# Constructing Chinese Abbreviation Dictionary: A Stacked Approach

*Longkai Zhang* *Sujian Li* *Houfeng Wang** *Ni Sun* *Xinfan Meng*
Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China
zhlongk@qq.com, lisujian@pku.edu.cn, wanghf@pku.edu.cn,
sunny.forwork@gmail.com, mxf@pku.edu.cn

ABSTRACT
Abbreviation is a common linguistic phenomenon with wide popularity and high rate of growth. Correctly linking full forms to their abbreviations will be helpful in many applications. For example, it can improve the recall of information retrieval systems. An intuition to solve this is to build an abbreviation dictionary in advance. This paper investigates an automatic abbreviation generation method, which uses a stacked approach for Chinese abbreviation generation. We tackle this problem in two stages. First we use a sequence labeling method to generate a list of candidate abbreviations. Then, we try to use search engine to incorporate web data to re-rank the candidates, and finally get the best candidate. We use a Chinese abbreviation corpus which contains 8015 abbreviation pairs to evaluate the performance. Experiments revealed that our method gave better performance than the baseline methods.

KEYWORDS: Chinese Abbreviation Generation, Abbreviation Mining.

*Corresponding author

*Proceedings of COLING 2012: Technical Papers*, pages 3055–3070,
COLING 2012, Mumbai, December 2012.

## 1 INTRODUCTION

Abbreviation is defined as a short description of the original long phrase. For example, "ACL" is the abbreviation for the full form "Association for Computational Linguistics". While abbreviation is a common linguistic phenomenon, it causes many problems like spelling variation (Nenadić et al., 2002).The different writing manners make it difficult to identify the terms conveying the same concept, which will hurt the performance of many applications, such as information retrieval (IR) systems.

In IR applications, one simple solution is to expand the original query by adding corresponding abbreviations to a search engine. For example, when using a search engine with an original query of "United States of America", a user will get more relevant results by expanding the query to include the abbreviation "USA." To achieve this we need to have an abbreviation dictionary, which is laborious to manually maintain because the number of abbreviations increases rapidly (Chang and Schutze, 2006). Therefore, it is helpful to automatically generate abbreviation from full forms. This leads to the idea of "abbreviation generation", i.e., finding the correct abbreviation for a full form.

The generation of abbreviations in Chinese differs from that for English. The reason is that Chinese itself lacks many commonly considered features in English abbreviation generation methods (Pakhomov, 2002; Yu et al., 2006; HaCohen-Kerner et al., 2008; Ao and Takagi, 2005). Detailed differences between English abbreviation generation and Chinese abbreviation features are listed in TABLE 1. Due to these differences, specific attention should be paid to Chinese abbreviation generation.

| Feature | English | Chinese |
|---|---|---|
| Word boundary | YES | NO |
| Case sensitivity | YES | NO |

Table 1: Comparison between Chinese and English abbreviation generation with regards to features.

Most of Chinese abbreviations are generated by selecting representative characters from the full forms[1]. For example, the abbreviation of "北京大学" (Peking University) is "北大" which is generated by selecting the first and third characters, see TABLE 2. This can be tackled from the sequence labeling point of view.

| Original | 北 | 京 | 大 | 学 |
|---|---|---|---|---|
| Keep/Skip | Keep | Skip | Keep | Skip |
| Result | 北 | | 大 | |

Table 2: The abbreviation "北大" of "北京大学" (Peking University)

Meanwhile, full forms and abbreviations show linguistic links like co-occurrence in large text materials. If we can find candidate abbreviations and rank them properly, the performance of abbreviation generation can be improved. Web pages can just serve as a large corpus to provide this information. While it is impractical to retrieve and analyze each web page individually, search engine provides an interface to this vast information. When querying a term in a search

[1]A small portion of Chinese abbreviations are not generated from the full form. For example, the abbreviation of "河北省"(He Bei Province) is "冀". However, we can use a look-up table to get this kind of abbreviations.

engine, titles and snippets of pages containing the query terms are returned, which provides a natural corpus for further analysis.

In this paper, we propose a stacked approach to automatically generate Chinese abbreviations. This method consists of a candidate generation phase and a ranking phase. First, we generate a list of candidates for the given full form using sequence labeling method. Then a supervised re-ranking method based on Support Vector Machine (SVM) using web data is applied to find the exact abbreviation.

We evaluate on a Chinese abbreviation corpus and compare it with previous methods. A pure sequence labeling approach by (Sun et al., 2009) and a state-of-art method to incorporate web data by (Jain et al., 2007) are chosen as baseline methods.

