Improved Spoken Query Transcription using Co-Occurrence Information

Jonathan Mamou[1], *Abhinav Sethy*[2], *Bhuvana Ramabhadran*[2], *Ron Hoory*[1], *Paul Vozila*[3]

[1]IBM Haifa Research Labs, Haifa 31905, Israel
[2]IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, U.S.A.
[3]Nuance Communications, One Wayside Road, Burlington, MA 01803, U.S.A.

mamou@il.ibm.com, {asethy,bhuvana}@us.ibm.com, hoory@il.ibm.com,
paul.vozila@nuance.com

10.21437/Interspeech.2011-260

Abstract

Spoken queries are a natural medium for searching the Mobile Web. Language modeling for voice search recognition offers different challenges compared to more conventional speech applications. The challenges arise from the fact that spoken queries are usually a set of keywords and do not have a syntactic and grammatical structure. This paper describes a co-occurrence based approach to improve the accuracy of voice queries automatic transcription. With the right choice of scoring function and co-occurrence level, we show that co-occurrence information gives a 2% relative accuracy improvement over a state of the art system.

Index Terms: voice search, language model, co-occurrence, information retrieval.

1. Introduction

The rapid growth of mobile devices with the ability to browse the Internet has opened up interesting application areas for speech and natural language processing technologies. Voice search is one such application where speech technology is making a big impact by enabling people to access the Internet conveniently from mobile devices. Spoken queries are a natural medium for searching the Mobile Web, especially in the common case where typing on the device keyboard is not practical. Voice search is now recognized as a core feature of mobile devices and several applications [10, 11, 12, 13] have been developed. Generally, in such applications, a spoken query is automatically recognized and the Automatic Speech Recognition (ASR) 1-best hypothesis is sent to a text-based web search engine. Modeling the distribution of words in spoken queries offers different challenges compared to more conventional speech applications. The differences arise from the fact that the voice search application serves as a front-end to web search engines. Users typically provide the search engine with the keywords that will aid them in retrieving the information they are interested in. Spoken web queries, especially keyword style queries, are typically short and do not follow the syntax and grammar observed in other ASR tasks.

In this paper, we look at measures related to semantic relatedness between query terms as a way to improve the language model (LM) for voice search ASR systems. The semantic relatedness between the keywords of a spoken query stems from co-occurring together in the same web document or context even if the keywords are not necessarily adjacent and ordered in the same way as in the query. Our approach is thus based on the idea that if the ASR hypothesis terms tend to co-occur frequently in the searched corpus, the hypothesis is more likely to be correct.

Example: Here are two different hypotheses of a same utterance: *tobacco Road Austin* and *tobacco road author*. The second hypothesis is the correct transcription of the voice query. Under n-gram LM, the first hypothesis is more probable. However, *tobacco road* co-occurs more often with *author* and therefore, the second hypothesis is preferred by the proposed co-occurrence approach.

The co-occurrence models presented in this paper for the voice search task provide supplementary information to the conventional n-gram statistical LM. We present various types of co-occurrence constraints and scoring functions which capture different forms of semantic relationship between query terms.

We review some related work in Section 2. Then, in Section 3, we describe the co-occurrence based approach for n-best rescoring. Our experimental setup is described in Section 4. We comment on the computational requirements (time and memory) of our approach in Section 5. Finally, we conclude with a summary of our findings.

2. Related Work

Different approaches have been developed in order to improve language models for voice queries.

Li. et al. [6] propose an n-gram based machine translation model. However, this approach is limited to directory assistance while our approach is for general mobile web search.

A natural approach for spoken queries language modeling consists of exploiting a variety of search query logs to model spoken queries [7]. Chelba et al. [3] have showed some improvement in the accuracy when building LM on textual and manually transcribed queries. Query stream is normalized in order to address OOV issue and Katz smoothing is applied on the n-gram LM. Franz and Milch [5] propose to use words and collocations to build a uni-gram LM based on query logs. However, collocation captures only information about adjacent terms while co-occurrence is a larger notion that captures also information about terms that are not necessarily ordered and adjacent, but tend to co-occur in same documents or contexts. In much of the existing literature related to language modeling for voice search, large in-house query log search corpora are typically used. However publically available large corpora for search query logs are rare and in most cases difficult to collect from the Internet. The proposed approach does not rely on the availability of a search engine query log data and thus has a broader application.

