Retrieval and Browsing of Spoken Content

Ciprian Chelba, *Member, IEEE,* Timothy J. Hazen*, *Member, IEEE*
and Murat Saraçlar, *Member, IEEE,*

I. Introduction

Ever increasing computing power and connectivity bandwidth together with falling storage costs are resulting in an overwhelming amount of data of various types being produced, exchanged, and stored. Consequently, information search and retrieval has emerged as a key application area. Text-based search in particular is the most active area, with applications that range from web and private network search to searching the private information residing on one's hard-drive.

Speech search has received less attention perhaps because large collections of spoken material have previously not been available. However, with cheaper storage and increased broadband access, there has been a subsequent increase in the availability of online spoken audio content such as news broadcasts, podcasts, academic lectures, etc. A variety of government and commercial uses also exist (e.g., indexing of customer service calls). As data availability increases, the lack of adequate technology for processing spoken documents becomes the limiting factor to large-scale access to spoken content.

The existence of time-aligned transcriptions for spoken documents transforms the speech search problem into a text search problem. Unfortunately, manually transcribing speech is expensive and sometimes infeasible due to privacy concerns. This makes automatic approaches for indexing and searching spoken document collections very desirable. An ideal system would simply concatenate an automatic speech recognition (ASR) system with a standard text indexing and retrieval system. Unfortunately, today's speech recognition systems are not yet robust enough to produce high quality transcriptions for unconstrained speech audio in uncontrolled recording environments. Though commercial systems have been deployed for specialized applications (e.g., medical transcription), error rates for more difficult tasks (e.g, transcription of multi-person meetings) can easily be in the 30%-50% range using start-of-the-art ASR systems. Under these circumstances, inaccurate transcriptions can lead to errors in spoken document retrieval.

Ciprian Chelba is with Google, Mountain View, CA, 94043, USA.

Timothy J. Hazen is with MIT Lincoln Laboratory, Lexington, MA, 02420, USA.

Murat Saraclar is with Bogazici University Electrical and Electronic Eng. Dept., Bebek, 34342, Istanbul, TURKEY

*This work was sponsored by the Department of Defense under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

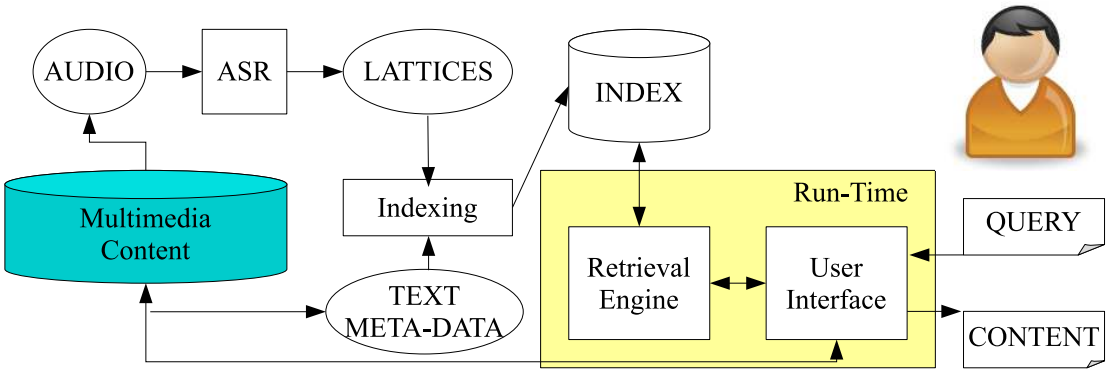

Fig. 1. A typical speech retrieval system.

In this paper, we strive to discuss the technical issues involved in the development of information retrieval systems for spoken audio documents, concentrating on the issue of handling the errorful or incomplete output provided by ASR systems. We focus on the usage case where a user enters search terms into a search engine (just as is done with text-based systems), and is returned a collection of spoken document *hits*. A typical system is depicted in Figure 1. This scenario has several difficult problems associated with it, which we can summarize with this set of questions:

- What data structures are needed to efficiently represent, index and search spoken content?
- How should speech search be conducted in the face of the errorful output of an ASR system?
- How can the retrieval process handle words that don't exist in the ASR system's vocabulary?
- How can the search integrate text-information and meta-data accompanying a speech file?
- What interface paradigm is needed for efficient search and browsing of speech files?
- How can speech search be evaluated to determine if its usefulness is approaching that of text search?

In this paper, we will attempt to provide an overview of current techniques for attacking these questions.

II. Automatic Speech Recognition

A. Probabilistic Framework

The core component of an information retrieval system for spoken audio documents is an automatic speech recognition (ASR) system for converting speech audio into linguistic hypotheses. Typically, the basic units of these linguistic hypotheses would be words. ASR systems generally employ a probabilistic modeling framework, in which the goal is to predict the most likely string of words W given the observed acoustic information A. Mathematically, the goal is to find the W which maximizes $P(W|A)$.

To model $P(W|A)$, a variety of simplifying assumptions must be made. To begin, words are typically decomposed into sequences of phonetic units (or *phones*) which represent the specific sounds used to distinguish between words (e.g., the word *cat* is comprised of the phones /k/, /ae/, and /t/). By applying Bayes Rule to $P(W|A)$ and decomposing the sequence of words W into a sequence of phonetic units U, the search for the best word hypothesis W given the acoustics A, can be expressed as follows:

$$\arg\max_W P(W|A) = \arg\max_W p(A|U)P(U|W)P(W) \quad (1)$$

In this expression, $p(A|U)$ is referred to as the acoustic model, $P(U|W)$ is referred to as the lexical pronunciation model, and $P(W)$ is referred to as the language model. There is a great deal of literature on the basic components of typical speech recognition systems, so we will only discuss the ASR modeling issues that are most relevant to the specific problems of audio information retrieval.

