Extracting Pronunciation-translated Names from Chinese Texts using Bootstrapping Approach

Jing Xiao
School of Computing,
National University of Singapore
xiaojing@comp.nus.edu.sg

Jimin Liu
School of Computing,
National University of Singapore
liujm@comp.nus.edu.sg

Tat-Seng Chua
School of Computing,
National University of Singapore
chuats@comp.nus.edu.sg

Abstract

Pronunciation-translated names (P-Names) bring more ambiguities to Chinese word segmentation and generic named entity recognition. As there are few annotated resources that can be used to develop a good P-Name extraction system, this paper presents a bootstrapping algorithm, called PN-Finder, to tackle this problem. Starting from a small set of P-Name characters and context cue-words, the algorithm iteratively locates more P-Names from the Internet. The algorithm uses a combination of P-Name and context word probabilities to identify new P-Names. Experiments show that our PN-Finder is able to locate a large number of P-Names (over 100,000) from the Internet with a high recognition accuracy of over 85%. Further tests on the MET-2 test set show that our PN-Finder can achieve a performance of over 90% in F_1 value in locating P-Names. The results demonstrate that our PN-Finder is effective.

1 Introduction

Pronunciation-translated names (P-Names) are those foreign names that are translated to Chinese characters according to their pronunciations. A P-Name sometimes forms part of but not a complete named entity. For instance, in the place name "贝克利大学" (Berkeley University), only the term "贝克利" (Berkeley) is a P-Name, while "大学" (University) is not since it is translated semantically.

The ability to recognize P-Names helps to reduce ambiguities in word segmentation and improve the performance of Chinese information retrieval since many unknown words are P-Names, especially for international Chinese news. Unlike English, there is no blank between words in Chinese, in which a word is a linguistic token consisting of one or more characters. In addition, the same characters may appear in multiple context with different meanings (Chua and Liu, 2002). The presence of P-Names brings more ambiguities to Chinese word segmentation since every character in a P-Name can be used as a common character.

Intuitively, we can extract the P-Names based on the distinctive sequence of characters that they are used as compared to common words. In addition, we can use local context around the P-Names to confirm and classify them into person or part of location and organization names. One way to perform these tasks effectively is to rely on statistics derived from a large corpus in which the P-Names are annotated. While some annotated corpuses with general named entities are available such as the PKUC (Yu, 1999) and MET-2 (Chinchor, 2001), there is no such annotated corpus for P-Names. While annotated data is difficult to obtain, un-annotated data is readily available and plentiful, especially on the Internet. To take advantage of that, we need to tackle two major problems. The first is how to gather sufficient distinct P-Names from the Internet, and the second is how to use the available resources to derive reliable statistical information to characterize the P-Names.

The problem of gathering sufficient reliable information from a small initial set of seed resources has been tackled in bootstrapping research for information extraction (Agichtein and Gravano, 2000; Brin, 1998; Collins and Singer, 1999; Mihalcea and Moldovan, 2001; Riloff and Jones, 1999). Bootstrapping approach aims to perform unsupervised text processing to extract information from open resources such as the Internet using minimum manual labor. Given the lack of annotated training samples for P-Name extraction, this paper introduces a bootstrapping algorithm, called PN-Finder. It starts from a small set of seed samples, and iteratively locates, extracts and classifies the new and more P-Names. It works in conjunction

with a general Chinese named entity recognizer (Chua and Liu, 2002) to extract general named entities.

In the remaining parts of this paper, we describe the details of PN-Finder in Section 2 and its application in locating P-Names from new documents in Section 3. Section 4 presents the experimental results using the MET-2 test corpus. Section 5 contains our conclusion and outline for future work.

2 Bootstrapping Algorithm for Locating P-Names

Currently, there is no standard corpus that annotates all P-Names. Since annotating thousands of P-Names is more difficult than collecting thousands of P-Names from the Internet, we recur to using the Internet search engine to collect a large set of P-Names. Figure 1 illustrates our main components in bootstrapping process.