In computational linguistics, co-occurrence is commonly used as an indicator of semantic relatedness for word sense

Copyright © 2011 ISCA

disambiguation [9]. In this work, we propose to extend co-occurrence paradigm to language modeling.

The proposed co-occurrence approach estimates exact co-occurrence scores for multiple terms in the ASR hypothesis using a search engine. Latent semantic analysis (LSA) based approaches [2], construct a low rank SVD based approximation for the term-document co-occurrence matrix and approximate the co-occurrence counts with a mixture of uni-gram probabilities (or topics). LSA based methods provide poor scalability for query data where the number of documents can easily exceed 100 million. Additionally, the search based approach can provide nearness based co-occurrence scores as presented in Section 3 where all the terms are required to be within a certain distance of each other. Such scores cannot be generated with a LSA based approach.

3. Co-Occurrence based n-best Rescoring

Given an utterance and the associated list of ASR n-best hypotheses, our approach consists of rescoring the different hypotheses using co-occurrence information. For each hypothesis, we estimate the frequency of the co-occurrence of its terms in the training data. Co-occurrence approach is not supposed to replace n-gram LM. Indeed, it is clear that for phrase search, n-gram LM based estimation will be more accurate. Co-occurrence approach supplements state of the art approaches based on acoustic and language modeling. Co-occurrence based scores are interpolated with acoustic model and n-gram LM based scores as an additional information source.

3.1. Co-Occurrence Semantics

We need to define a set of criteria in order to express co-occurrence relation between terms. For example, terms may be considered to co-occur if they appear in the same document or in the same local context. We have defined different query semantics in order to capture co-occurrence information at different levels. We review the different semantics from the weakest to the strongest:

- **Disjunction of hypothesis terms** (denoted OR): we search for the documents containing at least one term of the hypothesis.
- **Disjunction of conjunction of hypothesis terms** (denoted OR_n): we search for the documents containing at least n terms of the hypothesis, with n being a parameter.
- **Conjunction of hypothesis terms** (denoted AND): we search for the documents containing all the terms of the hypothesis.
- **Near search of hypothesis terms** (denoted $NEAR_n$): we search for documents containing all the terms of the hypothesis with a distance less than n terms between two hypothesis terms, with n being a parameter; however, the terms are not required to be ordered in the result as in the hypothesis. The distance between two terms is defined as the number of other terms from the document inserted between the two terms.
- **Phrase search of hypothesis terms** (denoted $PHRASE$): we search for the documents containing all the terms of the hypothesis as a phrase. The different hypothesis terms have to be adjacent and ordered in the relevant documents. Note that this approach is similar to the classical statistical n-gram LM approach without backoff and smoothing.

Note that under AND, $NEAR_n$ and $PHRASE$ semantics, all the hypothesis terms are required to appear in the result to be considered as relevant. Stop words are filtered for all the semantics except for $PHRASE$ in order to support exact phrase search. The hypothesis terms can appear in any order in the relevant documents except for $PHRASE$.

3.2. Co-Occurrence Scoring

Co-occurrence scoring functions are based on various estimates of the semantic relation between the different hypothesis terms in a corpus. These estimations stem from Information Retrieval (IR) theory [1]. The Term Frequency Inverse Document Frequency (*tf-idf*) of a term appearing in a document in a corpus is a classical statistical IR measure used to evaluate how important a term is to a document in a corpus. In our experiments, we have used the Lucene variation of *tf-idf*[1]. Different models based on and extending *tf-idf* have been proposed and Cohen et al. [4] have investigated the differences between some of them for text IR. The *tf-idf* scoring scheme is often used in the vector space model together with cosine similarity to measure the relevance of a document to a query or the similarity of two documents. We extend the *tf-idf* score to capture the co-occurrence information about an hypothesis in a corpus. We expect correct hypotheses to get higher co-occurrence score.

