A SCALABLE METHOD FOR VOICE SEARCH TO NATIONWIDE BUSINESS LISTINGS

A. Moreno-Daniel[*], *B.-H. Juang*

Georgia Institute of Technology
Atlanta GA, 30332, USA

J. Wilpon

AT&T Labs – Research
Florham Park NJ, 07932, USA

ABSTRACT

Voice search or 411-service is the task that finds a ranked set of directory listings that match a spoken query, where the target entries in the listing database and the spoken query may differ moderately in their syntactic form. While the conventional paradigm uses a *two-box* input (location + name), a *single-box* paradigm to voice search can allow users to provide all the information in a single utterance, thereby increasing query efficiency. Furthermore, the scalability of traditional methods used in the two-box paradigm is infeasible, and alternative strategies that sacrifice accuracy are normally adopted. This work presents a scalable algorithm for directory search over a nationwide database of listings (millions of entries) without compromising recognition accuracy.

Index Terms— voice search, directory assistance, 411-service, spoken query processing, speech recognition, vForms

1. INTRODUCTION

Searching data by voice has been an active area of research since the early days of second generation ASR [1]. The degree of syntactic diversity in which a spoken query can invoke a target database entry ($d(e, E)$) spans a spectrum of related applications. As shown in Fig. 1, three main categories of application scenario can be identified: directory assistance (DA), voice search (VS) and spoken query processing (SQP).

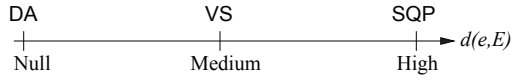


Fig. 1. Spectrum of related applications applications.

- DA: On the one extreme, directory assistance is the task with the longest history. In DA, as traditionally defined, fully constrained (FC) grammars can be used to anticipate the usually narrow syntactic formulations of queries in response to a rather rigid system prompt message (e.g. *name please*). Under this scheme, the "entry search" becomes an "acoustic search" of the path that best explains the observed spoken query within the ASR recognition network, therefore ASR and search form a single task.
- VS: In the center of the spectrum, voice search needs to handle moderate syntactic diversity. While spoken queries can be short and casual in format, text database entries tend to be lengthier and more formal. For example, the text entry: *Georgia Institute of Technology*, could be referred to as *Georgia Tech* in a spoken query. While an ontology-assisted scheme can certainly provide a mechanism that anticipates some of the alternate syntactic forms of a spoken query, unanticipated forms must be tolerated. Stochastic language models (SLMs), or grammars can tolerate discrepancies at the cost of an additional parsing layer that performs the search independently and treats ASR errors as typos. Therefore, ASR and search are traditionally disjoint tasks.
- SQP: On the other extreme of the spectrum, spoken query processing attempts to find database entries that are semantically (but not necessarily syntactically) close to the spoken query. Users may be oblivious of the textual content of a target entry or whether it indeed exists. Spoken queries tend to semantically describe what a relevant document might contain. It is the semantic gap between the spoken query and an entry what determines its relevance. While SQP remains the most interesting challenge in the spoken retrieval of documents, the absence of a proper corpus upon which a comprehensive evaluation can be performed has prevented this task from being fully explored [2].

The task we address in this work is *single-box* VS, a paradigm that lets users speak all the information in a single utterance (as opposed to the *two-box* paradigm that first asks for the city-state followed by the business name or category). Although FC grammars are optimum for modeling known simple languages, they are impractical for complex languages such as in single-box VS because of their rigidness and tremendous amount of memory required as the language grows. Methods currently used in the two-box paradigm are not scalable to the single-box case because of the large vocabulary in nationwide-databases.

This work presents a scalable algorithm for directory search over a nationwide database of listings (millions of entries) without severely compromising the recognition accuracy.

2. BACKGROUND

2.1. Traditional VS methods

There are *three key challenges* that VS methodologies must address: 1) anticipating the diverse syntactic forms in which each listing entry can be verbally queried; 2) modeling the spoken queries (language modeling over the spoken utterances as opposed to traditional well-formed text documents), and 3) searching for the relevant entry(-ies) with high efficiency.

