論文 / 著書情報
Article / Book Information

Title	Vocabulary Expansion through Automatic Abbreviation Generation for Chinese Voice Search
Authors	Dong Yang, Yi-cheng Pan, Sadaoki Furui
Citation	INTERSPEECH 2009 BRIGHTON, , , pp. 728-731,
Pub. date	2009, 9
Copyright	(c) 2009 International Speech Communication Association, ISCA
DOI	http://dx.doi.org/

INTERSPEECH 2009 BRIGHTON

Vocabulary Expansion through Automatic Abbreviation Generation for Chinese Voice Search

Dong Yang, Yi-cheng Pan, and Sadaoki Furui

Department of Computer Science
Tokyo Institute of Technology
Tokyo 152-8552 Japan
{raymond,thomas,furui}@furui.cs.titech.ac.jp

Abstract

Long named entities are often abbreviated in oral Chinese language, and this usually leads to out-of-vocabulary(OOV) problems in speech recognition applications. The generation of Chinese abbreviations is much more complex than English abbreviations, most of which are acronyms and truncations. In this paper, we propose a new method for automatically generating abbreviations for Chinese named entities and we perform vocabulary expansion using output of the abbreviation model for voice search. In our abbreviation modeling, we convert the abbreviation generation problem into a tagging problem and use the conditional random field (CRF) as the tagging tool. In the vocabulary expansion, considering the multiple abbreviation problem and limited coverage of top-1 abbreviation candidate, we add top-10 candidates into the vocabulary. In our experiments, for the abbreviation modeling, we achieved the top-10 coverage of 88.3% by the proposed method; for the voice search, we improved the voice search accuracy from 16.9% to 79.2% by incorporating the top-10 abbreviation candidates to vocabulary.
Index Terms: automatic abbreviation generation, vocabulary expansion, voice search

1. Introduction

In voice search applications, like directory assistance or car navigation systems, users seldom say a target name exactly the same as it appears in the database [1]. Usually what are contained in the database are official names of organizations or companies, which are very long and difficult to remember. Long names are frequently abbreviated for efficiency and convenience, like "google" instead of "google Inc" and "starbucks" instead of "starbucks coffee" [2]. The mismatch between the real queries and entries in the database can cause up to 35% absolute correct-accept rate difference under some condition [2].

There are mainly two methods to solve the problem. A rule based method has been proposed to construct a finite-state signature language model (LM) from all the entries in the database alone [3], which would accept different query variants. Here the signature is a subsequence of the words in a listing that uniquely identifies the listing. Under this method one entry may have either too many or no signatures. Meanwhile the reason behind signature grammar is that any term is droppable as long as the drop does not cause confusion, which may be not the case of real human language.

Another approach to improve robustness is via statistical n-gram models [4]. An interpolated LM was proposed to estimate the n-gram probability in $p(w) = \lambda p_t(w) + (1 - \lambda)p_l(w)$, where $p_t(w)$ is the LM model built using the transcripts of the real calls, and $p_l(w)$ is the LM built using the listing database, and λ is the interpolation weight. This method depends on the amount of real calls very much, but the real call data is usually very expensive to collect.

The signature grammar method tries to list all the word combinations which can identify the entry from others and add the combination into the grammar, and it doesn't consider the property of words and pattern behind the real data at all. The statistical method requires a high quality real data LM, and the training data should cover as many particular examples as possible. Another problem with these two methods is that they can model the variations based on word units, but they cannot solve the sub-word variation problems. For example, "Georgia Institute of Technology", abbreviated as "Georgia Tech", the word "Technology" has been abbreviated as "Tech", this problem cannot be covered by previous methods. This type of abbreviation may be not so common in English, and even if the abbreviation exists, most of them are just acronyms (eg.: "AT&T" abbreviated from "American Telephone & Telegraph".). But many other languages like Chinese and Japanese, the sub-word based abbreviation phenomena is very common and the abbreviation formalization process is much more complex than acronyms as well.

