ = (what), which is actually an error. The fifth rule can identify such errors, and filter them out.

Attributed to the fact that an entity could have more than one full name or abbreviation entities, it's necessary to sort all the candidates using the statistical information from $Sents$, the candidate with rank one is most common full name

or abbreviation. We define a sort comparison function to sort all the candidates. Denote the set of relevant sentences as $Sents$, and define C_{i}>C_{j} , meaning C_i precedes C_j, iff

1. $SubstrFreq(C_{\mathrm{i}}) \geq SubstrFreq(C_{\mathrm{j}})$
2. $LD(C_{\mathrm{i}},Ent) \leq LD(C_{\mathrm{j}}, Ent)$, if $SubstrFreq(C_{\mathrm{i}}) = SubstrFreq(C_{\mathrm{j}})$
3. $SubseqFreq(C_{\mathrm{i}}) \geq SubseqFreq(C_{\mathrm{j}})$, if $SubstrFreq(C_{\mathrm{i}}) = SubstrFreq(C_{\mathrm{j}})$ and $LD(C_{\mathrm{i}},Ent) = LD(C_{\mathrm{j}}, Ent)$.

In which, $SubstrFreq(Cand)$is the number of sentences which $Cand$ appear in the sentences of $Sent$ as a substring; $LD(Cand, Ent)$ is the Levenshtein distance of $Cand$ and Ent; $SubseqFreq(Cand)$ is the number of sentences which $Cand$ appear in the sentences of $Sents$ as a subsequence.

4 Experiments and Discussion

We collect 300 pairs of full name entity and abbreviation entity as test data, in which about 80% are named entity, e.g. ((Peking University), (Beida)), 20). We compute abbreviation entities and full name entities respectively for each pair. The following tables illustrate our results. The precision of top k in table 3, table 4 and table 5 for full name is defined follows, the recall is either.

$$Precsion_{top_k} = \frac{Count_{top_k}}{Count_{Full}} \tag{2}$$

In which, $Count_{top_k}$ represents the number of all correct full names extracted in top k, while $Count_{Full}$ represents the number of all full names.

Table 3. Performance with only *Pattern* 1

	Precision		Recall(of Top 1)	F-measure	
Extract Abbr	Top 1	Top 3	71.7%	Top 1	Top 3
	87.1%	92.0%		78.7%	80.6%
Extract Full	70.5%	79.3%	58.4%	63.9%	67.3%

Table 3 shows the performance when only pattern 1 is used. We can found that performance of extracting abbreviations is higher; partly because the pattern is more efficient for abbreviations, moreover, more boundary information (such as parenthesis and quotation marks) could be supplied when extracting abbreviations than full names. Performance of top 3 is higher than which of top 1; which illustrates effect of our ranking strategy; also prove that some entities have more than one abbreviation or full name. The recall of full names is only 58.4%, we found the reason is that pattern 1 cannot obtain sufficient corpus.

Table 4 shows that after pattern 2 is used, the performance is improved in both precision and recall, especially the recall of full names. The results also

Table 4. Performance with *pattern* 1 & *pattern* 2

	Precision		Recall(of Top 1)	F-measure	
Extract Abbr	Top 1	Top 3	90.2%	Top 1	Top 3
	84.7%	93.6%		87.4%	91.9%
Extract Full	77.0%	82.7%	85.3%	80.9%	84.0%

Table 5. Performance with *pattern* 1 & *pattern* 2 for named entity

	Precision		Recall(of Top 1)	F-measure	
Extract Abbr	Top 1	Top 3	91.1%	Top 1	Top 3
	89.5%	95.3%		90.3%	93.2%
Extract Full	82.3%	87.8%	85.9%	84.1%	86.8%

Table 6. Some sort comparison function values of abbreviations extracted

Entity	Abbreviation Candidates	Substr Freq	LD	Subseq Freq	Rank	Result correct?
(Chinese Academy of Social Sciences)		17	4	36	One	Yes
		3	2	30	Tow	Yes
		3	6	3	Three	No
(Doctoral candidate)		7	2	34	One	Yes
		6	3	36	Two	No

Table 7. New full names and abbreviations identified

Full name	Abbreviation	Trigger Entity	Result correct?
(Peking University)		(attached primary school)	Yes
(China)		(mobile)	Yes
(Beijing's Second)		(court)	No
(China Association of Trade in Service)		(customer service)	No

confirm us that the patterns accord with the most common expressions when full name and abbreviation entities co-appear.