One traditional method for VS is based on *signature grammars* [3] which automatically infers multiple query variants from the listing itself and constructs a FC grammar from them. While FC grammars indeed integrate the search and ASR into a single task (see Fig. 2-a), this scheme is not scalable to very large databases and it is fragile to queries that do not follow the enforced grammar rules. This has motivated statistical methods based on machine translation and word SLMs [4, 5]. Unlike FC grammars, word SLMs can recognize arbitrary word sequences at the expense of accuracy. As a result, the search process that finds the relevant entry(-ies) is deferred

[*]Performed part of this work while at AT&T's Summer internship.

to an independent post-ASR task (see Fig. 2-b), such as a vector space model (VSM) retrieval or a statistical retrieval [6], adding up to the CPU/memory consumption. Although the scalability issues with word SLM grammars are not as severe as with FC grammars, their large vocabulary (>100 K) still makes them CPU/memory intensive.

The strategy we adopted in this work combines ASR and search as an integrated task in a two-pass scheme (see Fig. 2-c) that is scalable to large databases such as nationwide business listings. This methodology is based on VFORMS [7], originally designed for DA. Section 2.2 presents a brief description of the principles, and Section 3.1 introduces VFORMS for VS.

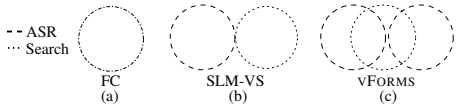


Fig. 2. Overlap of search and ASR tasks in VS paradigms.

2.2. VFORMS for DA

In DA applications, each database entry (or record) is organized in a multiplicity of fields. Inter-field information (constraints) can leverage ASR to improve its accuracy. Exploiting this information, however, is a task difficult to accomplish in scalable algorithms. The goal of VFORMS is to incorporate inter-field constraints into ASR and preserve scalability properties.

By design, VFORMS indexes (off-line) each database entry using sub-word units (e.g. phone n-grams), hence, we refer to these units as *phonedices* (π_k). In order to preserve scalability properties, VFORMS employs ASR as a lightweight operation (e.g. low CPU and memory requirements). The main components are depicted in Fig. 3 and the steps summarized as follows (see [7] for details).

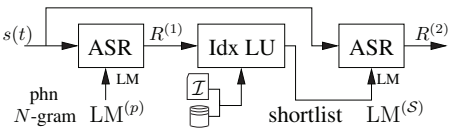


Fig. 3. VFORMS for DA.

1. **Indexing** (*off-line*): Assemble the reverse-index file $\mathcal{I}$ that tabulates the database entries containing each of the phonedices $\mathcal{I} = (\pi_k : \cup_{e^i \ni \pi_k} e^i)$.
2. **1st-pass recognition** (*on-line*): Use an n-gram phonotactic grammar $\mathrm{LM}^{(p)}$ to obtain the phone-lattice $R^{(1)}$ from the spoken query $s(t)$.
3. **Index access** (*on-line*): Use $\mathcal{I}$ to narrow down a shortlist $\mathcal{S}$ of entries that contain the phonedices recognized in $R^{(1)}$.
4. **2nd-pass recognition** (*on-line*): Use the top entries in $\mathcal{S}$ to dynamically assemble a FC grammar $\mathrm{LM}^{(\mathcal{S})}$, and do a second pass ASR to obtain the final recognition $R^{(2)}$.

The non-trivial syntactic diversity of queries in VS imposes an additional challenge.

3. METHODOLOGY

3.1. VFORMS for VS

The problem being addressed in this work is single-box VS to a database of nationwide business listings, a novel paradigm that differs from the traditional two-box VS (where location is first entered and then a business name). Single-box VS must be able to handle partial queries, i.e. those where no location (city/state) is present. This section describes how the proposed methodology addresses the aforementioned key challenges under this paradigm.

For the *first challenge*, we must devise a mechanism for converting the i-th listing entry (e^i) into a diverse set of plausible queries (E^i), which are syntactically distinct yet semantically equivalent:

$$e^i \rightarrow E^i = \bigcup_j \nu_j^i. \tag{1}$$

In order to infer such a set, we assume that the query formulation process invokes prior knowledge of the language ($\mathcal{G}$) and a vague notion of e^i to synthesize a simplified and coherent word sequence as shown in Fig. 4. One choice of model for $\mathcal{G}$ is a word n-gram: $\mathrm{LM}^{(\mathcal{G})}$.

