Search by Voice in Mandarin Chinese

Jiulong Shan[1], *Genqing Wu*[1], *Zhihong Hu*[2], *Xiliu Tang*[1], *Martin Jansche*[2], *Pedro J. Moreno*[2]

[1]Google China Engineering Center, No. 1 Zhongguancun East Road, Beijing 100084, PRC
[2]Google Research, 76 Ninth Avenue, New York, NY 10011, USA

jlshan@google.com, wugenqing@google.com, hzhsch@gmail.com, xiliu.tang@gmail.com
mjansche@google.com, pedro@google.com

Abstract

In this paper we describe our efforts to build a Mandarin Chinese voice search system. We describe our strategies for data collection, language, lexicon and acoustic modeling, as well as issues related to text normalization that are an integral part of building voice search systems. We show excellent performance on typical spoken search queries under a variety of accents and acoustic conditions. The system has been in operation since October 2009 and has received very positive user reviews.

Index Terms: speech recognition, Mandarin, voice search

1. Introduction

Over the last years there has been a tremendous amount of work in developing voice search systems in English [1]. However, comparatively little effort has been expended on other languages. The languages of East and South East Asia pose special challenges to both users and machines: The writing systems of Chinese, Japanese, Korean, Vietnamese, and Thai (among others) do not indicate word boundaries as clearly as modern European languages and require special word segmenters for processing by machine. The use of complex scripts with large character inventories make these languages hard to type in general, and even more so on small mobile devices. Given the special difficulty of typing in these languages, the availability of speech input can provide tremendous benefits to users.

As part of our ongoing internationalization efforts, to make voice search available broadly, our first decision was to identify what language should be our first target beyond English. As we looked at several metrics, such as number of speakers, annual growth of search traffic, statistics on the use of smart telephones etc., Mandarin quickly emerged as the most suitable candidate. Furthermore, we believed that Mandarin would expose our speech recognition infrastructure to many internationalization problems such as:

- Support for non-Latin scripts
- Dealing with mixed lexicons (Mandarin, English) and code switching
- Use of segmenters in text processing
- Data collection issues
- Language modeling problems

The system described in this paper represents our first efforts in expanding Google Search by Voice beyond the English language and provides us with a template for internationalization that we have successfully applied to other languages and locales.

The outline of the paper is as follows. In section 2 we describe our data collection strategy. In section 3 we present details of how our language models were built. We continue in section 4 describing our lexical modeling choices. In section 5 we describe our acoustic modeling approach. Finally we present our experimental results in section 6 and conclude with section 7.

2. Acoustic data collection and selection

The choice of an acoustic training corpus is critical for the success of a speech recognition application. Ideally the data collected must be as close as possible to the data observed in a deployed production system. For this reason commercially available corpora are often not useful, since they are not matched to our task. To address these problems, we decided to collect our own data. We use a client/server application consisting of a client application running on an Android mobile telephone with an intermittent connection to a server. The system is further described in greater detail in [2].

The application presents queries for users to read and caches the recorded audio data for batch upload to the server, when a network connection is available. We selected more than 100,000 queries from anonymized google.cn logs after some filtering to remove offensive terms. More than 1200 speakers with different cultural and educational backgrounds where selected for the data collection. We asked them to read the queries under varied acoustical conditions such in the office, in the street, in restaurants, etc. We also made an effort to select speakers from a variety of language backgrounds, whose native dialects include several varieties of Mandarin, Wu, Xiang, Gan, Kejia and Cantonese. More than 250,000 utterances were ultimately collected from these speakers.

About 20% of the queries we asked users to read contained English terms. To our surprise we discovered that users often had trouble speaking these terms. In other words, while users are happy to type English queries on search engines, they don't feel as confident speaking them. This insight was later used in refining the text used for language modeling; see section 3 for further details.

3. Language model development

Our system uses an *n*-gram language model over Mandarin words. As in all speech applications, language model performance is heavily dependent on the domain of the training corpus. Voice web-search applications have the added difficulty that they represent a wide and general domain, as users can

search for virtually anything. It is therefore quite different from more traditional speech recognition applications such as automatic dictation or broadcast news transcription. Issues like corpus selection and filtering, choice of language model n-gram order, vocabulary generation and selection, corpus segmentation, and language model pruning should be well investigated to find out the best combination for voice search.

We use typed web queries as our choice of data source for language modeling. Our selection is based on the assumption that spoken queries will be somewhat similar to typed queries. We take advantage of the large number of written queries that are submitted to search engines every day. These queries come from every corner of the world and relate to every aspect of people's daily life.

