Cross Domain Chinese Speech Understanding And Answering Based On Named-Entity Extraction

Yun-Tien Lee, Shun-Chuan Chen and Lin-shan Lee

Speech Lab, Department of Computer Science and Information Engineering
National Taiwan University, Taipei, Taiwan, ROC
steven@speech.ee.ntu.edu.tw, hypno@speech.ee.ntu.edu.tw

Abstract

Chinese language is not alphabetic, with flexible wording structure and large number of domain-specific terms generated every day for each domain. In this paper, a new approach for cross-domain Chinese speech understanding and answering is proposed based on named-entity extraction. This approach includes two parts: a speech query recognition (SQR) part and a speech understanding and answering (SUA) part. The huge quantities of news documents retrieved from the Web are used to construct domain-specific lexicons and language models for SQR. The named-entity extraction is used to construct a domain-specific named-entity database for SUA. It is found that by combining domain classifiers and named-entity extraction, we can not only understand cross-domain queries, but also find answers in a specific domain.

1. Introduction

Chinese language is not alphabetic. Every character has its own meaning. A word is composed of one to several characters. As a result, Chinese wording structure is quite flexible, and new words are easily generated every day, especially with large number of domain-specific terms for each domain. In addition, because there are no word boundaries in Chinese texts, the segmentation of a Chinese sentence into words is usually not straightforward, and it is a difficult task to identify key terms and named-entities in Chinese texts. All these make cross-domain Chinese language processing difficult, and it becomes even more challenging if speech understanding is considered. In this paper, a new approach for cross-domain Chinese speech understanding and answering is proposed. This new approach is able to handle speech queries across different domains including domain-specific terms outside the general lexicon and provide named-entities as answers to the queries. The performance of understanding is measured by the combined accuracy of domain classifiers, relevant documents and answers.

The proposed approach is based on an approach to extracting named-entities from documents. Using a general domain lexicon and a general domain language model with domain-specific documents collected from the Internet, we can create domain-specific lexicons and language models as well as domain-specific named-entity databases. The approach is not only applicable across different domains, but also can be dynamic with respect to the time-varying Internet content. Very encouraging initial experimental results were obtained, and further tests will be performed in the near future.

2. The Proposed Approach

The proposed approach includes two major parts, namely, speech query recognition (SQR) part as shown in the upper half of Figure 1, and speech understanding and answering (SUA) part as shown in the lower half of Figure 1. The details of these two parts are discussed in the following subsections.

2.1. Speech Query Recognition (SQR)

The detailed components of the speech query recognition (SQR) part include quite several smaller functional blocks, as can be seen in the upper half of Figure 1. Since we focus our discussions on understanding and answering, the necessay parts of acoustic modeling, pronunciation modeling, environmental robustness and speaker adaptation are temporarily ignored here in this paper. Instead, we focus on cross-domain understanding and answering of speech queries, therefore the core problem here is the adaptation of lexicons and language models. As can be seen in the left of the upper half of Figure 1, the recognizer used here is based on RWTH's word-conditioned search method [1]. The output of the recognizer is the 1-best search result. Other parts in SQR are briefly summerized below.

2.1.1. Domain Classifier

The domain classifier is used in order to classify the queries into different domains, such that domain-specific knowledge, from lexicon, language models to the various understanding components, as will be presented below, can be applied. We believe that classifying queries into different domains is the key for better spontaneous speech understanding. Four possible classifiers were adopted here to select which domain the query belongs to:

- Naïve-Bayes Classifier (NB)

$$\hat{d} = \arg\max_d p(d|w_q) = \arg\max_d p(d)p(w_q|d)$$
$$= \arg\max_d p(d)\prod_i p(w_i|d), \quad (1)$$

  where w_q is the vector representation of the input query q with dimension V, with V being the vocabulary size (an element is 1 if a given word is in the query and 0 otherwise), d being a given domain, $\hat{d}$ being the target domain, and $p(w_i|d)$ being the probability that a word w_i appears in a query of domain d. This classifier was trained with training documents with known words and known domains [2].