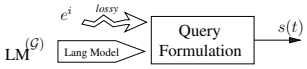


Fig. 4. A simple spoken query formulation model.

Let the i-th entry be formed by a sequence of words $e^i = (w_1^i, \ldots, w_{N_i}^i)$ and

$$\tilde{E}_a^i = \mathrm{Perm}_{N_i}^{N_i}(w_1^i, \ldots, w_{N_i}^i), \tag{2}$$

be the union of all $N_i!$ word permutations. The subset of word sequences in $\tilde{E}_a^i$ with likelihood greater than $p(e^i; \mathrm{LM}^{(\mathcal{G})})$ are selected as alternative valid word sequences for that entry:

$$E_a^i = \bigcup_{\tilde{e}^i \in \tilde{E}_a^i} 1\left(p(\tilde{e}^i; \mathrm{LM}^{(\mathcal{G})}) > p(e^i; \mathrm{LM}^{(\mathcal{G})})\right) \cdot \tilde{e}^i. \tag{3}$$

However, the number of words in the listing name (N_i) is often well beyond what a person would presumably speak in a query, thus $\tilde{E}_a^i$ can grow out of hand. In order to prevent this, we limit the length of the entry by choosing the word permutation with $N_{\max}$ words with maximum likelihood.

$$\tilde{E}_{\mathrm{clip}}^i = \mathrm{Perm}_{N_{\max}}^{N_i}(w_1^i, \ldots, w_{N_i}^i), \tag{4}$$

$$e_{\mathrm{clip}}^i = \underset{\tilde{e}^i \in \tilde{E}_{\mathrm{clip}}^i}{\mathrm{argmax}}\ p(\tilde{e}^i; \mathrm{LM}^{(\mathcal{G})}). \tag{5}$$

In the case $N_i!/(N_i - N_{\max})!$ (the size of $\tilde{E}_{\mathrm{clip}}^i$) is too large (hundreds), Eq. 4 can be limited to only sequences that preserve the original word order. The resulting sequence e_{clip}^i is used in place of e^i and passed to Eq. 2 where the word order is scrambled.

Furthermore, length diversity of the spoken queries ought to be anticipated. We generated shrunken versions of e^i using Eq. 4, for lengths shorter than $N_{\max}$, and selected those with likelihood greater than the original sequence (word-averaged) as valid. We denote this set as E_b^i. The final set, introduced by Eq. 1, becomes $E^i = E_a^i \cup E_b^i$.

The *second challenge* is to acquire prior knowledge about the language used in spoken queries. Under the VFORMS methodology, a phone n-gram language model $\mathrm{LM}^{(p)}$ is first used to generate the phone-lattice $R^{(1)}$. The use of a phone n-gram in VFORMS is key to maintain scalability properties, making ASR a lightweight process, capable of achieving high accuracy with low CPU and memory requirements. As the number of database entries grows, the memory required for a phone n-gram remains fixed, which is not the case for a word n-gram. Moreover, the inclusion rate of the correct sequence is higher for phone n-grams than for word n-grams.

Although it is reasonable to believe that a general purpose English phonotactic language model $\mathrm{LM}^{(p)}$ would be suitable for any task, the language used in VS has a lower perplexity than English in general. The high occurrence of a small set of words (state and city names, business categories, etc.) biases the phone statistics. Section 4.2 explores this issue experimentally.

The *third challenge* in VS pertains to finding the entries relevant to the query. A search process, in general, first finds the presence of index-terms in the query, and then locates the set of best matching entries. Conventionally, the domain of the index-terms is the set of *words* in the vocabulary (or some transformation of them). The recognition word best-path or word-lattice is then used to access a form of reverse-index file that returns the set of relevant entries. In this methodology, the domain of the indices is the set of *phone n-grams*, or phonedices (π_k).