To build our language model for Mandarin we selected and processed web queries in the following way:

- We selected simplified Chinese queries from domain google.cn.
- User segmentations were rejected and the data reformatted by running our own segmenter.
- Queries where further processed removing some unwanted terms, such as:
  - Very long queries were removed, under the assumption that they would not fit a voice search application, i.e. nobody would speak them.
  - Punctuation was removed, as this is not typically spoken.
  - Non-Chinese and non-English queries were further removed. We did maintain however a small list of English allowed words, mostly words that are sufficiently common in daily Chinese usage.
  - URLs were normalized, i.e. mapped from their written form to their spoken form.
  - Offensive terms such as sexually explicit queries were further removed.

We did explore using different amounts of training data, but in general we found that beyond using a few months worth of queries there was not much space to improve performance by adding more data. With this corpus and a testing corpus which contained a few thousand popular queries, experiments were conducted to compare the perplexities of 3-gram, 4-gram and 5-gram models. Refer to Table 1 for details.

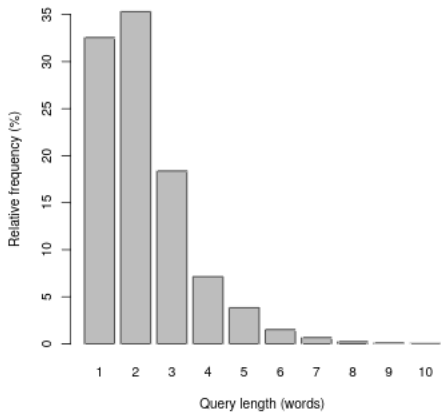
Table 1: *Perplexity comparison for 3, 4 and 5-gram models.*

n	perplexity
3	52.90
4	44.79
5	43.59

In the experiments, Katz smoothing [3] and entropy-based pruning [4] were used, and count-cutoff thresholds of 0, 0, 0, 1 and 2 were used for n-gram units (n is from 1 to 5) respectively. We can see that a 4-gram model is a good choice, as we did not get a significant improvement by increasing the order to 5. One reason for this is that most web queries are very short: The distribution of query length is shown in Figure 1.

Since there are no spaces separating tokens in written Mandarin, the concept of *word* is a fuzzy one. For example, 中国人民 (people of China) can be regarded as a single word, and it can also be treated as the combination of word 中国 (China) and 人民 (people). In our system, we use a segmenter which can automatically generate the top m words for a given vocabulary size m.

Figure 1: *Length distribution of web queries.*



To identify the proper vocabulary size for our system, we measure perplexity using a 3-gram model on the previously mentioned testing set for different vocabulary sizes. Table 2 shows our results.

Table 2: *Perplexity comparison for vocabulary size.*

size	perplexity
250k	51.46
500k	51.65
800k	41.76

We can see that vocabulary size beyond a few hundred thousand words does not have a big impact on the performance of a language model. Despite the drop in perplexity between 500k and 800k words, we saw no corresponding improvements in recognition accuracy, while increasing tremendously the memory footprint of the system. For these reasons, we decided to use a vocabulary size of about 500k words.

4. Lexicon development

In developing a pronunciation lexicon for Mandarin voice search we faced several challenges: (1) The use of words, rather than characters, in our language model means that the lexicon needs to supply pronunciations for words. While the inventory of Chinese characters is comparatively small (around 27,000 in the Basic Multilingual Plane of Unicode[1], out of which only about 3,000 occur frequently), our word-based vocabulary is large, comprising hundreds of thousnads of words. There simply are no existing Mandarin word dictionaries that would provide pronunciations for all of these words. (2) The occurrence of many foreign words, written in Latin characters, in our language model means that we need to assign them approximate Mandarin pronunciations. (3) Our language model further contains many non-standard words, including numbers and alphanumeric sequences. For example the query 三星e428 com-

[1] http://unicode.org

bines Chinese characters (to write the Korean brand name *Samsung*, which has the Mandarin pronunciation 'sānxīng'), Latin letters, and Arabic digits. We describe our efforts to address these challenges in the rest of this section.

First, for pronouncing Chinese words, our lexicon component first consults a small static dictionary of common words, idiomatic expressions, and frequent characters. This dictionary covers only about 10% of our entire vocabulary. This is augmented with a fallback rule set that disambiguates each Chinese character within its word context. The combination of static dictionary plus rules assigns a Pinyin pronunciation to every word in our vocabulary that's written entirely with Chinese characters. From Pinyin we map to our phonetic representation by a simple secondary dictionary lookup, since there are only about 2000 Pinyin syllables. Pivoting through Pinyin is beneficial in several regards: During lexicon development, it aids debugging, since people are much more accustomed to reading Pinyin than they are to reading any kind of phonetic notation. Moreover, the Pinyin representation of a character or word is standard and more or less fixed, whereas there is no standard choice for the phonetic inventory. By going via Pinyin, it becomes easy to experiment with different phone inventories while leaving the bulk of the lexicon representation unchanged.