In this paper, we treat the variations between real queries and full-names as abbreviation problems, based in either word units or sub-word units. Our paper focuses on Chinese named entity abbreviations and we try to solve this problem while satisfying following requirements:

- No requirement for a huge amount of training data which covers variations of all the entries in the database
- The ability to predict variations for unseen entries in a statistical way
- Covering sub-word variations

A hybrid automatic abbreviation generation method is proposed here for Chinese. We need a corpus of fullname-abbreviation pairs which doesn't have to be very huge, and then we model the abbreviation generation process as a statistical tagging problem. After trained on the corpus, we can predict the variations of names which are not covered by the fullname-abbreviation corpus. At last we can make use of the abbreviation output through vocabulary expansion in voice search applications. The rest of the paper is divided into four sections: Abbreviation modeling, Vocabulary expansion, Experiments and Conclusions.

2. Abbreviation modeling

A simple abbreviation dictionary cannot solve the abbreviation problem. The first difficulty is that no such dictionary exists, and further there is always OOV for a dictionary and meanwhile new named entities keep on coming into use. An automatic abbreviation generation method is required to tackle this problem.

There has been a considerable amount of research on extracting full-name and abbreviation pairs in the same document for obtaining abbreviations [5, 6, 7]. However, generation of abbreviations given a full-name is still a non-trivial problem. Chang and Lai [8] have proposed using a hidden Markov model to generate abbreviations from full-names. However, their method assumes that there is no word-to-null mapping, which means that every word in the full-name has to contribute at least one character to the abbreviation. This assumption does not hold for named entities which have many word skips in the abbreviation generation. Our method [9] has no such limitations and the structure is displayed in Figure 1, detail of which is given in the following subsections.

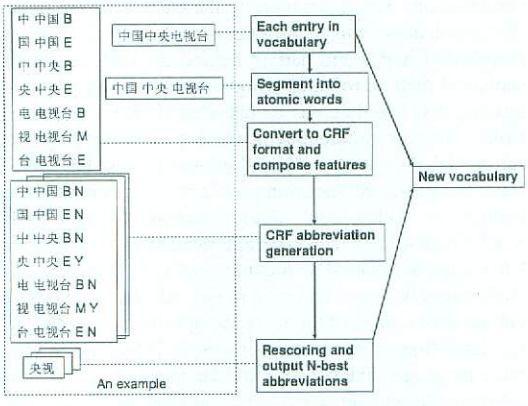

Figure 1: System structure

2.1. Chinese abbreviations

Chinese abbreviations are generated by three methods [10]: reduction, elimination, and generalization.

Both in the reduction and elimination methods, characters are selected from the full-name, and the order of the characters is sometimes changed. Note that this paper does not cover the case when the order is changed. The elimination means that one or more words in the full-name are ignored completely, while the reduction requires that at least one character is selected from each word. All the three examples in Figure 2 are produced by the elimination, where at least one word is skipped.

Generalization, which is used to abbreviate a list of similar terms, usually produces a word which is composed of the number of the terms and a shared character across the terms. A example is "三军" (three forces) for "陆军，海军，空军" (army, navy, air force). This is the most difficult scenario for the abbreviations and is not considered in this paper.

2.2. Segmenting named entities into atomic words

Chinese is written continuously and there is no word boundaries in text data; as a result, a segmenter is usually needed before any further processing. Almost all the Chinese segmenters are

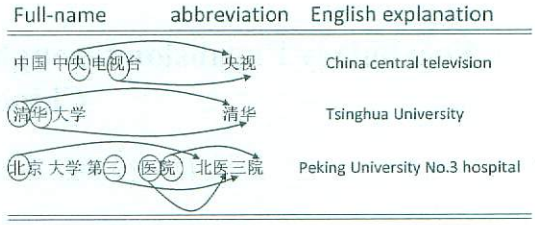

Figure 2: Chinese abbreviation examples

created with embedded functionality of named entity recognition. In our abbreviation modeling, we have to segment each named entity into a list of composing atomic words, eg. "中国中央电视台" (China Central Television) is segmented into "中国" (China), "中央" (central), "电视台" (television). Otherwise, we cannot make use of character position in atomic words, which is very useful in predicting abbreviations.