Once a set of verbalizations is obtained ($E^i = \bigcup_j \nu_j^i$) for every e^i, the reverse index $\mathcal{I}$ maps every phonedex against a list of entries containing it:

$$\mathcal{I} = \bigcup_k \left(\pi_k : \bigcup_{\nu_j^i \ni \pi_k} (i,j) \right). \tag{6}$$

Upon phone-recognition, the set phonedices present in $R^{(1)}$ are extracted, along with their normalized expected path cost: $\mathcal{C}$ (negative log likelihood), and assembled in Q:

$$Q = \bigcup_{\pi_k \in R^{(1)}} (\pi_k, \mathcal{C}(\pi_k)), \tag{7}$$

which is used to access the reverse-index file $\mathcal{I}$, and to assemble the shortlist $\mathcal{S}$ which is a sorted list of the best-matching ν_j^i. The search process now becomes an acoustic search with a language model: $\mathrm{LM}^{(\mathcal{S})}$ that consists of:

$$\mathrm{LM}^{(\mathcal{S})} = \bigcup_{\nu_j^i \in \mathcal{S}} \nu_j^i. \tag{8}$$

With this methodology, we integrated search as part of the speech recognition process.

3.2. Implementation remarks

Typically, the databases accessed by VS exhibit some structure that allows entries to be represented at different levels of semantic granularity. At the finest level, each entry represents a distinct unit (or concept). At a coarser semantic level, however, entries are tagged into possibly overlapping broad categories. For example, a business could be uniquely identified as *Taco Express in Atlanta Georgia*, or by a broad category like *Mexican restaurants*, or just *restaurants*. As a heuristic, we assume that the location information, if any, is given at the end of the query. The actual entry becomes:

$$\begin{aligned} < e^i > \quad ::= \quad & < e^i_{\mathrm{name}} > [< e^i_{\mathrm{city}} >][< e^i_{\mathrm{state}} >] \mid \\ & < e^i_{\mathrm{cat}} > [< e^i_{\mathrm{city}} >][< e^i_{\mathrm{state}} >]. \end{aligned} \tag{9}$$

WFSA (weighted finite state automata) provide a well defined efficient and compact framework for representing recognition networks in ASR at different layers of abstraction (see [8] for details). Typically, the recognition network is formed by *composing* three WFSTs (weighted finite state transducers): C for the phone-context dependency, L for the word to phones mapping and G for the word-level grammar; thus the final network is called CLG. For VS, the same WFSA framework can be extended to include a higher layer of abstraction that represents the relevant entries. The fourth layer: S is a transducer that translates word sequences into database entry-IDs, forming the expanded VS network: $\mathrm{CLGS} = C \circ L \circ G \circ S$.

For those word sequences, such as *Mexican restaurants*, with multiple relevant entries, S yields a union of the corresponding entry-IDs, which could be later weighted by other criteria (ratings, distance, sponsorship, etc.).

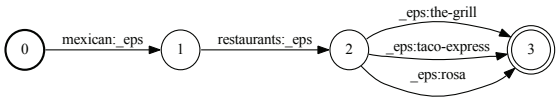


Fig. 5. Illustrative example of S transducer.

4. EXPERIMENTAL ANALYSIS

4.1. Case Study: Voice search for business listings

Inspired by YELLOWPAGES.COM™, the VS application scenario being studied is single-box verbal access to a nationwide directory of business listings[1]. A set of ten million nationwide business listings was collected. The information of each listing is organized in a multiplicity of fields including business name, category, telephone, address (street, city, state), etc. Additionally, a collection of ten million text queries was available from use logs of a traditional two-box application (business-name/category plus location). The vocabulary size of the directory was on the order of 850 K words.

A test set of 1430 spoken queries was recorded with an off-the-shelf smart-phone. The vocabulary size observed was 1108 distinct words, 994 distinct utterances and an average of 4.3 words per query. The content of such queries ranged from a single-word category to the name of a business and its location.

4.2. Phone recognition

This experimental analysis examines the complexity of the first-pass ASR in the proposed methodology (phone recognition). We used a generic US-English acoustic model in the following experiments (no device-specific models were used).