- Support Vector Machine (SVM) Classifier

  We used the "LIBSVM" tool kit developed by Prof. Chih-Jen Lin of National Taiwan University [3]. The training features are based on tf-idf computation of terms in the domains.

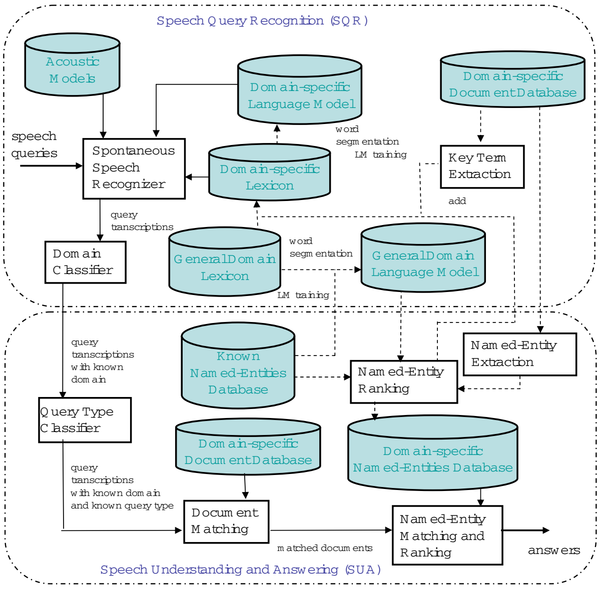
Speech Query Recognition (SQR)
Acoustic Models
Domain-specific Language Model
Domain-specific Document Database
speech queries
Spontaneous Speech Recognizer
Domain-specific Lexicon
word segmentation
LM training
Key Term Extraction
add
query transcriptions
Domain Classifier
General Domain Lexicon
word segmentation
General Domain Language Model
LM training
query transcriptions with known domain
Known Named-Entities Database
Named-Entity Ranking
Named-Entity Extraction
Query Type Classifier
Domain-specific Document Database
Domain-specific Named-Entities Database
query transcriptions with known domain and known query type
Document Matching
matched documents
Named-Entity Matching and Ranking
answers
Speech Understanding and Answering (SUA)

Figure 1: *The System Diagram.*

- N-Gram Classifier (N-GRAM)

$$\hat{d} = \arg\max_d p(d|w_q) = \arg\max_d p(d)p(w_q|d)$$
$$= \arg\max_d p(d) \prod_i p(w_i|d, w_1, w_2, ..., w_{i-1}), \qquad (2)$$

  where $p(w_i|d, w_1, w_2, ..., w_{i-1})$ is the domain-specific n-gram probability. We used trigram probability $p(w_i|d, w_{i-2}, w_{i-1})$ to approximate it [2].

- Latent Semantic Indexing (LSI) Classifier [4]

  First we construct a tf-idf weighted term-goal matrix X and normalize the rows of the matrices. Then let X go through singular value decomposition $X = USV^T$ to produce left orthomormal matrix U, singular value matrix S and right orthomormal matrix V. When a query q comes, its vector representation w_q is first multiplied by the matrix U, and cosine measure is used to select from the matrix VS the highest-scored vector, which will be the target domain.

2.1.2. *Construction of domain-specific lexicons and language models*

Here we start with a general domain lexicon, a general domain language model and domain-specific documents collected from the web, trying to construct domain-specific lexicons and language models. The construction process is shown in the rest parts of the upper half of Figure 1, which consists of the following steps:

- The initial general domain lexicon is the Chinese Electronic Dictionary (CED), and the initial corpus for training general domain language model is the Academia Sinica balanced corpus (ASBC) both offered by Academia Sinica at Taipei [5].
- Because there are no word boundaries in Chinese texts and Chinese wording structures are flexible, key terms are extracted from domain-specific documents using the method based on mutual information proposed previously [6].
- Add the newly-extracted key terms and extracted named entities for a given domain to the general domain lexicon to construct a domain-specific lexicon. The method for extracting named entities will be explained below.
- Use the domain-specific lexicon to perform word segmentation on the retrieved documents for the given domain, and this is again because that there are no word boundaries in Chinese documents.
- Train a domain-specific language model from the word-segmented documents using SRI tools [7].