The Chinese segmenter used in this paper is trained by ourselves. We trained a 2-tag CRF segmenter from the "Penn Chinese Treebank" corpus, in which Chinese text are segmented into atomic words.

2.3. CRF for abbreviation modeling

A CRF [11] is an undirected graphical model and assigns the following probability to a label sequence $L = l_1 l_2 \dots l_T$, given an input sequence $C = c_1 c_2 \dots c_T$,

$$P(L|C) = \frac{1}{Z(C)} exp(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(l_t, l_{t-1}, C, t)) \quad (1)$$

Here, f_k is the feature function for the k-th feature, λ_k is the parameter which controls the weight of the k-th feature in the model, and $Z(C)$ is the normalization term that makes the summation of the probability of all label sequences to 1. CRF training is usually performed through the typical L-BFGS algorithm [12] and decoding is performed by Viterbi algorithm. In this paper, we use an open source toolkit "crf++" [13].

In order to use the CRF method in abbreviation generation, the abbreviation generation problem was converted to a tagging problem. The character is used as a tagging unit and each character in a full-name is tagged by a binary variable with the values of either Y or N: Y stands for a character used in the abbreviation and N means not. An example is given in Figure 3.

Figure 3: Abbreviation in the CRF tagging format

In the CRF method, feature function describes a co-occurrence relation, and it is defined as $f_k(l_t, l_{t-1}, C, t)$ (Eq. 1). f_k is usually a binary function, and takes the value 1 when both observation c_t and transition $l_{t-1} \rightarrow l_t$ are observed. In our abbreviation generation model, we use the following features:

1. Current character
2. Current word
3. Position of the current character in the current word
4. Combination of the above features 2 and 3

2.4. Improvement via incorporating a length model

There is a strong correlation between the lengths of organizations' full-names and their abbreviations. We use the length modeling based on discrete probability of $P(M|L)$, in which the variables M and L are lengths of abbreviations and full-names, respectively. Since it is difficult to incorporate length information into the CRF model explicitly, we use $P(M|L)$ to rescore the output of the CRF.

We model the abbreviation process with two steps:

1st step: evaluate the length in abbreviation according to the length model $P(M|L)$;

2nd step: choose the abbreviation, given the length and full-name, written as $P(A|M,F)$. Our problem becomes:

$$\widetilde{A} = \arg\max_{A,M} P(M|L) \cdot P(A|M,F) \tag{2}$$

where variable A is the abbreviation and F is the full-name. The second term $P(A|M,F)$ is calculated using the CRF, according to the Bayesian rule:

$$P(A|M,F) = \frac{P(A,M|F)}{P(M|F)} = \frac{P(A,M|F)}{\sum_{length(A')=M} P(A',M|F)}. \tag{3}$$

Since A contains the information M implicitly, it is obvious that $P(A,M|F) = P(A|F)$, which can be calculated by the CRF.

2.5. Improvement via a web search engine

Co-occurrence of a full-name and an abbreviation candidate can be a clue of the correctness of the abbreviation. We use the "abbreviation candidate"+ "full-name" as queries and input them to the most popular Chinese search engine (www.baidu.com), and then we use the number of hits as the metric to perform re-ranking. The number of hits is theoretically related to the number of pages which contain both the full-name and abbreviation. The bigger the number of hits, the higher probability that the abbreviation is correct.

We then simply multiply the previous probability score, obtained from Eq. 2, by the number of hits and re-rank the top-30 candidates accordingly.

There are some other ways to use information retrieval methods [14]. Our method has an advantage that the access load to the web search engine is relatively small.

3. Vocabulary expansion

Since the top-1 coverage of the abbreviation modeling is not high enough, we have to add the N-best abbreviation candidates into the vocabulary. There is a tradeoff between the coverage of the abbreviation and the ambiguity caused by a larger vocabulary. If N is too small, the possibility that abbreviation is not included in the vocabulary is high; on the other hand, if N is too big, the confusions along with the increase of the vocabulary size will damage the performance of ASR.