For pronouncing foreign words, we first observed that most of them seem to either come directly from English or have well-known English pronunciations. We simply look up these words in a very large English pronunciation dictionary we previously developed for an English voice search system. This gives us the English pronunciation of these words, which we then map to Mandarin phones by simple context-independent phone substitutions that replace each English phone by a phonetically similar Mandarin phone.

Non-standard words, including numbers, portions of URLs, and other alphanumeric strings, are handled along similar lines: Frequent word-fragments such as 'www.' have entries in a small hand-crafted exception dictionary. Numbers up to 10,000 are expanded offline into their numeric reading. Longer digit strings are pronounced digit-by-digit, with a twist: The Chinese character that represents the number one has two readings in Mandarin, either *yī* or *yāo*. If we allowed these two readings to mix freely, than a long string of ones would have exponentially many pronunciations. Instead, we assume that the digit one is pronounced consistently as either *yī* or *yāo* within the same string, so that long digit strings have at most two pronunciations.

5. Acoustic modeling

Our acoustic models are standard 3-state context dependent (triphone) models with a variable number of Gaussians per state. These are trained on a 39-dimensional vector composed of PLP cepstral coefficients and their first and second order derivatives. Cepstral mean normalization is applied as well as an energy based endpointer to remove excessive silence. Our frontend also uses Linear Discriminant Analysis (LDA). We use standard decision-tree state-based clustering followed by semi-tied covariance (STC) modeling [5] and an FST-based search [6]. Our acoustic models are gender independent, maximum-likelihood trained followed by boosted MMI (BMMI) [7].

Mandarin is a tonal language, using 4 or 5 tonemes, where the tone of a syllable conveys semantic information. Therefore our acoustic modeling approach must account for these tonal differences somehow. In our system we opted for an indirect modeling approach, i.e. we did not use pitch features in our front-end feature vector representation as our initial experiments didn't show any improvements over the baseline. We assume that tone distinctions are tied to syllable nuclei and we differentiate vowels and diphthongs by tone. We use an inventory of 75 phonemes where vowels/diphthongs are replicated several times, one per tone. Notice that we do not explicitly model the light fifth tone. Table 3 shows the actual tonemes used for acoustic modeling.

Table 3: *Phoneme+toneme inventory*

phoneme	tone variants
y	y1 y2 y3 y4
u	u1 u2 u3 u4
uo	uo1 uo2 uo3 uo4
ou	ou1 ou2 ou3 ou4
i	i1 i2 i3 i4
ih	ih1 ih2 ih3 ih4
ei	ei1 ei2 ei3 ei4
e	e1 e2 e3 e4
a	a1 a2 a3 a4
ai	ai1 ai2 ai3 ai4
ao	ao1 ao2 ao3 ao4
@	@1 @2 @3 @4

While this results in a phonetic inventory that is larger than usual, it allows us to model tones indirectly. Our tree clustering algorithm is adapted to this inventory with appropriate tone related questions.

6. Experimental results

We use a testing set composed of about 4,000 queries coming from real users. Our testing set only contains Mandarin utterances, given the problems we encountered during data collection with English queries and the fact that most English queries were poorly spoken.

6.1. Performance metrics

We used different accuracy metrics to analyze the quality of our system. In Mandarin it is typical to measure character error rate (CER), which in practice maps to syllable error rate. However, our system often contains non-standard words. So our approach is to re-segment Mandarin recognition hypotheses to single characters while leaving non-standard words and other tokens (such as numbers or URLs) unsegmented. Therefore when we report CER we are actually reporting a hybrid metric combining CER for Mandarin and WER for the other tokens.

In general we focus mostly on web metrics such as *web score at one* (WSC@1). In this metric we compare the top web search results produced by the recognition hypothesis with the top web result produced by the reference transcript. We count the percentage of overlap between these two results. A WSC@1 of 100% would imply that there is no difference between the reference transcript and the recognition hypothesis from the point of view of web search. Note that often this is possible even if the reference transcript and recognition hypothesis are different, since web search engines often perform many complex

normalizations internally. Notice also that the metric can be extended to deeper lists such as Web Score at 5 (WSC@5) where we compare a list of the top 5 results from the search engine. Five results is arguable also a practical metric for current mobile devices given the size of their screens.