2.2. Speech Understanding and Answering (SUA)

The complete process of Speech Understanding and Answering (SUA) also includes quite several smaller functional blocks, as can be seen in the lower half of Figure 1. The core process is the bottom path from left to right. The query transcription with known domain comes from the SQR part. The queries are first classified based on their types, then indexed to documents based on latent semantic analysis, and from the matched documents, the named-entities were found and ranked as answers, using the domain-specific named-entity database. The matched documents may also be generated and presented to the user, because the user may wish to know further information regarding his query. This bottom path requires two databases, a known named-entities database and a domain-specific named-entities database. The construction of these databases will be explained below.

2.2.1. *Query type classifier*

The query type classifier is used to classify the queries regarding which kind of question they belong to. For example, the query ”Who is the prime minister of this country?” may be classified as a ”WHO” query, and the answer must be a person or a group of people. We adopted a keyword-based query type classification method here. For example, if the keyword ”who” appears in a query, it should be classified as a ”WHO” query. Since a query may have more than one type, there may be many types in the output of the classifier. In the initial experiments to be presented below, 3 query types are defined: WHO, HOW_MANY and WHERE.

2.2.2. *Document matching*

We match the query transcriptions with known domain to the documents in that domain based on latent semantic analysis [8]. We first construct a term-document matrix Y, and then put it into singular value decomposition process $Y = ACB^T$. When transcriptions come, we multiply the term vector of the transcriptions with the matrix A, and then compare the result vector with vectors in BC matrix and select top N most possible documents.

2.2.3. *Construction of the known named-entities database*

The Chinese Electronic Dictionary (CED) mentioned above contains not only the general domain lexicon, but some part-of-speech information, such as a term may be a noun or verb. For example, ”Nba+mankind” indicates this is the name of some person or a group of people, and ”Nca+organizations,terrains,buildings,districts,organizations” are for names of places or organizations. This database is used for picking up good cadidates of named-entities and filtering out not-so-good ones.

2.2.4. Named-Entity extraction

Key term extraction methods are useful for those key terms having good statistical behaviors or meeting some special criteria. But these key terms are not always meaningful terms. Named-Entity extraction, on the other hand, is not only important for text segmentation, but for extracting meaningful key terms also. Although there are errors, named-entity extraction methods will almost always outperform statistical term extraction ones in the intent of knowing queries. In this paper, we consider three different methods for extracting three different kinds of named-entities based on some approaches mentioned previously [9] [10], with some modifications to improve the results further. The details are stated below.

- Methods for extracting numbers and units

  Since there are relatively strict rules for the wording structure of numbers and units in Chinese, we used several of such rules to extract Chinese numbers. For units, they must be preceded by a number and they must belong to one of the terms in the *Unit Handbook* [11].

- Methods for extracting names of people

  Personal names in Chinese are at least two characters and no more than five characters. Very often characters used in male and female names come from different sets of characters. Last names may contain one character or two characters. And for females, last names may contain two one-character last names to form a two-character last name. With these principles as guidelines, we collect different databases for male and female names in Chinese, totally about 20,000, together with a *Last Name Handbook* [12]. We first segment the documents using the general domain lexicon which includes all characters as mono-character words, and then extract the personal names from the segmented documents with numbers and units filtered out according to the following criteria:

  – Try to concatenate all segmented mono-character words (those can not be matched to a poly-character word) to form a string S. Compound words, which contain more than one character, must be considered due to the coincidence of first names also being in the general domain lexicon, but it is considered when it contains last names only. The length of S must be between two and five characters.