4. Experiments

4.1. Abbreviation generation results

4.1.1. Data collection and introduction

The corpus we use for abbreviation training and evaluation in this paper comes from two sources: one is the book "modern Chinese abbreviation dictionary" [15] and the other is the wikipedia. Altogether we collected 1945 pairs of organization full-names and their abbreviations.

The data is randomly divided into two parts, a training set with 1298 pairs and a test set with 647 pairs. Table 1 shows the length mapping statistics of the training set. It can be seen that the average length of full-names is about 7.29. We know that for a full-name with length N, the number of abbreviation candidates is about $2^N - 2 - N$ (excluding length of 0, 1, and N) and we can conclude that the average number of candidates for each organization name in this corpus is more than 100.

length of full-name	length of abbreviation					sum
	2	3	4	5	>5	
4	107	1	0	0	0	108
5	89	140	0	0	0	229
6	96	45	46	0	0	187
7	60	189	49	16	0	314
8	48	29	60	3	6	146
9	10	47	35	12	2	106
10	18	11	29	8	6	73
others	21	43	38	17	14	133
average length of the full-name						7.27
average length of the abbreviation						3.01

Table 1: Length statistics on the training set

4.1.2. Experimental results

We plan to add up to 10 abbreviation candidates into the vocabulary of our voice search application for each named entity, hence here we consider top-10 coverage of the abbreviation modeling.

Figure 4 displays the coverage results obtained using the CRF method and the improvements gained from the inclusion of the length feature and the web search hits. As we can see the CRF gives a coverage of 79.9%. Both length model and web search engine show significant improvement over the CRF baseline and the coverage increases to 88.3%.

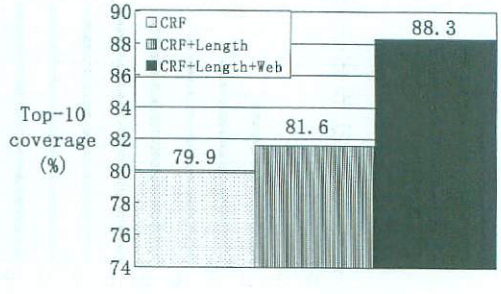

Figure 4: Results of different methods

4.2. Voice search results

4.2.1. Data collection and introduction

We selected 400 named entities from the test set used in previous section and collected speech data for them from 20 speakers. Each speaker was requested to process 80 named entities and each named entity was allocated to 4 speakers. As a result, we could guarantee the variations of the abbreviations were covered by the data.

We noticed that multiple abbreviations for one full-name was very common, such as "中国中央电视台" (China central

television) with abbreviations "央视" and "中央台". We planed to collect multiple abbreviations for reference along with the speech data from each speaker. So our collection process for each speaker was as follows:

1. Display a full-name and the speaker reads it
2. The speaker writes down an abbreviation and reads it
3. If there is another abbreviation, go to 2
4. Proceed to next full-name and go to 1

In total, we collected 783 unique abbreviations, and the average abbreviation number for each full-name was 1.96;

4.2.2. *Experimental results*

In the voice search experiment, the input is a piece of speech and the output is a fullname in the list represented by one or several items (full-name plus abbreviations) in the vocabulary. If the input speech is a full-name, the output is expected to be the full-name; when the speech is an abbreviation, the output should be the corresponding full-name. We measure the performance of voice search by the accuracy of output full-names.

If the abbreviation is not included in the vocabulary, the search will fail theoretically for abbreviated speech. However, when the abbreviation is long enough to be quite similar to the full-name, it is also possible to be recognized correctly and get the correct search result.

In our experiment, we started from the vocabulary containing full-names only, and then added all the top-1 abbreviation candidates into the vocabulary, and then top-2, top-3,..., until top-10. Figure 5 shows that, as the added abbreviation candidates increase, the search accuracy keeps increasing from 16.9% with no abbreviation in vocabulary to 79.2% with 10 abbreviation candidates in the vocabulary for each entry.