6.2. Experiments

In our acoustic modeling experiments we explored different clustering tree sizes as well as different modeling sizes (number of Gaussians). Our initial system was trained with about 250,000 transcribed utterances collected following the procedures described in Section 2. We refer to this training set as *TrainSet1*. For tuning purposes we did a rough parameter space exploration and quickly settled on an acoustic model with roughly 2,000 clustered states with about 100,000 Gaussians. We also explored the effect of different language model n-gram orders.

After the system was publicly launched we collected several additional training corpora. First we augmented the initial 250,000 training corpus with an additional 500,000 transcribed production utterances, creating a training set with 750,000 utterances. We refer to this training set as *TrainSet2*. We later created a third training set with an additional 250,000 transcribed utterances, hence containing one million utterances. We refer to this training set as *TrainSet3*. Table 4 shows our experiments with acoustic models trained on each training set and different n-gram orders.

Table 4: *Results with different training sets*

Train set (number of utterances)	LM order	CER	WSC@1
TrainSet1 (250k)	4-gram	35.3%	50.2%
TrainSet2 (750k)	4-gram	25.5%	59.9%
TrainSet2 (750k)	5-gram	25.9%	59.4%
TrainSet3 (1M)	4-gram	23.1%	63.3%

As expected we observe how adding more acoustic data further improves the performance of our system. It is remarkable how once the system is trained with real production data we observe an absolute error rate reduction of almost 10 percentage points and a similar increase in WSC@1. Indeed adding even more data by going from *TrainSet2* to *TrainSet3* yields yet another significant improvement of almost 4% absolute points in WSC@1. In light of these initial experiments our expectation is that more training data will continue to provide significant benefits.

Due to the great dialectal variability in mainland China we were concerned about its effects on recognition performance. In particular we looked at the performance of mostly Cantonese accented Mandarin. We selected a testing subset of utterances coming from Guangdong. This smaller set contained about 2,000 utterances. We observed a degradation of CER of up to 27.0% and WSC@1 down to 62.3%. Clearly as we collect more regional data our system can benefit from acoustic adaptation techniques. We plan to explore these ideas in future work.

Once the traffic to our voice search system became large enough, the recognition hypotheses themselves became a viable source of language modeling training data. We built a small language model with around 350k automatically transcribed queries and interpolated it with our best 4-gram query-based language model. We used an interpolation weight of 0.5. Table 5 shows our results. We observe a further improvement in CER and WSC.

Table 5: *Results of language model interpolation using TrainSet3*

Train set	LM type	CER	WSC@1	WSC@5
TrainSet3	Queries	23.1%	63.3%	64.3%
TrainSet3	Interpolated	21.0%	65.6%	66.5%

7. Conclusions

In this paper we have described our initial efforts building a Mandarin Chinese voice search system. It was successfully launched in October 2009 and since then it has served a large variety of searches from Mandarin speakers in Mainland China, Taiwan, Hong Kong, Singapore and other Chinese speaking regions across the world.

Our experiments show consistent gains by increasing the amounts of acoustic training data. They also show that using unsupervised recognition hypotheses as a source of language model training data leads to continuing improvements. Indeed we do not seem to have reached a saturation point for the current acoustic and language modeling techniques we use. The opinion of users in mainland China has been extremely positive and the system continues to get positive reviews. This system represents the first efforts beyond English for Google's speech team and as such it provides us with a template to iterate across many more languages.

8. Acknowledgements

We want to thank Frank Tang, Bin Lin, Mike Cohen and the rest of the Google speech team for their insightful discussions and inputs.

9. References

[1] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, "Google Search by Voice: A case study," in *Visions of Speech: Exploring New Voice Apps in Mobile Environments, Call Centers and Clinics*, A. Neustein, Ed. Springer, 2010 (in press).

[2] T. Hughes, K. Nakajima, L. Ha, A. Vasu, P. J. Moreno, and M. LeBeau, "Building transcribed speech corpora quickly and cheaply for many languages," in *Interspeech*, 2010 (submitted).

[3] S. M. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recogniser," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35(3), 1987, pp. 400–401.

[4] A. Stolcke, "Entropy-based pruning of backoff language models," in *Proc. DARPA News Transcription and Understanding Workshop*, 1998, pp. 270–274.

[5] M. Gales, "Semi-tied covariance matrices for hidden Markov models," *IEEE Transactions on Speech and Audio Processing*, vol. 7(3), pp. 272–281, 1999.

[6] M. Mohri, F. C. N. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Springer Handbook on Speech Processing and Speech Communication, Part E: Speech Recognition*, 2008.

[7] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature space discriminative training," in *ICASSP*, 2008.