B. ASR Lattice Generation

Given an audio file and a set of models, an ASR system must apply the model constraints to the acoustic observations, and then search through the vast network of possible word sequences. Because the size of this search space is immense, it is generally pruned on-the-fly during the search to include only the most likely hypotheses. The network of unpruned hypotheses that have been explored can be maintained and saved for future use. These networks, often called speech recognition *lattices*, typically contain all of the word timing information and modeling scores used by the recognizer.

An example speech recognition lattice is shown in Figure 1. In this figure each arc in the network contains a word label along with the probability of that arc being taken from the previous state. The single best scoring sequence of words that can be traversed in a lattice is typically called the 1-best result. If desired, secondary searches of this lattice can be made to produce the N-best sentence hypotheses beyond the top scoring hypothesis.

C. Vocabulary and Language Model Adaptation

When building an ASR system for an information retrieval application, the choice of words in the system's vocabulary system is vital. ASR systems typically employ a closed-vocabulary, i.e., the vocabulary is predetermined before the speech is passed to the ASR system for recognition. If a word spoken in the audio is not present in the vocabulary of the recognizer, the recognizer will always misrecognize this word, and requests by users to locate documents containing this spoken word will necessarily fail.

Unfortunately, it is often the less common, topic-specific words which form the basis for information retrieval queries. Studies on a variety of data have shown that out-of-vocabulary (OOV) word rates

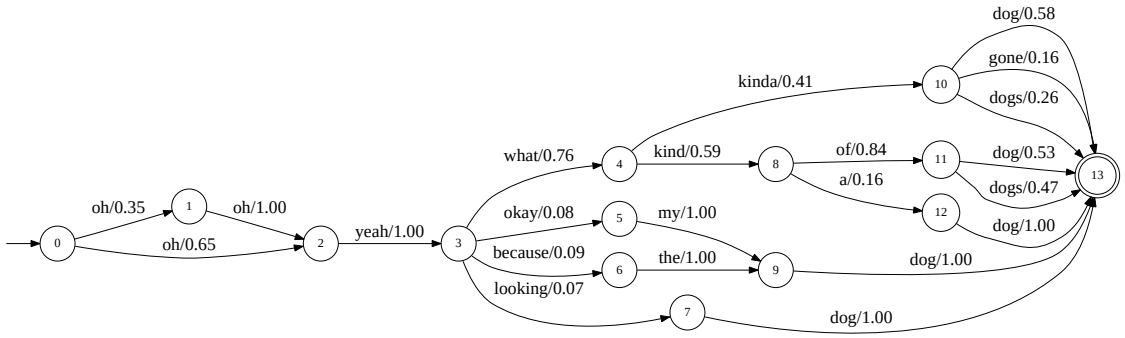

Fig. 2. Example speech recognition lattice.

for previously unseen data are not insubstantial even for large vocabularies [1]. The introduction of new vocabulary items (e.g., new proper names, scientific jargon, slang, etc.) in new audio data is also unavoidable. Thus, methods for countering the OOV problem in audio information retrieval are needed.

One potential method for countering the OOV problem is vocabulary adaptation. Commercial dictation software typically addresses this problem by providing the user with a topic-specific vocabulary and language model (e.g., medical transcription), and then allowing the user to add new vocabulary items on-the-fly as needed. In the absence of human supervision, it may still be possible to predetermine a topic-specific vocabulary and language model in some cases. For example, academic lectures could be classified into broad topics based on accompanying metadata such as the title of the academic subject.

In cases where the topic of the audio content is unknown, an automated solution to determining the topic might be possible. Under this scenario, the data is initially passed through a topic-independent recognizer. An analysis of the first pass recognition result is conducted to determine the subject matter of the audio. From this analysis, a topic-specific vocabulary and language model is created, and the new topic-specific recognizer performs a second recognition pass over the data to formulate a new set of recognition hypotheses. This approach can yield improved recognition accuracies at the cost of the additional computational cost of performing an additional recognition pass over the data.

D. Out-of-Vocabulary Word Modeling

Even with topic-specific vocabularies, OOV words are still possible. As a fall-back position, an ASR system can attempt to detect when an OOV word occurs then represent the OOV region by a sequence or network of phonetic units. This can be accomplished by building an independent OOV model from sub-word units (e.g., syllables or phones) that is examined in parallel with the existing vocabulary items during the ASR search, and hypothesizing an OOV word when the OOV model scores better than the

words in the known vocabulary [2]. The resulting search lattice can then be post-edited to replace any detected OOV word in the lattice with the underlying phonetic elements that represent it [3].

While word-based recognition is generally acknowledged to produce the most accurate information for audio indexing, another school of thought has suggested that the problem can be attacked by ignoring words entirely during the ASR search. Instead the ASR system would only produce a phonetic representation of the speech. Information retrieval of keywords would then be performed by locating audio files containing the phonetic sequences which match the pronunciations of the keywords [4]. This approach conceivably eliminates the ASR OOV problem, and the ASR engine can run with far greater efficiency (both in memory and speed) if it only has to produce phonetic hypotheses. However, the post-recognition indexing and retrieval stages necessarily become more complex under this approach.

III. Evaluation Metrics

Before discussing methods for speech retrieval, it is important to understand the problem and the method in which potential solutions will be evaluated. When discussing speech information retrieval applications, the basic scenario assumes that a user will provide a *query* and the system will return a list of rank-ordered *documents*. The query is generally assumed to be in the form of a string of text-based words. The returned documents are audio files purported by the system to be relevant to the query.

An extremely important aspect when developing such an application is the evaluation methodology. The obvious choice is to use human judges for annotating the degree of relevance of a document to a given query. Since the aim is to bridge the gap between text and speech search technology, one other possibility is to take as reference the output of a text retrieval engine that runs each query on the manually transcribed documents, rather than the spoken ones. Finally, and surely the most relevant is user satisfaction in a deployed system that is being actively used and improved.