First, we introduce some notations:

- D: the set of documents in the corpus.
- $h = (t_0, ...t_k)$: the hypothesis h and its terms t_i.
- $h \in d$ means that the document d matches the hypothesis h.
- $\{d : h \in d\}_n$: the set of the top n documents matching the hypothesis h under *tf-idf* scoring.
- $n_d(t_i)$: the number of occurrences of the hypothesis term t_i in the document d.
- $n_d = \sum_{t_j \in d} n_d(t_j)$: the number of terms in the document d.
- $tf(t_i, d) = \frac{n_d(t_i)}{n_d}$: the frequency of the hypothesis term t_i in the document d.
- $idf(t_i) = 1 + log(\frac{|D|}{|\{d:t_i \in d\}|+1})$: the inverse document frequency of the hypothesis term t_i in the entire corpus D.
- $coord(h, d)$: factor based on how many of the hypothesis terms are found in the specified document d. More precisely, it is the ratio of the number of hypothesis terms that are matched in the specified document d over the total number of hypothesis terms.
- $hypNorm(h) = \frac{1}{\sqrt{\sum_{t_i} idf(t_i)^2}}$: the norm of the hypothesis h.
- $norm(d) = \frac{1}{\sqrt{n_d}}$: the norm of the document d.

We introduce now two different co-occurrence based scoring functions:

- **Document frequency**: the rescoring is based on the document frequency of the hypothesis, i.e., the number of documents matching the hypothesis in the corpus. The

[1] http://lucene.apache.org/java/3_0_3/api/core/org/apache/lucene/search/Similarity.html

document frequency of the hypothesis h in the corpus D is defined as follows:

$$DF(h, D) = \frac{|\{d : h \in d\}|}{|D|}$$

- **Term Frequency Inverse Document Frequency**: the rescoring is based on the sum of the *tf-idf* of the hypothesis terms over the top n matching documents. It is defined as follows:

$$hypNorm(h) \times \sum_{\{d:h \in d\}_n} coord(h, d)$$
$$\times norm(d) \times \sum_{i=0}^{k} \left(\sqrt{tf(t_i, d)} \times idf(t_i)^2 \right)$$

  It is denoted $TFIDF_n(h, D)$. This scoring function takes into account the number of documents matching the hypothesis (as DF scoring function) and also the frequency of the hypothesis terms in the matching documents.

In addition, we apply eventually some normalization of co-occurrence based scores. Here are the two different normalization we have applied:

- Number of documents matching at least one hypothesis term. In other terms, it is the DF of the hypothesis under OR semantics.
- Minimum document frequency among all the hypothesis terms.

Co-occurrence information is computed after stemming and stop word removal and applied in an n-best rescoring framework. We observe that n-best rescoring has no impact on utterances which contain only one hypothesis and utterances where all the hypotheses have the same Word Error Rate (WER). In addition, since the co-occurrence scores are computed after stemming, some of the utterances will have some identical hypothesis for co-occurrence score computation.

4. Experiments

4.1. Data Collection

The language model training corpora for our experiments comprises of the following: unsupervised transcripts for spoken web queries (**UNSUP**), data collected from the web (**WEBDT**), a street address corpus (**ADDRESS**), directory assistance data (**DA**), lists of common urls (**URL**), stock names (**STOCK**) and other in-house data (**OTHER**). The training corpora are in US English.

We report on an in-house test set for the voice search task. To date, no standardized test exists in the community to benchmark systems for the voice search task. However, similar tasks have been studied in the literature [3] where the baseline systems range in WER's from 16% to 19%. Our voice search test set has been collected on real users of an application for mobile web voice search. It contains 40K voice queries with 160K words. The n-best's for the test set generated using a state of the art ASR system have a total of 160K hypothesis with an average of 4 hypotheses per utterance. The baseline WER of the n-best is 15.66% and the oracle error rate is 11%.