The contribution of this paper is that we integrate sequence labeling and web data to create a robust and automatic abbreviation generator. Experiments show that this combination gets better result than existing methods. Using this method we build a Chinese abbreviation dictionary, which later can be used in other NLP applications to help improve performance.

The paper is structured as follows. We first describe our approach. In section 2 we describe the sequence labeling procedure and in section 3 the re-ranking procedure. Experiments are described in section 4. In section 5 we give a detailed analysis of the results. In section 6 related works are introduced, and the paper is concluded in the last section.

## 2 Candidate Generation

### 2.1 Sequence Labeling

As mentioned in section 1, the generation of Chinese abbreviations can be formalized as a task of selecting characters from the full form, which can be solved by sequence labeling models. Previous works proved that Conditional Random Fields (CRFs) can outperform other sequence labeling models like MEMMs in abbreviation generation tasks (Sun et al., 2009; Tsuruoka et al., 2005). For this reason we choose CRFs model in the candidate generation stage.

A CRFs model is a type of discriminative probabilistic model most often used for the labeling or parsing of sequential data. Detailed definition of CRF model can be found in (Lafferty et al., 2001; McCallum, 2002; Pinto et al., 2003).

### 2.2 Labeling strategy

Considering both training efficiency and modeling ability, we use a labeling method which uses four tags, "BIEP". "B" stands for "Beginning character of skipped characters", "I" stands for "Internal character of skipped characters,"E" stands for "End character of skipped characters", and Label "P" means the current character to be preserved in abbreviation. An example is shown in TABLE 3.

### 2.3 Feature templates

The feature templates we use are as follows. See TABLE 4.

Templates 1 and 2 express uni-grams and bi-grams, which is widely used in abbreviation generation tasks. Template 3 is used to encode the ability of numbers in the generation of Chinese abbreviations. Templates 4 and 5 are designed to detect character duplication, because duplicated characters are often kept only once.

| "国家语言文字工作委员会" (National Linguistics Work Committee) The abbreviation is "国家语委" (the 1st, 2nd, 3rd, 9th characters of the full form) | |
|---|---|
| BIEP | 国/P 家/P 语/P 言/B 文/I 字/I 工/I 作/E 委/P 员/B 会/E |

Table 3: The abbreviation "国家语委" of " 国家语言文字工作会" (National Linguistics Work Committee)

| |
|---|
| 1. Uni-gram $X_i$ |
| 2. Bigrams $(X_i, X_{i+1})$ |
| 3. Whether $X_i$ is a number |
| 4. Whether character $i$ equals character $i+1$ |
| 5. Whether character $i$ equals character $i+2$ |

Table 4: Feature templates used in our algorithm.

## 3 Re-ranking

### 3.1 Re-rank with web data

Many abbreviations simply generated by the CRF model do not actually match the reference abbreviation. The reason is that as a sequence labeling model, CRF gives a most probable abbreviation character sequence by analyzing local information for each character. However, for Chinese abbreviations, local information alone is not adequate.

The full form and its abbreviation naturally co-occur in a large text corpus. This information contributes to the retrieval of an abbreviation given its full form. However, we cannot incorporate this information directly in traditional statistical learning models, because to get this information we first need a list of candidate abbreviations of the full form, which should be obtained in advance. We also observe that although the top-ranked output of the CRF model is not always correct, the true abbreviation very often appears in the top few outputs of the CRF. Therefore we choose to use the output of CRF model as the list of candidates. The remaining job is to find an effective method to re-rank the candidates using some additional information.

The additional information mentioned above can be obtained from search engines. Search engines index huge amount of web pages, providing an efficient interface to such vast information. The results returned by search engines typically contain the total number of related pages, title and snippet for each page. All the above text materials are useful for us to extract "implicit connections" between the full forms and abbreviations to re-rank the candidates generated in the previous phase.

An example of these "implicit connections" is shown in FIGURE 1. In this case we investigate on the search results of the full form "国家语言文字工作委员会" (National Linguistics Work Committee) and its abbreviation "国家语委". From FIGURE 1(a) we can find that the abbreviation "国家语委" appears in the title of the 3rd result when searching the full form. From FIGURE 1(b) we can also find that the full form "国家语言文字工作委员会" co-occurs with the abbreviation in the snippet of the 3rd result when searching the abbreviation. Furthermore, we see that the two queries share the same top-ranked result, which can be inferred from the same URL of the first search result. All of these evidences imply that "国家语委" seems to be the abbreviation for

"国家语言文字工作委员会". Note that the highlighted key words also indicate that the search


(a) Search result of the full form "国家语言文字工作委员会"

(b) Search result of the abbreviation "国家语委"

Figure 1: An example of search results. We can see clearly that the full form and the abbreviation do have implicit connections in search results.

engine itself does not know the correspondence of the two words. In FIGURE 1(a) we can see that in the result containing the abbreviation, the abbreviation itself is not highlighted as a keyword. Instead, it only matches the keyword "国家" (National). Therefore our method just learns to make use of the implicit connections, rather than exploits what the search engine has already learnt.