Document retrieval performance can be evaluated via various metrics. Precision-recall rates and F-measure are relatively familiar metrics. Roughly speaking, precision is the fraction of returned documents from the collection that are relevant to the query, and recall is the fraction of relevant documents in the collection that are returned. These measures are calculated as follows: Given Q queries, let $R(q)$ be the total number of documents that are relevant to query q, $A(q)$ be the total number of retrieved documents and $C(q)$ be the number of relevant documents correctly retrieved. Then:

$$\text{Precision} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{C(q)}{A(q)} \qquad \text{Recall} = \frac{1}{Q}\sum_{q=1}^{Q}\frac{C(q)}{R(q)} \tag{2}$$

and $F = (2 \times \text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$.

Related metrics are Mean Average Precision (MAP), R-precision, and Precision-at-N [5]. We can roughly describe them as follows: for computing MAP, precision values are calculated at each of the recall values $(0.0, 0.1, 0.2, ..., 1.0)$ by traversing the ranked list of results from most to least relevant; the area under this P-R curve is then averaged over all queries in the test set. Precision-at-N is the precision measure of the top N returned documents (where $N = 10$ is a common choice). R-precision is similar to Precision-at-N, only that N varies for each given query q, and is set to the number of relevant documents $R(q)$. A shortcoming of this family of metrics is that they assume document relevance to be binary valued, which implies that all possible rankings of the relevant documents on the reference side are considered to be equivalent.

Because improved accuracy of the ASR system can lead to improved retrieval performance, metrics for evaluating the ASR system are also commonly examined. For example, the OOV rate for a recognizer's vocabulary on new previously unseen data is often measured. Also of particular interest to spoken retrieval systems is the query-OOV rate, measuring the rate at which query words provided by users are not covered by the vocabulary of the ASR system. The ASR system's accuracy is also typically monitored. The standard metric for evaluating ASR performance is the word error rate (WER) metric, which measures the percentage of errors an ASR system makes relative to the actual number of spoken words.

Another metric designed more specifically towards the system's ability to identify specific keywords in spoken content is the *actual term-weighted value* (ATWV) which is defined in the NIST Spoken Term Detection (STD) 2006 Evaluation Plan [6] as:

$$\mathrm{ATWV} = 1 - \frac{1}{Q}\sum_{q=1}^{Q}\{P_{miss}(q) + \beta P_{FA}(q)\} \tag{3}$$

where β is a user defined parameter (set to 1000 in the 2006 NIST STD evaluation), and where

$$\mathrm{P_{miss}}(q) = 1 - \frac{C(q)}{R(q)} \qquad \mathrm{P_{FA}(q)} = \frac{A(q) - C(q)}{T_{speech} - C(q)} \tag{4}$$

with T_{speech} being the total duration of the speech in the collection. Here the definitions of $R(q)$, $C(q)$, and $A(q)$ refer to the specific individual examples of the query phrase q in the audio data and not to whole documents, i.e. $R(q)$ is the total number of times the specific query phrase q actually appears (word for word) in the audio collection, $A(q)$ is the number of purported examples of q returned by the system, and $C(q)$ is the number of these purported examples of q that are actually correct. This metric specifically measures the system's ability to locate query keywords within audio files relative to perfect audio transcriptions.

IV. Previous Work

Many of the prominent research efforts aimed at spoken document retrieval (SDR) were centered around the SDR portion of the TREC evaluations conducted in 1999 and 2000 (a.k.a. TREC-8 and TREC-9) [5]. While the TREC-SDR evaluations mark a significant milestone in the speech retrieval field, a large body of SDR research existed previously, including notable contributions of [7] and [8]. Significant recent contributions have also been made, including [9], [10], [11], [12] and [13].

One problem encountered in work published prior or outside the TREC-SDR community is that it does not always evaluate performance from a document retrieval point of view — using a metric like MAP or similar — but rather uses word-spotting measures, which are more technology-centric rather than user-centric. Depending on the application, the document retrieval performance may be more important, whereas the word-spotting accuracy is an excellent indicator for how an SDR system might be improved.

The TREC-SDR evaluations focused on using Broadcast News speech from various sources: CNN, ABC, PRI, Voice of America. About 550 hrs of speech were segmented manually into 21,574 stories each comprising about 250 words on the average. The pre-existing approximate manual transcriptions (or closed captions for the video case) used for the SDR system comparison with text-only retrieval performance had fairly high WER: 14.5% for video and 7.5% for radio broadcasts. ASR systems tuned to the Broadcast News domain were evaluated on accurate manual transcriptions at 15%-20% WER, not far from the accuracy of the approximate manual transcriptions[1].

In order to evaluate the accuracy of retrieval systems, search queries (created based on general "topics") along with binary relevance judgments were compiled by human assessors for each of the 21,574 retrieval documents. SDR systems indexed the ASR 1-best output and their retrieval performance (measured in terms of MAP) was found to be flat with respect to ASR WER variations in the range of 15%-30%. The most significant finding was that no severe degradation in retrieval performance was observed when evaluating with the ASR outputs in comparison with the approximate manual transcriptions. As a result NIST's final report on the TREC-SDR evaluations declared the research effort "a success story" [5].

Having a common task and an evaluation-driven collaborative research effort represents a huge gain for the community, but there are shortcomings to the TREC-SDR framework: the speech recognizers were heavily tuned for the domain, leading to very good ASR performance. In general it is unrealistic to expect error rates in the 10-15% range, especially when decoding speech is mismatched to the training

[1]The distribution of errors in manual transcriptions or closed captions can be very different from the ASR errors, and may have a very different impact on retrieval performance.

data. It is not uncommon to observe WER values of 30% to 50%.

The large vocabulary ASR systems used in the TREC studies had very low OOV rates, typically below 1%. Since the queries were long and stated in plain English rather than using the keyword search scenario, the query-side OOV (Q-OOV) was very low as well, an unrealistic situation in practice. A study in [14] evaluates the effect of Q-OOV rate on retrieval performance by reducing the ASR vocabulary size so that the Q-OOV rate comes closer to 15%, a much more realistic figure since search keywords are typically rare words. The study showed severe degradation in MAP performance—50% relative, from 44 to 22.