  – Let M be the possible last names found in S and N be the possible first names found in S. MN (in Chinese names the last names are usually printed first and the first names follow) will be considered a male name only if $p(MN) > th1$ and $p(N) > th2$, and it will be considered a female name only if (a) ($p(MN) > th3$ and $p(N) > th4$) or (b) ($p(M) > th5$ and $p(N) > th6$). The thresholds $th1$ through $th6$ are also trained from the name databases.

- Methods for extracting names of places and organizations

  Most names of places and organizations in Chinese contain keywords. For example, in “台北市(Taipei City)”, the character “市 (City)” is a keyword. We obtained the keyword set from CED. Because there are more errors in names of Chinese places and organizations, we extract names of places and organizations after segmenting the documents using general domain lexicon, with extracted numbers and names. From the segmented documents, scan the terms to find if they contain keywords, and if they do, try to concatenate preceded terms until one of the following conditions holds:

  – A name of a person or known place has been found. In this case, the concatenated term is considered a place or an organization and concatenation will be stopped. This is because many place and organization names begin with names of people or places.

  – A number has been found. In this case, the concatenation will be stopped, and the number will not be added to the concatenated term.

  – Another keyword has been found. In this case, the concatenated term will be considered a name of a place or an organization, and the concatenation will still continue, because some places and organizations contain consequtive keywords. For example, “台北縣新店市 (Hsin-Tien City in Taipei County)”.

All the extracted named-entities must be added to the domain-specific lexicon to ensure they are recognizable.

2.2.5. Construction of domain-specific named-entity database

From the known named-entities and general domain language model, the distribution of these named-entities can be obtained. We can then rank the named-entities extracted from the above using some distance measure. This is because we may extract some wrong named-entities, so we have to rank and select the most probable named-entities. We use KL-distance measure to rank the named-entities [13] according to the distance of the left and right contexts of the named-entities.

3. Experiments

The domains used in our initial experiments are HEALTH(HLT) and RELEXATION(RXN) domains selected from *Yahoo!* web news [14]. We consider here three query types in each domain, WHO, HOW_MANY and WHERE. The number of documents in each domain is 3,000, and the sizes of the documents are 3,200KB and 3,300KB, respectively, in each domain.

3.1. Tests with Text queries

We collected 100 queries for each query type and each domain, totally 600 text queries. These queries were first segmented using domain-specific lexicons.

- Domain classifiers

  As stated above, we implemented four domain classifiers. We used four domain classifiers simultaneously (SML) and took majority vote to decide the domain. If there are ties, we choose the result classified by the n-gram classifier because its performance is the best. The results are shown in Table 1. We can see that if we use the four classifiers at the same time (SML), the performance is better than any single classifier.

- Document matching

  We used latent semantic analysis to retrieve the relevant documents considering the queries and selected the top 20 documents. The match rate is the performance for the correct document occurring in the top 20 documents. The results are shown in the last column of Table 1.

- Named-entity extraction

  Personal names are for WHO, numbers and units are for HOW_MANY, place and organization names are for WHERE. The recall rates are shown in Table 2. We can see that numbers and units are the easiest to extract because of their simple rules while place and organization names are hard to extract because of their flexibility in wording structure. The results of personal names stand between.

- Overall result

  After the 20 relevant documents are generated, we see if these documents contain the desired named-entities as answers. The results are shown in Table 3. The errors come from three different sources: the domain classifier, the document matching process and the named-entity extraction method. We have to improve the accuracy of the three components in order to make the overall performance better.

3.2. Tests with Speech queries

We collected 20 speech queries for each query type in each domain, totally 120 queries. The results are shown below.

- Domain classifiers

  The same four classifiers are used here for speech query transcriptions. We also use the classifiers simultaneously (SML). The results are shown in Table 4. From the table, we know that using classifiers simultaneously improves the accuracy more in speech queries than in text queries.