Besides search results, another appealing source of text corpus that we should mention is Chinese Wikipedia. Wikipedia seems to be more structured, however, we choose not to use Wikipedia in our context because many Chinese abbreviations like coordinate phrases are not collected in wiki-texts. Besides, new abbreviations spring out almost every day, while manually maintained Wikipedia is updated slowly. These shortcomings of Wikipedia make it less competitive than search engines.

For the re-ranking phase, we generate lists of candidates for the training data and label reference abbreviations as positive instances, and the incorrect candidates as negative instances. Then a SVM classifier is trained for its advantage in processing continuous values. The original SVM model itself does not calculate probability, while there are various ways to estimate the probability (Platt et al., 1999). What we use in our approach is the probability a candidate to be labeled as positive. We re-rank these candidates by these probabilities in decreasing order, and choose the first one as the final result.

## 3.2 Features for re-ranking

The results returned by search engines mainly contain the total number of related pages, title and snippet of each page. Search engines usually automatically highlight the key words in title and snippet by bolding (or red coloring) the keywords. We once considered using the highlighted keywords as counting criterion in our algorithm, but soon we found that this criterion has many deficiencies. Take "清华大学" (Tsinghua University) as an example, one of its false candidate is "清华大", which happens to be the first 3 characters of the full form. When

searching "清华大", many "清华大"s are highlighted, but they are all appears as part of the full form "清华大学". So it will be biased if we choose highlighting as our criterion. All things considered, we use the direct matching schema in our algorithm, instead of only considering the highlighted words given by search engines.

The following are the features we choose.

**Factor 1: how often the full form appears in the title when searching for a candidate**
We score this factor by taking the first 20 results of searching for the candidate abbreviation form, and counting the number of results for which the title contains the full form. The text containing the abbreviation usually also contains its full form. To avoid misjudge, if the candidate itself does not appear in the search results, its score will be set 0.

**Factor 2: how often the candidate form appears in the title when searching for itself**
We score this factor by searching for the candidate abbreviation, and counting the number of results whose the title contains the candidate. The popularity of the candidate form to some extent reflects how common it is in daily life. We find that misspellings may have impact on this factor. Therefore, we require the full form to appear in the title of all search results at least once, or the score will be set 0.

**Factor 3: how often the full form appears in snippet when searching for a candidate**
This factor considers the occurrence of the full form in search result snippets, which is similar to factor 1. The only difference here is that we consider snippets, instead of titles.

**Factor 4: how often the candidate form appears in snippet when searching itself**
Similar to factor 2, this factor considers the occurrence in search result snippets instead of titles, which serves as a validation for whether the candidate is a legal phrase.

**Factor 5 and 6: how often the candidate appears in title and snippet when searching its full form**
These factors are represented as factor 5 and 6, corresponding to title and snippet respectively. The two factors are complementary to factor 1 and 3, differing in whether one searches the candidate or the full form. These factors serve as verification for the candidate form in searching full form results, testing whether the candidate is a legal term.

**Factor 7: comparing similarity between searching candidate and full form**
We first use factor 7 to denote similarity between the titles of the first 20 results of searching a candidate and same amount of titles from searching its full form. For two titles, we say they are same only if they fully match with each other, which indeed is the case in search results.

**Factor 8: search results count**
This factor is scored by the total number of results returned by searching "full-form AND candidate". As far as we can see, more results when searching the full form and a candidate together indicate a stronger link between these two terms.

**Factor 9: the co-occurrence of a candidate and its full form**
This factor considers how often a candidate and its full form co-occur in results of searching "full-form candidate". The co-occurrence of the full form and a candidate will increase the probability for this candidate to be the true abbreviation.

**Factor 10: matching forward syntactic patterns**
We first define syntactic patterns such as "X简称Y" ("Y is short for X"). Then we score this factor by counting how many times the results of searching "full-form candidate" match these patterns.