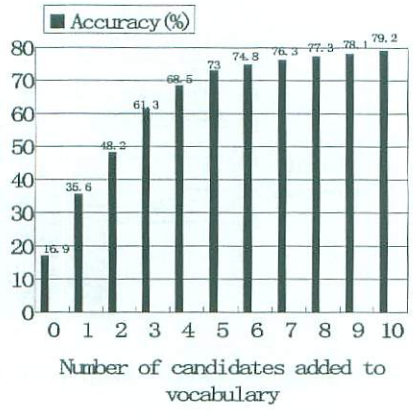

Figure 5: Results of voice search accuracy

5. Conclusions and future work

In this paper, we presented an automatic abbreviation generation method and successfully applied this method into voice search via vocabulary expansion. Our statistical abbreviation generation method uses CRF first and then makes use of length information and web search engine to rescore the abbreviations. The top-10 coverage was able to reach 88.3%. After adding abbreviations into vocabulary, the accuracy of voice search experiment increased from 16.9% to 79.2%.

One limitation of our work is the limited size of test data, and we are planning to apply our method to a much larger database. One direction of our future work will be to make better use of web data, for example, to extract fullname-abbreviation pairs from a huge amount of web text to make a much larger corpus.

6. References

[1] Ye-Yi Wang, Dong Yu, Yun-Cheng Ju and Acero, A., *"An introduction to voice search"*, Signal Processing Magazine, IEEE, May 2008, pages:28-38.

[2] Bacchiani M., Beaufays F., Schalkwyk J., Schuster M. and Strope, B., *"Deploying GOOG-411: Early lessons in data, measurement, and testing"*, Proceedings of ICASSP 2008, pages: 5260-5263.

[3] E.E.Jan, B. Maison, L. Mangu, and G. Zweig, *"Automatic construction of unique signatures and confusable sets for natural language directory assistance applications"*, Proceedings of Eurospeech 2003, pages 1249-1252.

[4] Dong Yu, Yun-Cheng Ju, Ye-Yi Wang, Geoffrey Zweig and Alex Acero, *"Automated Directory Assistance System - from Theory to Practice"*, Proceedings of Interspeech 2007, pages 2709-2712.

[5] Zhifei Li and David Yarowsky, *"Unsupervised Translation Induction for Chinese Abbreviations using Monolingual Corpora"*, Proceedings of ACL 2008, pages 425-433.

[6] Xu Sun, Houfeng Wang and Yu Zhang, *"Chinese Abbreviation-Definition Identification: A SVM Approach Using Context Information"*, Lecture Notes in Computer Science,Volume 4099/2006, pages 495-504.

[7] Guohong Fu, Kang-Kwong Luke, GuoDong Zhou and Ruifeng Xu, *"Automatic Expansion of Abbreviations in Chinese News Text"*, Lecture Notes in Computer Science, Volume 4182/2006, pages 530-536.

[8] Jing-shin Chang and Yu-Tso Lai, *"A Preliminary Study on Probabilistic Models for Chinese Abbreviations"*, Proceedings of ACL SIGHAN Workshop 2004, pages 9-16.

[9] Dong Yang, Yi-cheng Pan and Sadaoki Furui *"Automatic Chinese Abbreviation Generation Using Conditional Random Field"*, Proceedings of HLT-NAACL 2009 (Short), pages 273-276.

[10] Hiu Wing and Doris Lee, *"A Study of Automatic Expansion of Chinese Abbreviations"*, MA Thesis, 2005, The University of Hong Kong.

[11] John Lafferty, Andrew McCallum, and Fernando Pereira, *"Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data"*, In Proceedings of International Conference on Machine Learning 2001, pages 282-289.

[12] Hanna Wallach, *"Efficient Training of Conditional Random Fields"*, M. Thesis, University of Edinburgh, 2002.

[13] Taku Kudo, *"http://crfpp.sourceforge.net/"*.

[14] Rila Mandala, Takenobu Tokunaga, and Hozumi Tanaka, *"Query expansion using heterogeneous thesauri"*, in Information Processing and Management Volume 36, Issue 3, 2000, Pages 361-378.

[15] Hui Yuan and Xianzhong Ruan, *"Modern Chinese abbreviation dictionary"*, Yuwen press, 2002, Beijing, China.