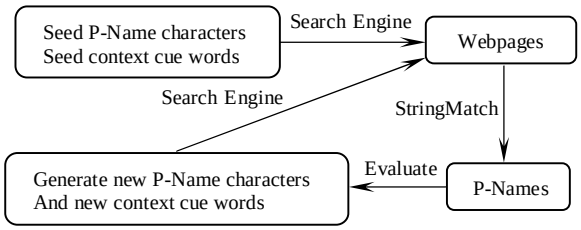

Figure 1: Main components of the bootstrapping

The inputs to the PN-Finder are:

a) A seed P-Name character set $\underline{C_s}^{(0)}$ consisting of 5 characters {"阿", "尔", "巴", "斯", "基"}.
b) A set of seed context cue words $\underline{CW}^{(0)}$ consisting of 60 context words, such as {"NULL", "记者", "说", "据", "在", "到"}. These are typical context words found around person, location and organization names in PKUC[1] (the PoS Corpus of Peking University), which contains one month of news from the People Daily.
c) A set of P-Name candidates $\underline{P}^{(0)}$, which is null at the beginning.
d) A common word dictionary extracted from PKUC by removing proper nouns, numbers and non-Chinese symbols. It contains about 37,000 words.

From the initial seeds, we perform the followings:

a) We use every two characters in $\underline{C_s}^{(i-1)}$ as query to retrieve relevant web pages from the Internet using a commercial search engine. We then extract possible P-Names from the returned web pages to update $\underline{P}^{(i)}$.
b) We find a most probable new P-Name character. Update $\underline{C_s}^{(i-1)}$ to $\underline{C_s}^{(i)}$ by adding the new character.
c) We bootstrap new context words around the new P-Names found to derive $\underline{CW}^{(i)}$. We then perform the lexical chaining to generalize these context words to generate semantic classes.
d) We repeat the process from step (a) until any of the following conditions is satisfied: (i) when no new P-Name is found; (ii) when the desired number of iterations is reached; or (iii) when the number of P-Names found exceeds a desired number.

The following subsections discuss the details of the bootstrapping process.

2.1 Querying and Extracting the P-Names from the Web

The first step of the algorithm is to derive good queries from the character set $\underline{C_s}^{(m-1)}$ to search the Internet to obtain new web pages. If we use all single characters from $\underline{C_s}^{(m-1)}$ to perform the search, we are likely to get too many pages containing irrelevant information. As a compromise, we use every two characters c_ic_j in $\underline{C_s}^{(m-1)}$ (except those combinations that have been used in the previous iterations) to search the Internet using Google (by using its language tool[2]). We consider only up to 300 entries returned by Google. We divide the content of the web pages into text segments by using the non-alphanumeric characters as delimiters. We extract those text segments that contain the search characters c_i, c_j or both and store them in $\underline{R}^{(m)}$. For example, from the web page given in Figure 2, the text segments extracted include: strings "基斯洛夫斯基" and "基斯洛夫斯基的影片被认为" from the first entry; and strings "塔科夫斯基", "你是否知道当塔科夫斯基站立在街头巷尾的时候" and "请记住塔科夫斯基吧" from the second entry.

Given $\underline{R}^{(m)}$, we next extract the possible P-Names. Firstly, we segment those entries in $\underline{R}^{(m)}$ by performing the longest forward match using the common word dictionary. We then remove all

[1] http://icl.pku.edu.cn/Introduction/corpustagging.htm

[2] http://www.google.com/intl/zh-CN/

non-Chinese letters and common words containing more than one character. From the remaining string segments in $\underline{R}^{(m)}$, we locate all sub-strings with at

Figure 2: A web page returned by Google

least 2-character in length and contain the query terms c_i, c_j or c_ic_j. We extract those sub-strings that appear at least τ_n (we use 3) times as P-Name candidates by string matching. We store the new P-Name candidates found in $\underline{P}^{(m-1)}$ to obtain $\underline{P}^{(m)}$.

For example, if we use “斯, 基” obtained from $\underline{C}_s^{(0)}$ as query to Google, among the returned entries, we will have:

“巴尔舍夫斯基(访问)(人民)(日报)...”
“(美国)(贸易)(代表)巴尔舍夫斯基说”
“就中美(WTO)(协议)(细节)巴尔舍夫斯基答(记者)问...”
“巴尔舍夫斯基说(中国)入世(谈判)...”
“巴尔舍夫斯基走出(使馆)...”