4.2. Implementation

Our experiments were conducted using Lucene [2], an Apache open source library for indexing and searching written in Java [8]. Training data is indexed and co-occurrence scores are generated using query rewriting and search API.

The training corpora are stored in separate indices in order to allow tuning of parameters at corpus level. The corpora are stemmed before indexing using Snowball implementation[3]. Each stemmed term is stored in an inverted index along with its posting list that contains the following pieces of information: the different documents containing it and the positions within the document. Positions are stored in the index in order to support $PHRASE$ and $NEAR_n$ semantics. For corpora where document boundaries are not clearly marked, each sentence is indexed as a separate document. Note that a single index is built per corpus and it supports search under all the different semantics.

At search time, the hypotheses are reformulated in order to capture from the index co-occurrence information under the required semantics.

For each corpus, we report in Table 1 its size, the size of the associated Lucene search index and the average posting list length – estimated by the average number of documents containing a term.

corpus	corpus size (G)	index size (G)	posting list length
ADDRESS	6	20	15378
DA	0.5	2.5	1689
OTHER	0.05	0.23	219
STOCK	0.05	0.25	3902
UNSUP	0.01	0.07	96
URL	0.1	0.61	157
WEBDT	3.3	16.8	287

Table 1: Corpus and Index Analysis

4.3. WER Analysis

We report in Table 2 the lowest WER's obtained with different co-occurrence semantics interpolated with AM and LM scores. We denote by **ALL** the union of all the corpora. Weights of the linear interpolation were optimized using the Powell algorithm with n-best WER as the objective function with a training set of 4K utterances (10% of the query set). Best accuracy is achieved when estimating co-occurrence on all the corpora under $NEAR_{10}$ semantics. We notice a WER reduction from 15.66% to 15.31% that represents a relative reduction of 2% vs. the baseline.

4.4. Query Length Influence

We report in Table 3 some analysis on the relative WER reduction from the co-occurrence based approach according to the query length. We can note that co-occurrence helps more on short queries; indeed, we are not able to collect enough significant co-occurrence information on long queries especially under strong co-occurrence conditions.

In the above results, the WER baseline was estimated on transcripts generated with a pruned 4-gram LM which satisfies the decoding constraints of our baseline system. With a

[2] http://lucene.apache.org/

[3] http://snowball.tartarus.org/

corpus	semantics	scoring	WER
ALL	$NEAR_{10}$	$TFIDF_{100}$	15.34
ALL	OR_2	DF	15.41
WEBDT	$NEAR_{10}$	$TFIDF_{100}$	15.41
UNSUP	$NEAR_{10}$	$TFIDF_{100}$	15.45
OTHER	$NEAR_{10}$	$TFIDF_{100}$	15.47
WEBDT	$NEAR_{10}$	DF	15.47
URL	OR_2	DF	15.48
WEBDT	OR_2	DF	15.48

Table 2: Accuracy for different Co-occurrence Approaches on different Corpora

utt. length	distrib. (%)	baseline WER	co-occ. WER	relative WER reduction (%)
1	13	51.75	48.30	6.67
2	23	25.15	24.15	3.98
3	18	17.10	16.84	1.52
4	16	13.88	13.53	2.52
5	11	12.24	12.17	0.57
> 5	19	10.42	10.40	0.19

Table 3: Accuracy Improvement vs. Query Length using Co-occurrence

large unpruned n-gram LM, WER is reduced from 15.66% to 15.24%; however, most of the gains from the large language model are for long queries (more than 5 terms). Based on the accuracy improvement analysis presented in Table 3 we conducted an experiment where the co-occurrence score is used for short queries while unpruned LM is used for longer queries. This combination leads to a WER of 15.13%.