The ability to effectively deal with OOV query words is an important issue. The most common approach is to represent both the query and the spoken document using sub-word units—typically phones or phone n-grams—and then match sequences of such units. The work in [4] shows the feasibility of sub-word SDR and advocates for tighter integration between ASR and IR technology. This approach was to index phone n-grams appearing in ASR N-best lists. This work also focused on Broadcast News speech, thus benefiting from good ASR performance. Similar conclusions are drawn by the excellent work in [15].

As pointed out in [16], word level indexing and querying is still more accurate and thus more desirable, were it not for the OOV problem. The authors argue in favor of a combination of word and sub-word level indexing. Another problem pointed out by the paper is the abundance of word-spotting false-positives in the sub-word retrieval case, somewhat masked by the MAP measure.

Similar approaches are taken by [17]; one interesting feature of this work is a two-pass system whereby an approximate match is carried out on the entire set of documents after which the costly detailed phonetic match is carried out on only 15% of the documents in the collection.

More recently [18] proposes an approach that builds an inverted index from ASR lattices—word or phone (sub-word) level—by storing the full connectivity information in the lattice; retrieval is performed by looking up strings of units. This approach allows for exact calculation of n-gram expected counts but more general proximity information (distance-k skip n-gram, $k > 0$) is hard to calculate.

The work in [15], [18] and [19] shows that making use of more than just the 1-best information—N-best lists or ASR lattices—improves retrieval accuracy significantly.

For browsing long spoken communications it becomes important to locate the portion that is actually relevant to the query. From the SDR point of view, this can be achieved by segmenting the documents into topics and locating topics. When dealing with spoken communications where these segments are short or when they consist of short utterances, the task becomes that of Spoken Utterance Retrieval (SUR) [18]. The purpose of SUR is to find all the utterances containing the query. Applications include browsing Broadcast News, telephone conversations, teleconferences [18] and lectures [3].

More recently, the NIST Spoken Term Detection (STD) 2006 Evaluation [6] introduces the task of locating the exact occurrence of a query in large heterogeneous speech archives — broadcast news, telephone conversations and roundtable meetings. The evaluation also included Arabic and Mandarin in addition to English and attracted many sites [20], [21], [22]. One notable technique is setting the detection thresholds in a term-specific fashion so as to maximize the ATWV metric [22].

V. Overview of Text Indexing and Search

A. Vector Space Models for Information Retrieval

Probably the most widespread model for text retrieval is the TF-IDF vector model [23]. For a given query $\mathcal{Q} = q_1 \dots q_i \dots q_Q$ and document D_j one calculates a similarity measure by accumulating the TF-IDF score $w_{i,j}$ for each query term q_i:

$$S(D_j, \mathcal{Q}) = \sum_{i=1}^{Q} w_{i,j}, \quad w_{i,j} = f_{i,j} \cdot idf_i \tag{5}$$

where $f_{i,j}$ is the term frequency (TF) of word q_i in document D_j, and $idf_i = \log \frac{N}{n_i}$ is the inverse document frequency (IDF), n_i/N being the fraction of documents containing q_i.

The main criticism to the TF-IDF algorithm is that the query terms are assumed to be independent. *Proximity information* is not taken into account at all. For example, the fact that the words LANGUAGE and MODELING occur next to each other in a given document is not used for relevance scoring, although the occurrence of the bigram LANGUAGE MODELING is more relevant than the combined occurrences of LANGUAGE and MODELING as unigrams. Moreover, the ability to evaluate proximity of query terms in the document becomes critical if one wishes to enhance the query language such that it allows phrase matching functionality, e.g. returning only documents that contain "LANGUAGE MODELING".

Another issue is that query terms may be encountered in different *contexts* in a given document: title, abstract, author name, font size, etc. For hypertext document collections even more context information is available: anchor text[2], various HTML sections of a given document being just a few examples. The TF-IDF algorithm completely discards such information although it is clearly important in practice.

B. Language Modeling Approach

One can rank using the "point-wise mutual information" between the query and some document:

$$S(D_j, \mathcal{Q}) = \log \frac{P(\mathcal{Q}, D_j)}{P(\mathcal{Q})P(D_j)} \;\propto\; \log P(\mathcal{Q}|D_j) \tag{6}$$

[2]Text describing the hypertext link pointing to the given document/web page.

This amounts to building a language model $P(W|D_j)$ from each document, and then using it to score the query, $P(\mathcal{Q}|D_j)$. If the language model is an n-gram with order higher than 1, then this solution will indeed take into account word adjacency, or even more general proximity features depending on the language modeling approach being used.

C. Early Google Approach

Aside from the use of PageRank (which is query independent and derived from the WWW connectivity graph), the early Google approach also uses both word *proximity* and *context* information heavily when assigning a relevance score to a given document, see [24], Section 4.5.1.

For each given query term q_i one retrieves the list of *hits* in document D; hits can be of various types depending on the *context* in which they occurred: title, anchor text, etc.; each type of hit has its own *type-weight*. For a single word query, the ranking algorithm takes the inner-product between the type-weight vector and a vector consisting of count-weights (tapered counts that discount the effect of large counts) and combines the resulting score with PageRank in a final relevance score. For multiple word queries, terms co-occurring in a given document are considered as forming different *proximity-types* based on how close the hits are, from adjacent to "not even close". Each proximity type comes with a proximity-weight and the relevance score includes the contribution of proximity information by taking the inner product over all types.

D. Inverted Index

Of essence to fast retrieval on static document collections of medium to large size is the use of an *inverted index*. The inverted index stores a list of hits for each word in a given vocabulary—the indexing terms.

For the TF-IDF model, the inverted index is the term-document co-occurrence matrix itself $(w_{ij})_{\substack{i=1,V\\j=1,D}}$. In the "early Google" approach, the hits are grouped by document; the list of hits for a given index term must include position—needed to evaluate counts of proximity types—as well as all the context information needed to calculate the relevance score of a given document using the scheme outlined previously; for details, the reader is referred to [24], Section 4.