Here the bracketed words are common words or English letters and they are removed from string matching. The sub-string “巴尔舍夫斯基” appears 5 times and it is matched as a possible P-Name.

2.2 Deriving New P-Name Characters

Given the set of P-Name candidates in $\underline{P}^{(m)}$, we next use both the context words and corpus statistics to confirm the P-Name and extract new P-Name characters.

2.2.1 Classifying P-Names

From observation, context information is useful to comfirm a P-Name and determine its left and right boundary. Thus we use one-word context to confirm and classify P-Name candidates into person names or part of location or organization names. For each context word w_c in $\underline{CW}^{(m)}$, we first compute its probability vector of occurrences, $PV(w_c)$, around person, location and organization names in PKUC as follows:

$$PV(w_c) =< c_{-p}, c_{+p}, c_{-l}, c_{+l}, c_{-o}, c_{+o} > \quad (1)$$

where:
$$c_{+x}(w_c) = \frac{n_{+x}}{n_{+p} + n_{+l} + n_{+o}} \quad (1.1)$$

$$c_{-x}(w_c) = \frac{n_{-x}}{n_{-p} + n_{-l} + n_{-o}} \quad (1.2)$$

Here x∈ {p, l, o}, and n_{-p} (or n_{+p}), n_{-l} (or n_{+l}), n_{-o} (or n_{+o}) respectively give the number of times w_c appears at the left (or right) boundary of person (p), location (l) and organization (o) names in PKUC. c_{-x} (or c_{+x}) gives the probability that the P-Name is of type x, if this cue-word is at the left (or right) boundary of the P-Name.

Given a P-Name candidate $p_k^{(m)}$ in $\underline{P}^{(m)}$, we extract the set of its left and right context words as $\underline{W}_c^{l}$ and $\underline{W}_c^{r}$. We then derive the average probability vectors of $\underline{W}_c^{l}$ = <c^{l}_{-p}, c^{l}_{+p}, c^{l}_{-l}, c^{l}_{+l}, c^{l}_{-o}, c^{l}_{+o}> and $\underline{W}_c^{r}$ = <c^{r}_{-p}, c^{r}_{+p}, c^{r}_{-l}, c^{r}_{+l}, c^{r}_{-o}, c^{r}_{+o}>, and use these to compute the confidence vector of $p_k^{(m)}$ as:

$$CV(p_k^{(m)}) =< c_p, c_l, c_o > \quad (2)$$

where c_p=c^{l}_{-p}+c^{r}_{+p}, c_l=c^{l}_{-l}+c^{r}_{+l}, c_o=c^{l}_{-o}+c^{r}_{+o}. Here we simply average the probabilities of the left and right context words to derive the final probability vector.

We assign $p_k^{(m)}$ to be part of a named entity of type x, if $c_x \geq \tau_p$ for x∈ {p, l, o}. Here we set τ_p to be 0.8. In case that there are more than one value greater than τ_p, we select the one with the highest value in the type vector as the type of that P-Name.

2.2.2 Evaluating P-Names

We next derive an objective measure to evaluate how likely a candidate in $\underline{P}^{(m)}$ could be a P-Name. We observe that a string is likely to be a P-Name if: (a) it contains some sub-strings that frequently appear in typical P-Names such as “阿尔”, “斯基”, “洛夫”, etc; and (b) it has context words in $\underline{CW}^{(m-1)}$ set that indicates that it has high probability of being part of a named entity. Thus for each P-Name candidate $p_k^{(m)}$ ($p_k^{(m)}$=$c_1c_2…c_n$) in $\underline{P}^{(m)}$, we compute:

$$s(p_k^{(m)}) = s_1(p_k^{(m)})/M + \beta * s_2(p_k^{(m)}) \quad (3)$$

$$s_1(p_k^{(m)}) = \omega \sum_{i=1}^{n} \sum_{j=1}^{N} n_j(c_i) + \omega^2 \sum_{i=1}^{n-1} \sum_{j=1}^{N} n_j(c_i c_{i+1}) + \omega^3 \sum_{i=1}^{n-2} \sum_{j=1}^{N} n_j(c_i c_{i+1} c_{i+2}) \quad (3.1)$$