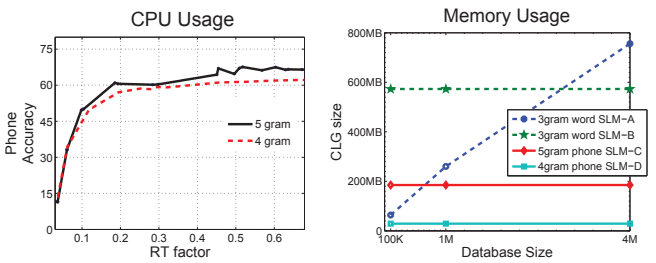
A total of six different phonotactic grammars $\mathrm{LM}^{(p)}$ were considered. We trained a 4-gram and a 5-gram with three separate data-sets, each with ten million samples: WP-LN (name+city+state white-pages listings), YP-LN (name+city+state yellow-pages listings) and YP-TQ (actual text queries from yellow-pages). Table 1 summarizes the perplexity found for these six phonotactic grammars. We observed that the 5-gram $\mathrm{LM}^{(p)}$ consistently has lower perplexity than its 4-gram $\mathrm{LM}^{(p)}$ counterpart, and that the phone statistics learned from the YP-TQ data-set were the closest to the ones observed in the test set, thus we selected this combination in the subsequent experiments.

	WP-LN	YP-LN	YP-TQ
phone 5-gram	18.20	14.27	10.25
phone 4-gram	26.99	19.34	13.82

Table 1. Perplexity of two phone n-gram LMs trained with ten million instances of WP-LN: white-pages listings, YP-LN: yellow-pages listings and YP-TQ: yellow-pages text queries.

The PER (phone error rate), shown in Fig. 6-a as 100 minus phone accuracy, converges to 33% at a real-time (RT) factor of 0.5 for a phone 5-gram LM, and to a PER of 37% for the phone 4-gram LM. Notice the knee is around 0.2–0.3 RT.

[1] The research being conducted is academic and exploratory, disconnected from any business or services provided by any company.



(a) (b)

Fig. 6. CPU (a) and memory (b) usage.

Under the proposed methodology, the CPU and memory usage of the phonotactic $\mathrm{LM}^{(p)}$ remain constant as the number of database entries (business listings) grows. Figure 6-b shows the size in megabytes of the CLG for two phonotactic grammars (labeled SLM-C and SLM-D), a word 3-gram LM (labeled SLM-B) trained with the same ten million YP-TQ text queries used to train these phone n-grams, and a conventional word 3-gram LM (labeled as SLM-A) trained with $\cup_i E^i$ (from Eq. 1) for different database sizes.

We observed that the memory required by SLM-C (phone 5-gram) is smaller than the one required by the traditional SLM-A (word 3-gram LM) for databases larger than 600 K entries; while SLM-D (phone 4-gram) is virtually always smaller than SLM-A. Notice how the memory required by SLM-A grows with the database size. The additional memory required by $\mathrm{LM}^{(\mathcal{S})}$ is negligible (kilobytes) as well as the time consumed to perform the second-pass recognition. Given that the reverse-index file $\mathcal{I}$ is read-only, a single copy can be loaded into memory and shared across multiple simultaneous VS sessions. Conventional reverse-index file management schemes are applicable.

4.3. Search

For these experiments, phonedices were set to phone-trigrams, the $\mathrm{LM}^{(p)}$ was set to SLM-C (phone 5-gram), and the auxiliary language model $\mathrm{LM}^{(\mathcal{G})}$ (used offline) was set to a word-bigram trained with the union of YP-TQ and YP-LN.

By design, vFORMS for VS recognizes only word sequences that have been inferred from the database entries ($\cup_i E^i$), therefore any query that targets an "out of directory" (OOD) business listings, e.g. an unregistered or a poorly expanded listing, will result in misrecognition. Confidence scores ought to be used to detect and handle the OOD cases. Three moderately sized random subsets of the business listings database were extracted (with 100 K, 1 M and 4 M entries). In order to prevent OOD from occurring, the corresponding listings of any OOD-queries were appended to the set of database entries ($\cup_i e^i$).