The language modeling approach does not immediately lend itself to inverted indexing, and storing an n-gram language model for each document becomes prohibitively expensive for large collections of documents. However, the advantage over TF-IDF and other vector space retrieval techniques due to better use of proximity may become very important when sequencing of index terms is critical to good retrieval

performance, such as when using sub-word indexing units for being able to deal with OOV words. A good solution for storing a very large set of small n-gram models—one per document—would make this approach very appealing for many problems.

VI. SOFT INDEXING

As highlighted in the previous section, position information is taken into account when assigning relevance score to a given document. In the spoken document case however, we are faced with a dilemma. On one hand, using 1-best ASR output as the transcription to be indexed is suboptimal due to high WER, which is likely to lead to low recall—query terms that were in fact spoken are wrongly recognized and thus not retrieved. On the other hand, ASR lattices (Fig. 2) do have much better WER—[19] reports 1-best WER of 55% whereas the lattice WER was 30%—but the position information is not readily available: it is easy to evaluate whether two words are adjacent but questions about the distance in number of links between the occurrences of two query words in the lattice are hard to answer.

To simplify the discussion let's consider that a text-document hit for some word consists of (document id, position)—a pair of integers identifying the document and the position of the index term in the document, respectively. For speech content, the occurrence of a word in the lattice is uncertain and so is the position at which it occurs. However, the ASR lattice does contain the information needed to evaluate proximity information, since on a given path through the lattice we can easily assign a position index to each link/word. Each path occurs with some posterior probability, easily computable from the lattice, so in principle one could index *soft-hits* which specify the (document id, position, posterior probability) for each word in the lattice. A simple dynamic programming algorithm which is a variation on the standard forward-backward algorithm can be employed for performing this computation. The computation for the backward pass stays unchanged, whereas during the forward pass one needs to split the forward probability α_n arriving at a given node n according to the length l of the partial paths that start at the start node of the lattice and end at node n. For details on the algorithm and the resulting position specific posterior probability lattices (PSPL, see Fig. 3) the reader is referred to [19].

Soft-indexing for speech content could easily use other representations of the ASR lattices such as confusion networks (CN, see Fig. 4) developed by [25], where lattice links are approximately binned based on the time span of the link. Both approaches result in approximate word proximity and adjacency representations of the original lattice but have the advantage of compressing it. The PSPL representation guarantees that all N-grams present in the original lattice (with arbitrarily large N as allowed by the lattice) will also be present in the PSPL lattice; it is unclear whether this is true for the CN. Non-

0		1		2		3		4		5		6		7	
oh	1.0	**yeah**	.65	**what**	.46	**kind**	.27	dog	.26	EOS	.34	**EOS**	.44	EOS	.16
—		oh	.35	yeah	.35	what	.27	**of**	.23	**dog**	.29	dog	.09	—	
		—		because	.06	kinda	.19	kind	.16	dogs	.13	dogs	.06		
				okay	.05	the	.06	kinda	.11	of	.13	—			
				looking	.05	my	.05	dogs	.05	a	.03				
				—		dog	.05	EOS	.05	gone	.02				
						...	...	...	...	—					

Fig. 3. Position-specific posterior probability lattice derived from ASR lattice; similar to a text document, each "soft-token" (list of words with associated probability) occurs at some integer position in the document.

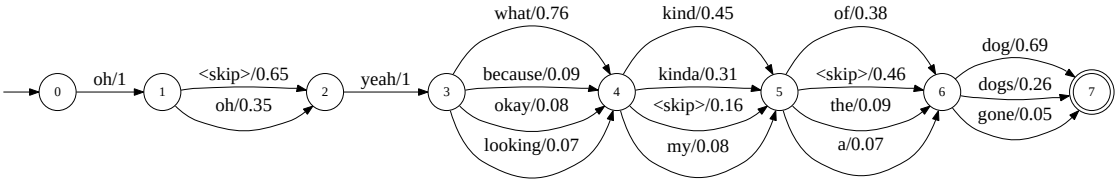

Fig. 4. Confusion network derived from ASR lattice.

emitting ϵ arcs may complicate the evaluation of proximity and adjacency features in a CN, in particular when using sub-word units such as phones. It is important to note that the manual transcription for the spoken content, as well as any text meta-data available can be represented as a lattice with a single path or equivalently a PSPL/CN with exactly one word per position bin and posterior probability 1.0.

Also noteworthy is that the PSPL/CN representation enables porting of any text information retrieval algorithm to the spoken document retrieval case: instead of simply counting the number of occurrences of various features in a given document, one now has to calculate expected counts according to the posterior probability distribution $P(W|A)$ as approximated by the PSPL/CN.

The effects of various approximations of hit proximity information—PSPL, CN, or other methods—deserve a more careful evaluation and comparison. The experiments in [19] show a 15-25% gain in MAP when indexing ASR lattices instead of 1-best output; similar results were reported in [15] and [18].

One aspect specific to soft-indexing—whether 1-best hypothesis with word-level confidence scores, or word alternates with posterior probabilities—is the ability to allow for different precision-recall operating points of the spoken document retrieval system, a feature that is not available when indexing only the 1-best word sequence; Fig. 5 shows a typical P/R curve. Indexing word alternates as opposed to 1-best increases the support of the posterior probability distribution and may be valuable in scenarios where recall is at a premium.

A. Indexing of Weighted Automata

Soft-indexing can also be implemented by representing lattices as weighted automata and building an index of all (or a restricted subset) of the possible substrings (called factors) contained in the automata [26]. Under this general framework, the index itself is a weighted automaton which allows efficient retrieval using string, and even regular expression queries. The procedure consists of turning each automaton into a transducer where the inputs are the original labels (e.g. words) and the outputs are the index labels (e.g. utterance number or position in an archive). Next, these transducers are combined by taking their union. The final transducer is determinized, resulting in optimal search complexity – linear in the length of the query string. The weights in the index transducer correspond to expected counts that are used for ranking.