Table 5 shows the performance of our method for named entities, we can find that our method is also efficient. Many organization entities end with a suffix and we called it "suffix abbreviation", because it usually follows many different entities in sentences. E.g. (association),(university). We found that pattern 2 is very efficient for entities end with suffix abbreviation.

Table 6 shows some results with their sort compare function value; we can see that our three-level sort strategy is effective. For example, two abbreviation candidates of (Chinese Academy of Social Sciences) are (CASS) and (Siweiliangyuan); Although their SubstrFreq value is the same, we can sort them using Levenshtei Distance value correctly.

Table 7 illustrates the new pairs of full name entity and corresponding abbreviations found when we implement our algorithm for an entity (see the third column). In most cases, they are extracted when encountering with suffix abbreviation. Some pairs are partially correct and could be amended moreover. It also illustrates that our method can extend itself and get more pairs iteratively.

Furthermore, we summarize some difficulties of extracting full names or abbreviations as follows:

1. Abbreviation entities are usually OOV and difficult to segment. Therefore, the segments of abbreviation candidate supplied us are not reliable.
2. Obtain least but most useful corpora: we should have a tradeoff between more corpora and less Web search. We believe that the recall of extracting full name entities may be improved if we introduce more query items.
3. The prefix of a correct full name entity or abbreviation entity is difficult to identify when extracting candidates. In addition, the length and constitutes of prefix words are hard to be determined when we figure out the left boundary of the candidate.

5 Conclusions

In this paper, we aim at extracting full names and abbreviations for a given entity from the Web. We propose a method of combining both patterns and rules to solve the problem. In addition, new pairs of full name entity and abbreviation entity can be extracted simultaneously. Our experiment shows our method is efficient. However, there is still more future research to improve this work. For example, how to construct more query items to obtain more corpora, how to use named entity identification method to validate candidates.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant No.60496326, 60573063, and 60773059; the National High Technology Research and Development Program of China under Grant No. 2007AA01Z325; the Education Commission Program of Beijing under Grant No. KM201010009004.

References

1. Che, W.X., Liu, T., Li, S.: Automatic Entity Relation Extraction. Journal of Chinese Information Processing 19(2), 1-6 (2005)
2. Dong, J., Sun, L, Feng, Y.Y., et al.: Chinese Automatic Entity Relation Extraction. Journal of Chinese Information Processing 21(4), 80-85,91(2007)
3. Kambhatla, N.: Combining lexicalsyntactic and semantic features with Maximum Entropy models for extracting relations. In: Proceedings of 42th Annual Meeting of the Association for Computational Linguistics, pp.21-26 (2004)
4. Li, W.G., Liu, T., Li, S.: Automated Entity Relation Tuple Extraction Using Web Mining. Acta Electronica Sinica 35(11), 2111-2116 (2007)
5. Liang, H., Chen, J.X., Wu, P.B.: Information Extraction System Based on Event Frame. Journal of Chinese Information Processing 20(2), 40-46 (2006)
6. Liu, K.B., Li, F., Liu, L., et al.: Implementation of a Kernel-Based Chinese Relation Extraction System. Journal of Computer Research and Development 44(8), 1406-1411 (2007)
7. Luo, S.F., Sun, M.S.: Two-Character Chinese Word Extraction Based on Hybrid of Internal and Contextual Measures. In: Proceedings of the Second SIGHAN Workshop, on Chinese Language Processing ACL, pp. 24-30 (2003)
8. Maynard, D., Ananiadou, S.: Identifying Terms by Their Family and Friends. In: Proceeding of COLING, pp. 530-536(2000)
9. Miller, S., Fox, H., Ramshaw, L., Weischedel, R.: A Novel Use of Statistical Parsing to Extract Information from Text. In: Proceedings of 1st Meeting of the North American Chapter of the Association for Computational Linguistics(NAACL), pp.226-233 (2000)
10. Tian, G.G.: Research of Self-Supervised Knowledge Acquisition from Text based on Constrained Chinese Corpora (Doctor thesis). Institute of computing technology, Chinese Academy of Sciences. 2007
11. Wang, S., Cao, Y.N., Cao, X.Y, Cao, C.G.: Learning Concepts from Text Based on the Inner-Constructive Model. In: Second International Conference on Knowledge Science, Engineering and Management(KSEM), pp. 255-266 (2007)