- Document matching

  We also select top 20 documents in this case. The match rate is the performance for the correct document occurring in the top 20 documents. The recognition error is measured by character accuracy. The results are shown in Table 4. The match rate does not degrade as much as that expressed in recognition errors. This is part of the advantages of using latent semantic analysis.

Table 1: *Accuracy of domain classification and document matching for different domains of text queries.*

DOMAIN	NB	SVM	N-GRAM	LSI	SML	match rate
HLT	91%	82%	95%	99%	99%	85%
RXN	93%	83%	99%	89%	99%	83%

Table 2: *Recall rates of named-entity precision in different domain.*

DOMAIN	WHO	HOW_MANY	WHERE
HLT	82%	98.4%	62.9%
RXN	83%	95.9%	77.4%

Table 3: *Results of overall system of text queries.*

DOMAIN	WHO	HOW_MANY	WHERE
HLT	70%	89%	52%
RXN	71%	79%	62%

Table 4: *Accuracy of domain classification and document matching for different domains of speech queries.*

DOMAIN	NB	SVM	N-GRAM	LSI	SML	match rate	recog. error
HLT	84%	90%	80%	80%	90%	65%	30%
RXN	82%	73%	93%	82%	93%	72%	20%

Table 5: *Results of overall system of speech queries.*

DOMAIN	WHO	HOW_MANY	WHERE
HLT	55%	65%	28%
RXN	67%	65%	40%

- Overall result

  The results are measured based upon the same method as in the text queries cases. The results are shown in Table 5. The recognition error adds in the error sources of the overall results. We are trying to improve our methods for better recognition rate and named-entity extraction.

4. Conclusions

In this paper, we proposed a new approach toward Chinese speech understanding and answering across domains. We tried to classify queries into different domains. We showed feasible approaches to constructing domain-specific lexicons and language models as well as various named-entity databases from documents collected from the Internet. One possible application is to find important named-entities in a given time period using speech input when the user wants to retrieve them.

5. References

[1] S. Ortmanns and H. Ney, "A word graph search algorithm for large vocabulary continuous speech recognition," *Computer Speech and Language*, pp. 43–72, 1997.

[2] Y.-Y. Wang, A. Acero, C. Chelba, B. Frey, and L. Wong, "Combination of statistical and rule-based approaches for spoken language understanding," *Proceedings of ICSLP*, pp. 609–612, 2002.

[3] http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html.

[4] J. Chu-Carroll and B. Carpenter, "Vector-based natural language call routing," *Computational Linguistics*, vol. 25, no. 3, 1999.

[5] A. S. Institute of Information Science, "Chinese electronic dictionary for coling and academia sinica balanced corpus."

[6] G. Saon and M. Padmanabhan, "Data-driven approach to designing compound words for continuous speech recognition," *IEEE Trans. on Speech and Audio Processing*, vol. 9, no. 4, May 2001.

[7] http://www.sri.com.

[8] George W. Furnas et al., "Information retrieval using a singular value decomposition model of latent semantic structure," *J. ACM*, pp. 465–480, 1988.

[9] H.-H. Chen, Y.-W. Ding, and S.-C. Tsai, "Named-entity extraction for information retrieval," *Computer Processing of Oriental Languages*, vol. 11, no. 4, 1998.

[10] J. Sun, J. Gao, L. Zhang, M. Zhou, and C. Huang, "Chinese named entity identification using class-based language model," *COLING*, 2002.

[11] http://www.edu.tw/mandr/allbook/liangtz/c7.htm.

[12] http://www.greatchinese.com/surname/surname.htm.

[13] A. Pargellis, E. Fosler-Lussier, and A. Tsai, "Using part-of-speech tags, context thresholding and trigram contexts to improve the auto-induction of semantic classes," *Proceedings of ICSLP*, pp. 605–608, 2002.

[14] http://www.yahoo.com.tw.