B. Spoken Document Ranking in the Presence of Text Meta-data

Spoken documents rarely contain only speech. Often they have a title, author and creation date. There might also be a text abstract associated with the speech, video or even slides in some standard format. Saving hit context information (type of content where the hit occurred) emerges as a natural way to enhance retrieval quality: e.g., a hit in the title deserves different treatment compared to a hit in some other part of the document.

As mentioned above, PSPL/CN lattices can be used to represent text content as well, and consequently to naturally integrate the text meta-data in a unified search framework. As a data point, the experiments in [19] use a simple linear interpolation scheme for combining text meta-data and speech relevance scores. When placing all the weight on meta-data segments and ignoring the speech content, there is a significant drop in MAP performance; looking at it the other way, the performance gain obtained by adding speech content instead of only indexing text meta-data is 302% relative, despite the fact that the ASR system operates at about 50% WER. This dramatic improvement can be easily explained by the fact that the meta-data constitutes only about 1% of the amount of words in the transcription of speech content, a situation representative for many other scenarios.

VII. Sub-word Units

Words are the most natural and most common units used for ASR and retrieval. However certain limitations of word based systems have forced many researchers to investigate sub-word units both for recognition and retrieval. Although very large vocabulary ASR systems are becoming more common, typically the vocabulary is limited for practical reasons, as well as due to limited amount of training

data. Since query words are chosen to be discriminative, they tend to be low frequency words. This means that a typical query word is likely to be either an OOV word or a word for which the language model probability has not been well estimated due to data sparseness. Morphologically rich languages pose related challenges for ASR systems. For agglutinative languages such as Finnish and Turkish, it has been shown that sub-word based language models yield better accuracies than word based language models. In addition, sub-words obtained by morphological analysis or simply by stemming improve retrieval performance.

A wide range of sub-word units for language modeling and retrieval have been proposed, including linguistically motivated units such as phones, syllables and morphemes, as well as data driven units called multigrams, particles and morphs. For retrieval, indexing fixed length sequences of these sub-word units (typically phones) has also been proposed.

The smallest typical linguistic units are phonetic. These are the counterpart of using letters in text retrieval. Although using phones completely solves the OOV problem, the performance of phone recognition is inferior to word recognition even when the OOV rates are very high. This can be explained by the short acoustic duration of these units as well as by poor language model predictability. Syllables have been proposed since they are considered to be stable acoustically, but they still yield poor language models. Morphemes, being the smallest meaningful units, provide better language modeling but can be hard to distinguish acoustically. An alternative which proved successful in agglutinative languages groups all the suffixes in a word together to yield a stem-ending decomposition of a word. Stem-endings result in acceptable OOV rates while keeping acoustically long and distinguishable segments.

Linguistically motivated units require knowledge about specific languages and may be costly to extract, especially in the case of morphologic analysis. Data driven units are derived by utilizing statistical and information theoretic principles. Phone *multigrams* [4] are non-overlapping, variable-length, phone subsequences with some predefined maximum length. These are found using an unsupervised iterative algorithm maximizing the likelihood of the training data under the multigram language models. Similarly, *particles* [16] are selected in a greedy fashion so as to maximize the leave-one-out likelihood of a bigram language model. Statistical *morphs* [27] are based on the Minimum Description Length (MDL) principle, which means that in addition to the corpus representation given by the data likelihood, the lexicon representation is also taken into account.

A. Sub-word units for recognition, indexing and retrieval

Depending on the structure of the language, the amount of OOV words, and language model mismatches, ASR systems based on sub-word units may improve the recognition accuracy. In addition to improving ASR performance by decreasing or eliminating OOVs, in certain cases sub-word units can also be used solely at the indexing and search stage. Even when there is no improvement in accuracy, the additional information provided by the sub-word units is often beneficial for retrieval. In fact, when word based ASR is more accurate than sub-word based ASR, converting the output of word based ASR into sub-words improves the retrieval performance on OOV terms. This technique was shown to be effective for phone based indexing [16], where the phone index is obtained by converting the words in the lattice into phone sequences. At retrieval time, when an OOV query is encountered, the query is converted into a phone sequence and the phone index is used for retrieval. The conversion is performed using a grapheme-to-phoneme mapping module typically found in text-to-speech systems. The ASR system substitutes OOV words with similar sounding words which means that the phonetic sequence corresponding to the query may be present in the phone lattice used for indexing. In languages where homophones (words that sound the same but written differently) or homophonic word sequences (as in the classical example: recognize speech vs. wreck a nice beach) are common, using phonetic units in retrieval makes it possible to retrieve OOV words.

Using sub-words for indexing requires efficient methods for indexing and retrieving sequences. Proposed methods range from indexing fixed length subsequences such as triphones [28] to full indexing of lattices represented as weighted automata [26]. Some of these methods were explained in Section VI.

B. Query and Document Expansion for Speech Retrieval

Query and document expansion are techniques used in text based information retrieval to reduce the mismatch between the queries and documents. These techniques also have their counterparts in speech retrieval. One approach to the OOV problem is to expand the queries into similar in-vocabulary phrases [16]. The expansion utilizes a phone confusion matrix to represent the acoustic confusability between words. The selection is also guided by a language model so that reasonable phrases are chosen.

Stemming can also be considered as query expansion, in that words with the same root are considered equivalent. Query expansion might also use semantic similarity. For the case of speech retrieval, using alternate hypotheses in addition to the one-best hypothesis could be viewed as document expansion. These hypotheses may be represented as lattices or confusion networks. Similar to the query case, an expansion of these representations can be achieved by adding similar words.

C. Hybrid and Combination Methods

In many scenarios it is necessary to use both words and sub-word units for speech retrieval. The combination can be done at different stages of the process and using different strategies. Hybrid language models with both words and sub-words have been utilized with success for different tasks. These models can be structured or flat. In the structured case, the sub-word language model – used to model OOV words – is embedded in the word language model. In flat models, there is no embedding and the recognition units can be mixed arbitrarily. In both cases, the recognition output contains both words and sub-words.