$$s_2(p_k^{(m)}) = \max(c_p, c_l, c_o) \quad (3.2)$$

where n is the number of characters in $p_k^{(m)}$, and N equals $|\underline{P}^{(m)}|$. $n_j(c_i)$, $n_j(c_i c_{i+1})$ and $n_j(c_i c_{i+1} c_{i+2})$ are respectively number of times the character strings c_i, $c_i c_{i+1}$ and $c_i c_{i+1} c_{i+2}$ in $p_k^{(m)}$ also appear in other P-Name candidates in $\underline{P}^{(m)}$. β and ω are predefined constants (here we use β =0.5 and ω =1.5). Equation (3.1) gives higher weight to $p_k^{(m)}$ that has better match with longer string sequence of, say, $c_i c_{i+1} c_{i+2}$ with other known P-Names candidates. Equation (3.2) selects the highest confidence value of context words around $p_k^{(m)}$ as support for $p_k^{(m)}$. As s_1 and s_2 are of different scales, we normalize s_1 by dividing it by M, the maximum s_1 values found in the current iteration, before fusing the two values in Equation (3).

2.2.3 Generating New P-Name Characters

Since we would like to obtain more new P-Names during bootstrapping, in each iteration, we would like to expand the P-Name character set. In order to select the most likely P-Name characters, we derive a quasi-probability, $Conf(c_i^{(m)})$, to estimate how likely a character $c_i^{(m)}$ in the P-Name candidate set $\underline{P}^{(m)}$ could be used as a P-Name character. To do this, we make use of both the PKUC corpus and $\underline{P}^{(m)}$. We observe that most characters in $\underline{P}^{(m)}$ also appear in the PKUC corpus, sometimes as P-Name characters sometimes as common characters. Thus, intuitively we estimate $Conf(c_i^{(m)})$ by its occurrences in both PKUC and $\underline{P}^{(m)}$ as:

$$Conf(c_i^{(m)}) = \frac{\sum_{k=1}^{N_c} s(p_k^{(m)})}{\sum_{k=1}^{N_c} s(p_k^{(m)}) + N_{neg}} * \ln\left(\sum_{k=1}^{N_c} s(p_k^{(m)})\right) \quad (4)$$

Here we assume that there are N_c P-Name candidates in $\underline{P}^{(m)}$ that contain $c_i^{(m)}$; and N_{neg} is the number of times that $c_i^{(m)}$ is used as single-character word in PKUC. Equation (4) aims to identify characters that appear frequently as part of P-Names, but rarely as part of common words. It also favors characters that appear in more probable P-Names through the $s(p_k^{(m)})$ measures.

Although Equation (4) is effective in identifying individual P-Name characters, it is not good at locating the sequences of P-Name characters that form the P-Names. This is because there are many characters that have low $Conf(c_i^{(m)})$ values that are part of a P-Name. For example, in a P-Name "波列修克", the character "修" has low confidence to be a P-Name character as defined by Equation (4). However, it co-occurs with high confident P-Name characters such as "列" and "克". To overcome this problem, we modify the confidence value of each character by considering its neighbors (context) to derive a smoothed confidence measure in Equation (5).

$$SConf(c_i) = Conf(c_i) + \alpha * \max\{e^{B^-(c_i)} * Conf(c_{i-1}), e^{B^+(c_i)} * Conf(c_{i+1})\} \quad (5)$$

where α is a predefined constant (we use α = 1), and $B^+(c_i) = \frac{C(c_i c_{i+1})}{C(c_i)}$; $B^-(c_i) = \frac{C(c_{i-1} c_i)}{C(c_i)}$.

$Conf(c_i)$ is defined in Equation (4); $C(c_i)$ and $C(c_i c_j)$ is respectively the co-occurrence of characters c_i and $c_i c_j$ in the P-Name set. Equation (5) tries to supplement the confidence of c_i by its context, that is, it uses the higher of the bi-gram statistics with its preceding and succeeding word to enhance its confidence. We rank all the characters in $\underline{P}^{(m)}$ using Equation (5) and add the top new character into $\underline{C}_S^{(m-1)}$ to obtain $\underline{C}_S^{(m)}$.