The word "forward" means the full form appears ahead of the candidate.
Our pattern extraction algorithm is illustrated in TABLE 5. The GetSnippets function returns a list of snippets for the given joint query "full-form + candidate" for each pair (A, B) in S. For each snippet found by GetSnippets function, we replace the full form and the abbreviation with wildcards "X" and "Y". Then we use function GetNgrams to extract character n-grams for n = 2, 3, 4, 5, 6 and 7. The n-grams are guaranteed to contain exactly one X and one Y. We sort the n-grams by their frequency and select the top patterns. We then use these patterns to score candidates in the re-ranking phase. Some of the patterns we use are shown in TABLE 6

| Algorithm 1 : ExtractPatterms() |
|---|
| • Initialize: Let S be a "Full form"-"Abbreviation" set.<br>• Begin:<br>• For each full-abbreviation $pairs(A,B) \in S$<br>Do $D \leftarrow GetSnippets(A,B)$<br>• For each snippet $d \in D$<br>Do $N \leftarrow N \cup GetNgrams(A,B,d)$<br>• $Patterns \leftarrow SortByFreq(N)$<br>• Return Patterns |

Table 5: Algorithm for extract patterns.

| |
|---|
| • X（Y）<br>• X（简称Y）<br>• X(Y)<br>• X（以下简称Y）<br>• X简称Y， |

Table 6: Forward patterns used.

**Factor 11: matching backward syntactic patterns**
The term "backward", in contrast to the previous "forward", means that the abbreviation appears in front of the full form. The algorithm to extract patterns is the same as factor 10. Some of the patterns we use are shown in TABLE 7.

| |
|---|
| • Y X<br>• Y-X<br>• Y（X）<br>• Y和X是同义词<br>• Y是X的简称 |

Table 7: Backward patterns used.

## 4 Experiments

We use the abbreviation corpus provided by Institute of Computational Linguistics (ICL) of Peking University in out experiments. The corpus is homogeneous to the corpus used in (Sun et al., 2008, 2009). It contains 8,015 Chinese abbreviations. Various kinds of abbreviation pairs can be found in this corpus, including noun phrases, organization names and some other types. Some examples are presented in TABLE 8. The length distributions of full form and references are shown in FIGURE 2.

| Type | Full form | Abbreviation |
|---|---|---|
| Noun Phrase | 优秀稿件(Excellent articles) | 优稿 |
| Organization | 作家协会(Writers' Association) | 作协 |
| Coordinate phrase | 受伤死亡(Injuries and deaths) | 伤亡 |
| Proper noun | 传播媒介(Media) | 传媒 |

Table 8: Examples of the corpus (Noun Phrase, Organization, Coordinate Phrase, Proper Noun)

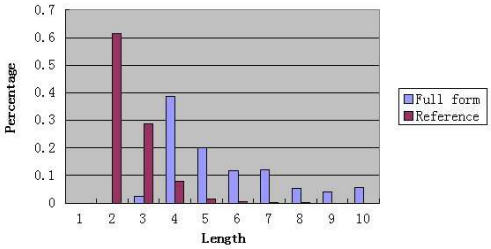


Figure 2: Length distribution of training set.

In some cases a long phrase may contain more than one abbreviation. For these cases, the corpus just keeps their most commonly used abbreviation one for each. Meanwhile to accurately get the results from the search engine that we need in our algorithm, we only keep the pairs with abbreviation containing more than 1 character, because the search results of a single Chinese character are usually ambiguous.

To improve the reliability of the experiment, we use 10 fold cross-validations. The evaluation metric used in our experiment is the top-k accuracy, which is also used by (Tsuruoka et al., 2005) and (Sun et al., 2009). The top-k accuracy measures what percentage of the reference abbreviations are found if we take the top N candidate abbreviations from all the results. In our experiment, top-10 candidates are considered in re-ranking phrase and the measurement used is top-1 accuracy because the final aim of the algorithm is to detect the exact abbreviation, rather than a list of candidates.

CRF++[2] and libsvm[3] , two open source tools, are used, with parameters are kept as default. The kernel function we use in our experiment is RBF kernel. All numeric values in SVM are scaled between 0 and 1. The generation of training examples for re-ranking considers the fact that a full form corresponds to a few candidate abbreviation forms, while only one of them is its reference. During the SVM process, we treat the reference as a positive instance, and treat the other false candidates as negative instances. Take the full form "北京大学" (Peking University) as an example. It corresponds to many candidate abbreviations like "北大", "京学". Only the reference "北大" is regarded as positive instance while the rest are negative. We then normalize the factors described in section 3 and use them together with the CRF score as features for each positive and negative instance in the re-ranking procedure.

The trained SVM classifier is then used in testing to give each candidate a label. For a given

[2]http://crfpp.sourceforge.net/
[3]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

candidate, what we are interested in is not the label, but the probability it will be labeled as positive. In libsvm the probability is calculated based on the vertical distance to the hyper-plane[4] . We follow this schema. This probability is then used as the re-ranking standard and we select the top-ranked candidate as the final result.