Word based indexing and sub-word based indexing have different strengths and weaknesses. Word based approaches suffer from OOV words and as a result have lower recall. Sub-word based approaches result in higher recall at the expense of lower precision. Hence a combination of both methods yields the best performance. One way to achieve this is combined indexing resulting in a joint index [3]. Other strategies keep the word and sub-word indexes separate and use both for retrieval. When each index has a score associated with each entry, it is possible combine the results returned via score combination. However, this approach requires determining some parameters such as interpolation weights or normalization constants. A simpler and more effective approach is using word based and sub-word based retrieval in cascade. Since the word based retrieval is more accurate, the word index is the default. One cascade alternative (vocabulary cascade) uses the sub-word index only for OOV words, while another (search cascade) uses the sub-word index whenever word retrieval returns no answers. The latter was shown to be slightly better [18]. Figure 5 illustrates the effects of using lattices, sub-word units, and hybrid methods on various tasks and in terms of different metrics.

VIII. Browsing Search Results

While this paper has largely focused on the technology required to index, search, and retrieve audio documents, it is important not to overlook the final utility to the end user. For an application to be truly useful, the interface must enable users to search for and browse audio documents quickly and efficiently. One can imagine that an audio document search can be initiated in much the same way as a text search, i.e. the user enters a set of key words in a search field and is returned a set of putative *hits*. Unfortunately, unlike text, audio is a linear medium which is not easy to browse once the hits are returned. It would be highly inefficient for a user to have to listen to each hit in order to determine its relevance to his query.

To allow visual browsing, the interface could approximate text-based browsing by providing a snippet of the automatically transcribed speech produced by the ASR system. Even if ASR errors corrupt the

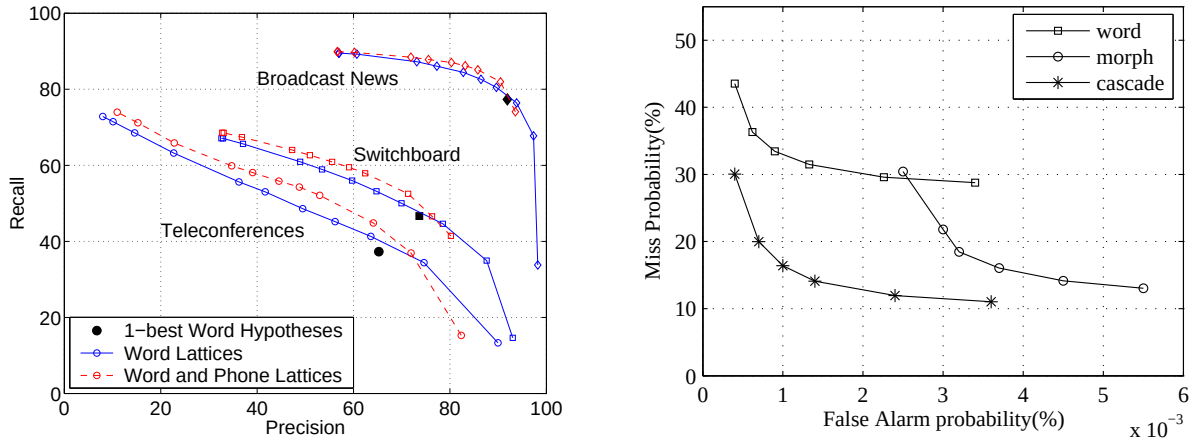

Fig. 5. Effects of using lattices, sub-word units, and hybrid methods on various tasks and in terms of different metrics. On the left is a precision v. recall curve on three English tasks with varying ASR accuracies showing the effectiveness of lattices and word/phone hybrids [18]. On the right is a miss v. false alarm curve on Turkish Broadcast News illustrating the use of words and sub-words as well as their combination [29].

transcription, users should be able to determine the relevance of the hit as long as the error rate of the transcription is not excessively high. Human comprehension of errorful transcripts is generally not degraded for lower error rates (i.e., $<= 15\%$)[30], and identification of the general topic is generally possible even for higher error rates. Comprehension can be aided by incorporating ASR word confidence information into the interface, i.e., highlighting confident words and graying out words with low confidence.

Once a relevant document is identified, it is important to provide the user with the capability to easily traverse the audio file in order to listen to the specific regions which are of interest. The ability to speed up or slow down an audio recording during playback can be helpful toward this end. At the very least the browser should allow the user to listen to the audio region immediately surrounding keywords hits.

For long audio files, it is also desirable to segment the file into shorter segments that represent specific topics of discussion within the file (e.g., different stories in a news broadcast). This would allow users to jump to the start of relevant audio segments and not just to the points where specific keywords occur.

As an example interface, Figure 6 shows a screen shot of the MIT Lecture Browser, a system designed to allow searching and browsing of academic lectures recorded at MIT [13]. This browser was designed to provide users with a range of methods to efficiently search for and browse through lectures. The browser enables the user to type a text query and receive a list of hits contained within the indexed lectures. Queries can be constrained by allowing users to specify a topic category from a pull-down menu before searching. An automatically derived segment structure for each lecture is displayed graphically as a series