2.3 Deriving New Context Words

In addition to finding new P-Name characters, there is also a need to expand the context word set $\underline{CW}^{(m-1)}$ in order to help identify more P-Names. As mentioned before, if at least one of c_p, c_l, c_o values of a P-Name candidate in Equation (2) is greater than a threshold τ_p, we regard it as part of a named entity. For these P-Names which could be possible part of named entities, the following steps are performed:

a) We retrieve all their context words in $\underline{R}^{(m)}$.
b) We add all new context words to form $\underline{CW}^{(m)}$.
c) We update probability vectors of the new context words using Equation (1).
d) We group these context words under the category of c_{-x} or c_{+x} (for $x \in \{p, l, o\}$) if their probabilities under that category is greater than a threshold τ_g (say, 0.5).
e) We then perform lexical chaining using HowNet to generalize the context words under each of the 6 categories separately. The general lexical chaining algorithm is given in detail in Chua and Liu (2002).

f) After lexical chaining, some semantically related words are grouped together. We update the confidence vectors of the semantic groups by averaging the confidence values of words in each of the semantic groups.

At the end of this process, we obtain a new set of context word $\underline{CW}^{(m)}$ which contains some generalized context word classes.

3 Identifying P-Names from New Texts

At the end of the bootstrapping process, we obtain expanded lists of likely P-Name characters $\underline{C}_s^{(m)}$, context cue words $\underline{CW}^{(m)}$ and P-Names $\underline{P}^{(m)}$. Given a new document, we want to use these resources to identify all P-Names. The process is carried out as follows:

a) We first use our common word dictionary to remove all common words.
b) Next we use knowledge of P-Name candidates and corpus statistics to identify a sequence of likely P-Name characters. Any sub-string in which the *Sconf*(c_i) (see Equation (5)) of each consecutive character in that string is greater than a pre-specified threshold τ_c (we use 5) is considered as a P-Name.
c) A frequently occurring problem during testing is how to handle new characters not found in the $\underline{C}_s^{(m)}$ set that we do not know their confidence values. Such problem occurs as a same foreign name may be translated to different P-Names with similar Chinese PinYin. For these characters, we adopt the similar homophone approach to relate unknown characters to the known characters in $\underline{C}_s^{(m)}$ set with similar Chinese PinYin.

4 Evaluation

We devise several tests to evaluate our extraction scheme with bootstrapping. We use the MET-2 test corpus for two of the tests, and PKUC as basic language resource to support the process. We use PKUC to extract common word dictionary, which consists of about 37,000 words. We also use PKUC to extract and evaluate typical context cue words around person, location and organization names. Our experiments start from a "seed" P-Name character set:
$\underline{C}_s^{(0)}$ = {"阿", "尔", "巴", "斯", "基"};
and a set of 60 context cue words.

4.1 Obtaining P-Names from the Internet

We perform the bootstrapping process as discussed in Section 2 to extract P-Names from the Internet, and stopped after about 650 iterations. We manually count the number of correct P-Names obtained at the end of every 65-iterations. We also use the first 100,000 P-Names found at the end of the bootstrapping process as the ground truth to compute the accuracy of P-Name identification.

Figure 3 presents the results of the P-Name extraction process. From the figure, we can see that as we increased the number of iterations, the number of P-Names obtained also increased proportionally. This demonstrates that our bootstrapping process is consistent. We also observe that the system is able to maintain a high accuracy of over 85% even when the number of P-Names found approaches 100,000. This demonstrates that our method is effective.

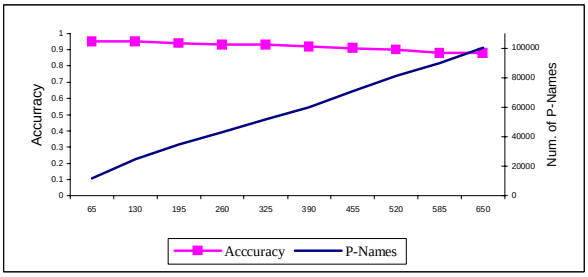

Figure 3: Obtaining P-Names with Bootstrapping

4.2 Extracting P-Names from MET-2 set

We use MET-2 test corpus to test the effectiveness of our approach to identify P-Names from new texts as discussed in Section 3. We consider a P-Name as correctly extracted only when every of its character are correctly identified. The results are presented in Table 1. The results show that we are able to achieve a recall of over 95% and precision of close to 90%. The results are encouraging as we did not use the training resource of MET-2 corpus to train the system, which is expected to lead to higher accuracy.