For search engine, we use the search engine Baidu[5] in the re-ranking phase, which is the biggest Chinese search engine.

## 5 Results and Discussion

### 5.1 Comparison of re-ranking

TABLE 9 shows the top-10 accuracy of the candidate generation stage, which is the first stage of our method. We can see the top-10 candidates include the reference abbreviation for most full forms. The top 10 candidates already cover 92% of the reference abbreviations using BIEP labels. In theory if we can find web data to re-rank the candidates, as high as 92% accuracy can be achieved compared to the original 58% accuracy.

| Top-K | 1 | 2 | 3 | 5 | 10 |
|---|---|---|---|---|---|
| Accuracy | 0.5812 | 0.7293 | 0.7975 | 0.8652 | 0.9240 |

Table 9: Top-10 Accuracy of CRF-BIEP.

We then use search results to re-rank the top-10 candidates. After re-ranking we select the top-ranked candidate as the final abbreviation of each instance. TABLE 10 shows the results. We can see that the accuracy of our method is 64.25%, which improved by +6% compared to using sequence labeling models alone.

| Method | Without re-rank | With re-rank |
|---|---|---|
| Top-1 accuracy | 0.5812 | 0.6425 |

Table 10: Results of Chinese abbreviation generation after re-ranking.

We also compare our method with previous methods. The first two are $CRF + GI$ and $DPLVM + GI$ in (Sun et al., 2009). We compare our approach with another web-based method used in (Jain et al., 2007), which is slightly different from ours. The work in (Jain et al., 2007) focuses on extracting full-abbreviation pairs, rather than generating abbreviations from full forms. However, we think it is meaningful to compare because in both cases the web data is used only to extract the useful information lie between the full form and abbreviation, which is independent of the problem settings. This method is denoted as "$CRF + AEPW$" used point wise mutual information (PMI), popularity of the abbreviation and the pagerank of the URLs in search results as features and integrate these features by multiplying them all. We also compare with another approach denoted as "$CRF + MUL$" which also multiplies all the features described in section 3. We add this comparison to see whether the difference is made by the feature set, not the re-ranking model itself.

TABLE 11 shows the results of the comparisons.

[4]http://www.csie.ntu.edu.tw/ cjlin/libsvm/

[5]http://www.baidu.com

[6]DPLVM is a model that needs multiple random initializations to get closer to the global optimal point. So we did not apply cross-validation for DPLVM+GI.

| Method | $CRF + AEPW$ | $CRF + MUL$ | $CRF + GI$ | $DPLVM + GI$[6] | Our method |
|---|---|---|---|---|---|
| Top-1 Accuracy | 0.5698 | 0.6039 | 0.5850 | 0.5990 | 0.6425 |

Table 11: Performance of different method.

While our method outperforms other methods, we surprisingly find that the CRF+AEPW slightly decreases performance compared to the pure CRF approach. The reason is that CRF+AEPW tries to extract information between well-formed full forms and also well-formed abbreviations. However, in the current Chinese abbreviation generation process, some ill-formed candidates may be generated, like include illegal terms and common phrases which are in fact substrings of the full form.

From CRF+MUL we can also find that simply multiplying the scores of each of the features does improve performance, however, the improvement is not as much as our approach. This indicates that our approach can better model the information extracted from search results than the simply treating the features equally. We measure what extent each feature contributes to the re-ranking process by adding one/two feature alone each time. For results see FIGURE3.

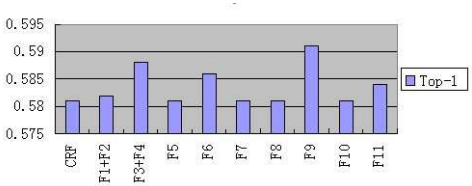


Figure 3: Contribution of each feature. The first column is the original sequence labeling score, which we use as a comparison.

FIGURE3 shows that feature 3+4, and 9 are the top contributing features. From feature 9 we can see that co-occurrence is indeed the most important factor. From feature 3 and 4 we can see that if the full form appears in the search results of a candidate, the candidate tends to be correct. This agrees to our intuition. If a candidate appears in the search results of the full form, it may happen to be a popular word as well as a substring of the full form. However, if the full form appears in the search results of a candidate, it means the full form does have strong link to the candidate.
We find that the re-ranking phase do play an important role in selecting the reference. Some reference abbreviations with low CRF scores can be reordered to the front after re-ranking. TABLE 12 shows the example of the organization name "阿拉伯国家联盟" (Arab League). The CRF score of its reference "阿盟" is low compared to other candidates, while after re-ranking, "阿盟" becomes the top-ranked candidate among all candidates.