5. Computational Requirements

Experiments have been achieved on a computer with following characteristics: HP DL145 G2 4x AMD Opteron 2.20GHz, 8GB RAM.

If possible, the entire index is loaded to the RAM at the beginning of the process. In this case, no disk I/O is required during the search. Note that some RAM is also needed for some computations at search time. Index size is reported in Table 1. If the index cannot be loaded to the RAM, the posting list of each term is loaded to the RAM at search time. The average posting list length is reported in Table 1. It is a good estimation of the amount of data that has to be loaded to the RAM at search time per hypothesis term. In average, an hypothesis contains 3.2 terms, i.e., we need approximately 3.2 index lookups per hypothesis at search time.

We report in Table 4 the average system time in milliseconds for search per hypothesis at different sizes of RAM (1G, 3G and 7G). The system time is a good estimation of the actual process time; however, if the CPU is also running some other processes at the same time, the system time will be higher than the actual process time.

6. Conclusion

In this paper, we have presented an information retrieval based co-occurrence language model which provides significant improvement in WER on the voice search task over a state of the art system. We have explored various query semantics which differ in the nature of underlying co-occurrence constraints including weak disjunction constraints which consider the occurrences of any query term to strong conjunction constraints which require all the terms to be present in a certain context and order. Our results indicate that the co-occurrence model can provide improvements for the case of short keyword style queries which are difficult to model using more conventional language modeling techniques which rely on the structure of the word sequence. The improvements from the proposed co-occurrence model are thus complimentary to other approaches such as using large unpruned language models. We also provide insight into the computational requirements of the proposed approach in terms of both CPU time and memory usage.

corpus	1G	3G	7G
ADDRESS	139.75	137.7	137.16
DA	61.24	52	8.56
OTHER	0.43	0.42	0.42
SPELL	0.28	0.27	0.26
STOCK	0.19	0.18	0.17
UNSUP	0.72	0.68	0.68
URL	1.77	1.76	1.76
WEBDT	259.92	232.33	219.36

Table 4: Search time vs. RAM size

7. References

[1] Baeza-Yates, R. A. and Ribeiro-Neto, B., "Modern Information Retrieval", Addison-Wesley Publishing Company, 2008.

[2] Bellegarda, J.R., "Large vocabulary speech recognition with multispan statistical language models", IEEE Transactions on Speech and Audio Processing, 2000, volume 8, issue 1, pages 76–84

[3] Chelba, C., Schalkwyk, J., Brants, T., Ha, V., Harb, B., Neveitt, W., Parada, C. and Xu, P., "Query Language Modeling for Voice Search", Proceedings of the 2010 IEEE Workshop on Spoken Language Technology.

[4] Cohen, D., Amitay, E., and Carmel, D., "Lucene and Juru at TREC 2007: 1-Million Queries Track", Proceedings of TREC. 2007.

[5] Franz, A. and Milch, B., "Searching the Web by Voice", Proceedings of the 19th International Conference on Computational Linguistics (COLING), 2002, pages 1213–1217.

[6] Li, X., Ju, Y.-C., Zweig, G. and Acero A., "Language modeling for voice search: a machine translation approach", Proceedings of the 2008 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '08), pages 4913–4916.

[7] Li, X., Nguyen, P., Zweig, G. and Bohus, D., "Leveraging multiple query logs to improve language models for spoken query recognition", Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '09), pages 3713–3716.

[8] McCandless, M., Hatcher, E. and Gospodnetic O., "Lucene in Action, 2nd Edition", Manning Publications Co., 2009.

[9] Turney, P.D., "Word sense disambiguation by Web mining for word co-occurrence probabilities", In Proceedings of the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3), Barcelona, Spain, pages 239-242

[10] "Google voice search", http://www.google.com/mobile/voice-search.

[11] "Microsoft Tellme", http://www.microsoft.com/en-us/tellme/.

[12] "Nuance Dragon search", http://www.dragonmobileapps.com/-apple/search.html.

[13] "Vlingo", http://www.vlingomobile.com.