Table 1: Results of P-Name extraction from MET2

Actual #	System	N_c	N_p	N_m	N_s	Rc	Pr
457	491	437	20	0	34	95.6%	89%

N_c = number of P-Names correctly recognized.
N_p = number of P-Names partially recognized.
N_m = number of P-Names missed.
N_s = number of P-Names found but not in the annotated list.

Recall (Rc) = $N_c/(N_c + N_p + N_m)$;
Precision (Pr) = $N_c /(N_c + N_p + N_s)$.

As a by-product of the PN-Finder, we obtained a large set of context words. We found that we can use these context words to correctly classify about 25% of the extracted P-Names in MET-2 test set into person names or part of location or organization names using the method described in Section 2.2. The employing of context words to classify P-Names is mainly to confirm more P-Names and P-Name characters.

4.3 Contributions of PN-Finder to a Generic NE Recognition Module

The most important contribution of PN-Finder is that it can be used to improve the performance of a generic Chinese named entity recognizer as discussed in Chua and Liu (2002). Here, we conducted several trials by using the PN-Finder to extract a different number of P-Names. We use the first 100,000 P-Names found by the PN-Finder, together with the pattern rules in the general named entity recognizer, to conduct a baseline test. This test merely performs direct table look-up to locate all possible P-Names. Table 2 lists the performance of the general NE recognition system by using an increasing number of P-Names found by the PN-Finder, together with the use of the confidence statistics, context words obtained from the current sets of P-Names and pattern rules. The results indicate that as we increase the number of P-Names found by the PN-Finder, the performance of the general NE recognition system is improved steadily until it reaches over 92% in average F_1 value.

Table 2: Contributions to general NE recognition

# of P-Names used	Ave F_1
100,000 (baseline)	71.3
40,000	88.9
60,000	90.5
80,000	91.7
100,000	92.3

5 Conclusion and Future Work

The presence of P-Names brings more ambiguities to Chinese word segmentation and general Chinese named entity recognition. However, there is a dearth of annotated corpus for extracting and classifying P-Names. To cope with the problem of sparse training resources, this paper presents a bootstrapping module to identify P-Names and classify them into parts of named entitites if possible. The PN-Finder could also contribute to general Chinese named entity recognition and achieve promising performance on the MET-2 test corpus.

Currently, we use only a single word as the context, more context could be considered in the future research. We also aim to extend this method to extract organization names from Chinese documents obtained from the Internet.

References

Agichtein E. and Gravano L. (2000). *Snowball: Extracting Relations from Large Plain-Text Collections*. Proceedings of the 5th ACM International Conference on Digital Libraries.

Brin S. (1998). *Extracting Patterns and Relations from the World Wide Web*. WebDB Workshop at 6th International Conference on Extending Database Technology, EDBT' 98.

Chinchor A. Nancy (2001). *Overview of MUC7/MET-2*. available at:

http://www.itl.nist.gov/iaui/894.02/related_projects/muc/proceedings/muc_7_proceedings/overview.html

Chua T.S. and Liu J.M. (2002), *Learning Pattern Rules for Chinese Named Entity Extraction*. To Appear in AAAI'02.

Collins M. and Singer Y. (1999). *Unsupervised Models for Named Entity Classification*. In Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

Liu J.M. and Chua T.S. (2001), *Building semantic Perceptron net for topic spotting*, In Proceeding of Association for Computational Linguistics 39th Anniversary Meeting, 306-313

Mihalcea F. R. and Moldovan I. D. (2001) *A Highly Accurate Bootstrapping Algorithm for Word Sense Disambiguation*. International Journal on Artificial Intelligence Tools. Vol.10, No 1-2(2001). pp. 5-21

Riloff E. and Jones R. (1999) *Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping*. In Proceedings of the Sixteenth National Conference on Artificial Intelligence, pp. 1044-1049.

Yu S. (1999), *The Specification and Manual of Chinese Word Segmentation and Part of Speech Tagging*, available at:
http://www.icl.pku.edu.cn/Introduction/corpustagging.htm