TABLE 13, TABLE 14 and TABLE 15 show 3 more examples, which belong to different phrase types: noun phrase, coordinate phrase and proper noun. In all these cases, the references of the full form are picked out from the top 10 candidates. The results indicate that the re-ranking phrase can improve the performance of abbreviation generation.

| Candidate | CRF Score | Re-rank Score |
|---|---|---|
| 阿国联 | 0.427687 | 0.115977 |
| 阿联 | 0.182765 | 0.109433 |
| 国联 | 0.085203 | 0.0736369 |
| **阿盟(Reference)** | 0.053718 | 0.973178 |
| 阿国盟 | 0.043342 | 0.0225648 |
| 阿国联盟 | 0.032406 | 0.0361468 |
| 阿伯国联 | 0.021623 | 0.0213784 |
| 阿拉联 | 0.015541 | 0.0211315 |
| 阿联盟 | 0.013848 | 0.028979 |
| 阿拉伯国联 | 0.008748 | 0.0207346 |

Table 12: Generated abbreviations for Organization Name "阿拉伯国家联盟" (Arab League) and the correct re-ranking results.

| Candidate | CRF Score | Re-rank Score |
|---|---|---|
| 公共关 | 0.119895 | 0.0482089 |
| 公关系 | 0.099415 | 0.0365399 |
| 公共系 | 0.095923 | 0.0296014 |
| 公共 | 0.069083 | 0.036555 |
| 公系 | 0.058653 | 0.0250627 |
| 共关系 | 0.0545110 | 0.0604979 |
| **公关(Reference)** | 0.027417 | 0.96299 |
| 共关 | 0.015033 | 0.0450968 |
| 共系 | 0.012027 | 0.0284318 |
| 关系 | 0.001589 | 0.044727 |

Table 13: Generated abbreviations for Noun Phrase "公共关系" (Public Relation) and the correct re-ranking results.

| Candidate | CRF Score | Re-rank Score |
|---|---|---|
| 体医 | 0.522066 | 0.280765 |
| 体医疗 | 0.318698 | 0.0788524 |
| **体疗(Reference)** | 0.1497140 | 0.850495 |
| 体育医 | 0.003325 | 0.0244886 |
| 体育疗 | 0.001119 | 0.0303553 |
| 育医 | 0.001106 | 0.0234508 |
| 育医疗 | $6.75E-4$ | 0.0270482 |
| 育疗 | $3.72E-4$ | 0.0304597 |
| 医疗 | $2.28E-4$ | 0.0430604 |
| 体育 | $2.0E-5$ | 0.0394013 |

Table 14: Generated abbreviations for Coordinate Phrase "体育医疗" (Sports and Health) and the correct re-ranking results.

| Candidate | CRF Score | Re-rank Score |
|---|---|---|
| 物疗 | 0.344123 | 0.146159 |
| 物法 | 0.121928 | 0.0462008 |
| 物理疗 | 0.084886 | 0.0426191 |
| 物疗法 | 0.081885 | 0.0287058 |
| 物理法 | 0.073357 | 0.027162 |
| **理疗(Reference)** | 0.055539 | 0.906018 |
| 物理 | 0.050708 | 0.0491229 |
| 理法 | 0.047949 | 0.037015 |
| 理疗法 | 0.013221 | 0.054479 |
| 疗法 | 0.002413 | 0.0440872 |

Table 15: Generated abbreviations for Proper Noun "物理疗法" (Physiotherapy) and the correct re-ranking results.

## 5.2 Performance considering length

Long terms contain more characters, which is much easier to make mistakes during the sequence labeling phase. FIGURE 4 shows the top-1 accuracy respect to the term length using BIEP labeling method. The x-axis represents the length of the full form. The y-axis represents top-1 accuracy. We find that the search result based re-ranking method works especially well than pure CRF approach when the full form is long. By re-ranking using web data, additional information is incorporated. Therefore many of these errors can be eliminated. Meanwhile, if the reference

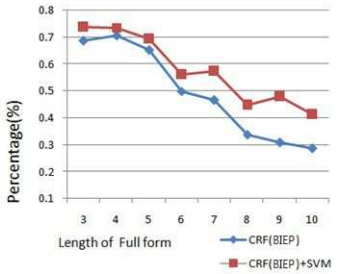


Figure 4: Accuracy grouped by length of full form.

itself is commonly used, the search results tend to contain more information of the relation between the candidate and the full form. With more information at hand, the re-ranking phase can make correct decisions with more confidence.