For comparison, we report the performance of two traditional single-pass ASR systems, each using a word 3-gram as language model: SLM-B (trained with ten million instances of YP-TQ) and SLM-A (trained with $\cup_i E^i$ for the three databases). Notice that while conventional word SLMs recognize word sequences that still need to be parsed and handled by a separate search system, the vFORMS-based method returns the recognized word sequence along with the set of relevant entry-IDs without the need of any further search.

Table 2 shows the word error rate (WER) and sentence error rate (SER) in the test set of spoken queries, for the SLM-A, SLM-B and the proposed vFORMS-based method.

Notice that the SER of vFORMS is consistently lower than the traditional word SLMs because vFORMS only recognizes entries that exist in the directory, while the SLM may recognize sequences that are not valid database entries. Bear in mind that although the SER is an ASR performance measure, it sets a lower bound on the search accuracy. For example, a SER of $N\%$ implies that the search result will be correct at least $100 - N\%$ of the time, because misrecognized sentences may still have the same semantics (e.g. *pizza/pizzeria*). The acoustic mismatch present in the recorded test set prevents error rates from dropping further at the moment, however, acoustic model adaptation techniques can be applied to improve the overall performance.

Method	Train Data	nr. of distinct ν_j^i	WER	SER
word SLM-B	YP-TQ	-	40.2%	74.4%
word SLM-A	$\bigcup_{i=1}^{100\,\mathrm{K}} E^i$	1 M	31.8%	45.4%
vFORMS			29.3%	40.4%
word SLM-A	$\bigcup_{i=1}^{1\,\mathrm{M}} E^i$	5.8 M	36.6%	53.4%
vFORMS			35.1%	49.2%
word SLM-A	$\bigcup_{i=1}^{4\,\mathrm{M}} E^i$	20 M	39.3%	59.4%
vFORMS			38.2%	55.9%

Table 2. Query word and sentence error rates (WER, SER).

5. CONCLUSIONS

This work presented a scalable two-pass methodology for voice search to a directory of nationwide business listings, based on the vFORMS methodology. Three key challenges of VS were identified and addressed. The memory requirements for the proposed method scales up for large nationwide databases (million of entries), without compromising accuracy with respect to conventional methods.

Future work will include an ontology inlet in $\mathrm{LM}^{(\mathcal{G})}$ to allow a richer expansion of semantically equivalent expressions for each listing. A confidence score that can help vFORMS detect OOD queries remains to be designed. Our current implementation starts the search only after $R^{(1)}$ has been obtained, which adds a small yet intrinsic delay. This could be mitigated by allowing a real-time construction of the shortlist $\mathcal{S}$. Additionally, computation can be further saved if part of the likelihood calculation performed in the first pass is reused in the second.

6. REFERENCES

[1] S. Furui, "50 years of progress in speech and speaker recognition," *Proc. SPECOM 2005*, pp. 1–9, 2005.

[2] A. Moreno-Daniel, J. G. Wilpon, B.-H. Juang, and S. Parthasarathy, "Towards the integration of automatic speech recognition and information retrieval for spoken query processing," *Proceedings of Interspeech*, Sep. 2008.

[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"," *Eurospeech*, pp. 1249–1252, Sept. 2003.

[4] X. Li, Y-C Ju, G. Zweig, and A. Acero, "Language modeling for voice search: A machine translation approach," *Intl. Conference on Acoustics, Speech and Signal Processing*, March 2008.

[5] Y. Wang, D. Yu, Y. Ju, and A. Acero, "An introduction to voice search," *IEEE Signal Processing Magazine*, vol. 25(3), pp. 28–38, May 2008.

[6] P. Natarajan, R. Prasad, R. M. Schwartz, and J. Makhoul, "A scalable architecture for directory assistance automation," *Intl. Conference on Acoustics, Speech and Signal Processing*, vol. 1, pp. 21–24, May 2002.

[7] S. Parthasarathy and A. Moreno-Daniel, "Directory retrieval using voice form-filling," *Intl. Conference on Acoustics, Speech and Signal Processing*, pp. IV:161–165, May 2007.

[8] M. Mohri, F. C. Pereira, and M. Riley, "Weighted finite state transducers in speech recognition," *Computer Speech and Language*, vol. 16, no. 1, pp. 69–88, 2002.