## 5.3 Error analysis

Though the accuracy is improved by +6% after we re-rank using search. There are still false candidates generated by the current method. We categorize the remaining errors as follows:

1. Candidates produced in the sequence labeling phase are only a portion of all possible combinations. Considering a full form with length 10, there are $2^{10} - 2 = 1022$ potential

candidates (the original term and empty string are omitted). Note that in our method we only take the top 10 candidates into consideration. If the references do not appear in the top 10 candidates, it is impossible for us to find the reference during the re-ranking phase. This kind of errors usually appears when the full form is very long. We find that this kind of error flourishes when the reference contains a Chinese word as its component. For example, "中央人民政府交通部" (Communication Department of the Central Government) is the full form for "中央交通部". The front "中央" (Central) and end "交通部" (Communication Department) are both Chinese words and appear continuously in the full form. Statistics show that this error makes up 13% of all errors.

2. Besides, character based candidate generation cannot use word level features, like word position. For example, if word "医院"(Hospital) is used in the middle of a full form, it is often abbreviated as "医", while in the end of a full form it will be abbreviated as "院". This is the shortcoming of al character based methods. However, we do not incorporate word information into our framework. As far as we investigate, previous works also seldom involve word information. The reason is that Chinese lacks natural word boundary, which cannot be segmented automatically with perfect accuracy. Current state-of-art Chinese word segmentation tools have at least 5% error rate, which can hurt consequent generation of abbreviations.
3. Search engines may provide biased information when handling location sensitive phrases. Take "香港民主同盟" (Democracy league of Hong Kong) as an example. Its correct abbreviation is "港同盟". Our method choose "民盟" as its abbreviation, which is the abbreviation of "中国民主同盟" (Democracy League of mainland China). Because the search engine we choose is Baidu.com, which is the most prevalent search engine in mainland China. Thus the number of search results related to "民盟" overwhelms that of "港同盟", with "民盟" 5200000 results compared with "港同盟" 13700 results. Besides when the web pages mentioning "民盟" (most of the pages are news pages), the "香港民主同盟" is always mentioned as well because there are homogeneous. Thus it is hard for the algorithm to exclude these interferences using localized search results. However, this kind of errors can be eliminated by using location-independent search engines.
4. Although some false candidates are not the standard reference, they are indeed used colloquially, only not as formally as the reference abbreviations. The reason for this phenomenon lies in the fact that the verification data we use is web search results. Web search results are sometimes colloquial, compared with official documents or other formal materials. Take "丁型病毒性肝炎"(Viral Hepatitis D) as an example, our method generates "丁肝", while the reference is "丁型肝炎". Both of these results are acceptable, while the reference is more formal.

Interestingly, we find that in this kind of errors, the "false" abbreviations are always shorter in length than the standard abbreviations, which is identical to the intuition that these abbreviations are more widely used orally.

## 6 Related work

Previous research on abbreviations mainly focuses on "abbreviation disambiguation", and machine learning approaches are commonly used (Park and Byrd, 2001; HaCohen-Kerner et al., 2008; Yu et al., 2006; Ao and Takagi, 2005). These ways of linking abbreviation pairs are effective, however, they cannot solve our problem directly because the full form is not always ambiguous. In many cases the full form is definite while we don’ t know the corresponding abbreviation.

To solve this problem, some approaches maintain a database of abbreviations and their corresponding "full form" pairs. The major problem of pure database-building approach is obvious. It is impossible to cover all abbreviations, and the building process is quit laborious. To find these pairs automatically, a powerful approach is to find the reference for a full form given the context, which is referred to as "abbreviation generation".

There is research on heuristic rules for generating abbreviations (Barrett and Grems, 1960; Bourne and Ford, 1961; Taghva and Gilbreth, 1999; Park and Byrd, 2001; Wren et al., 2002; HEARST, 2002). Most of them achieved high performance. However, hand-crafted rules are time consuming to create, and it is not easy to transfer the knowledge of rules from one language to another.

Recent studies of abbreviation generation have focused on the use of machine learning techniques. (Sun et al., 2008) proposed a supervised learning approach by using SVM model. (Tsuruoka et al., 2005; Sun et al., 2009) formalized the process of abbreviation generation as a sequence labeling problem. In (Tsuruoka et al., 2005) each character in the full form is associated with a binary value label y, which takes the value S (Skip) if the character is not in the abbreviation, and value P (Preserve) if the character is in the abbreviation. Then a MEMM model is used to model the generating process. (Sun et al., 2009) followed this schema but used DPLVM model to incorporate both local and global information, which yields better results.

While there are many statistical approaches, there are few approaches using Web as a corpus in machine learning approaches for generating abbreviations. Early examples like (Adar, 2004) proposed methods to detect such pairs from biomedical documents. Related work using web data includes (Liu et al., 2009; Jain et al., 2007). For example (Jain et al., 2007) used web search results as well as search logs to find and rank abbreviates full pairs, which show good result. But in fact search log data is only available in a search engine backend. In contrast, ordinary approach does not have access to search engine internals. Besides, they all use web data to expand the abbreviations to their full form, which is the opposite process of ours.

## Conclusion and future work

To build an abbreviation dictionary, we used a stacked method to generate abbreviations from the full forms. We used sequence labeling method with BIEP labels to generate candidates for each full form, and used a SVM classifier which utilizes search results to re-rank the candidates to generate the final result.

The results are promising and outperformed the baseline methods. The accuracy can still be improved. Potential future works may include using semi-supervised methods to incorporate unlabeled data, or use more powerful methods to extract the characters of abbreviations in web data.

## Acknowledgments

This work was partially supported by National High Technology Research and Development Program of China (863 Program) (No. 2012AA011101), National Natural Science Foundation of China (No.91024009, No.60973053), and the Specialized Research Fund for the Doctoral Program of Higher Education of China (Grant No. 20090001110047 )

## References


Adar, E. (2004). Sarad: A simple and robust abbreviation dictionary. *Bioinformatics*, 20(4):527–533.

Ao, H. and Takagi, T. (2005). Alice: an algorithm to extract abbreviations from medline. *Journal of the American Medical Informatics Association*, 12(5):576–586.

Barrett, J. and Grems, M. (1960). Abbreviating words systematically. *Communications of the ACM*, 3(5):323–324.

Bourne, C. and Ford, D. (1961). A study of methods for systematically abbreviating english words and names. *Journal of the ACM (JACM)*, 8(4):538–552.

Chang, J. and Schutze, H. (2006). Abbreviations in biomedical text.

HaCohen-Kerner, Y., Kass, A., and Peretz, A. (2008). Combined one sense disambiguation of abbreviations. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 61–64. Association for Computational Linguistics.

HEARST, A. (2002). A simple algorithm for identifying abbreviation definitions in biomedical text. In *Pacific Symposium on Biocomputing 2003: Kauai, Hawaii, 3-7 January 2003*, page 451. World Scientific Pub Co Inc.

Jain, A., Cucerzan, S., and Azzam, S. (2007). Acronym-expansion recognition and ranking on the web. In *Information Reuse and Integration, 2007. IRI 2007. IEEE International Conference on*, pages 209–214. IEEE.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Liu, H., Chen, Y., and Liu, L. (2009). Automatic expansion of chinese abbreviations by web mining. *Artificial Intelligence and Computational Intelligence*, pages 408–416.

McCallum, A. (2002). Efficiently inducing features of conditional random fields. In *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, pages 403–410. Morgan Kaufmann Publishers Inc.

Nenadić, G., Spasić, I., and Ananiadou, S. (2002). Automatic acronym acquisition and term variation management within domain-specific texts. In *Third International Conference on Language Resources and Evaluation (LREC2002)*, pages 2155–2162.

Pakhomov, S. (2002). Semi-supervised maximum entropy based approach to acronym and abbreviation normalization in medical texts. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 160–167. Association for Computational Linguistics.

Park, Y. and Byrd, R. (2001). Hybrid text mining for finding abbreviations and their definitions. In *Proceedings of the 2001 conference on empirical methods in natural language processing*, pages 126–133.

Pinto, D., McCallum, A., Wei, X., and Croft, W. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 235–242. ACM.

Platt, J. et al. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74.

Sun, X., Okazaki, N., and Tsujii, J. (2009). Robust approach to abbreviating terms: A discriminative latent variable model with global information. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 905–913. Association for Computational Linguistics.

Sun, X., Wang, H., and Wang, B. (2008). Predicting chinese abbreviations from definitions: An empirical learning approach using support vector regression. *Journal of Computer Science and Technology*, 23(4):602–611.

Taghva, K. and Gilbreth, J. (1999). Recognizing acronyms and their definitions. *International Journal on Document Analysis and Recognition*, 1(4):191–198.

Tsuruoka, Y., Ananiadou, S., and Tsujii, J. (2005). A machine learning approach to acronym generation. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases: Mining Biological Semantics*, pages 25–31. Association for Computational Linguistics.

Wren, J., Garner, H., et al. (2002). Heuristics for identification of acronym-definition patterns within text: towards an automated construction of comprehensive acronym-definition dictionaries. *Methods of information in medicine*, 41(5):426–434.

Yu, H., Kim, W., Hatzivassiloglou, V., and Wilbur, J. (2006). A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations. *ACM Transactions on Information Systems (TOIS)*, 24(3):380–404.