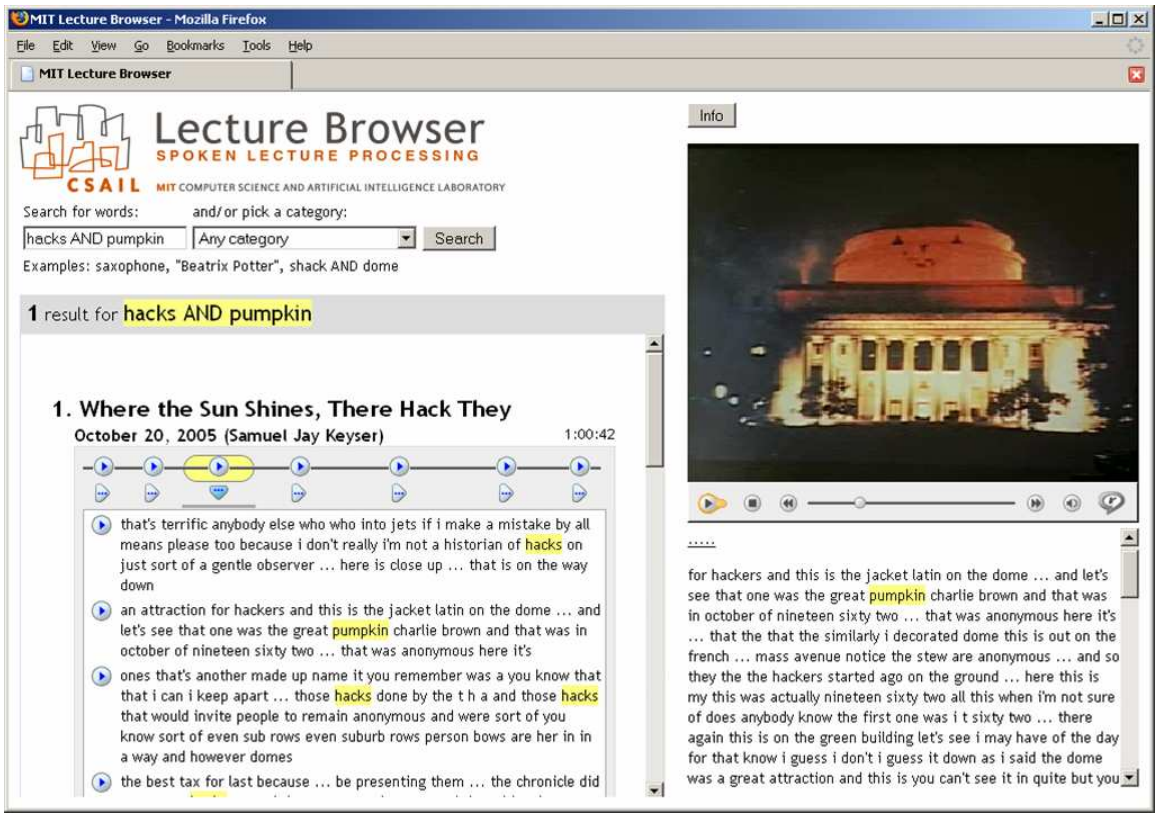

Fig. 6. A screen shot of the MIT Lecture Browser, available at http://web.sls.csail.mit.edu/lectures.

of *Play* buttons along a time line, with segments containing query word hits highlighted in yellow. The individual query word hits within each segment can be displayed together with their surrounding context in the transcript. The user can play the video starting at any displayed word, utterance, segment, or lecture that is shown on the screen. Accompanying the streaming video is a scrolling window displaying the synchronized text transcript. Individual words in the transcript are underlined as they are played, providing easier access for hearing-impaired users. The user can also scroll the text transcript window and begin playing the video starting from any specific word.

IX. Conclusions

In this paper we have examined recent research efforts in the areas of spoken document processing for retrieval and browsing. To conclude this paper, we wish to highlight a few key points.

First, experimental evidence shows that using the spoken content in an audio collection provides a significant improvement in performance with respect to using only accompanying text meta-data for retrieval, even when the word-error-rate is as high as 50%. The audio content and text meta-data can

also be used jointly for further improvements in retrieval performance.

Second, soft-indexing provides better retrieval performance than using the 1-best ASR output in scenarios with high WER. In such cases the 1-best document representation has in fact high variance and by taking into account the confidence of the ASR system in its output, as well as the possible alternatives, the ASR lattice better represents the document content. Soft hits also allow the precision-recall operating point to be adjusted to the needs of a given application or user.

Next, a variety of methods for handling the OOV word problem using sub-word units have demonstrated improved retrieval performance, in particular when they are used in conjunction with existing word-based methods. This remains an active area of research, and new techniques will hopefully continue to surface.

Finally, though system developers often focus on the technical issues of performing accurate search and retrieval, it is vital not to overlook the importance of the user interface. Only through a careful integration of speech search technology with user-friendly interface design will end-to-end systems actually allow users to efficiently search for, retrieve and browse audio content. Important for this aim is the adoption of an evaluation framework that assesses the ability of users to achieve their objectives.

REFERENCES

[1] I. Hetherington, "A characterization of the problem of new, out-of-vocabulary words in continuous-speech recognition and understanding," Ph.D. dissertation, Massachusetts Institute of Technology, 1995.

[2] I. Bazzi, "Modelling out-of-vocabulary words for robust speech recognition," Ph.D. dissertation, Massachusetts Institute of Technology, 2002.

[3] T. Hori, I. Hetherington, T. Hazen, and J. Glass, "Open-vocabulary spoken utterance retrieval using confusion networks," in *Proc. ICASSP*, 2007, pp. 73–76.

[4] K. Ng, "Subword-based approaches for spoken document retrieval," Ph.D. dissertation, Massachusetts Institute of Technology, 2000.

[5] J. Garofolo, G. Auzanne, and E. Voorhees, "The TREC spoken document retrieval track: A success story," in *Proc. of the Recherche d'Informations Assiste par Ordinateur: Content Based Multimedia Information Access Conference*, 2000.

[6] NIST, "The spoken term detection (STD) 2006 evaluation plan," http://www.nist.gov/speech/tests/std/, 2006.

[7] M. Brown, J. Foote, G. Jones, K. Jones, and S. Young, "Open-vocabulary speech indexing for voice and video mail retrieval," in *Proc. ACM Multimedia 96*, 1996, pp. 307–316.

[8] D. James, "The application of classical information retrieval techniques to spoken documents," Ph.D. dissertation, University of Cambridge, Downing College, 1995.

[9] J. V. Thong, P. Moreno, B. Logan, B. Fidler, K. Maffey, and M. Moores, "SpeechBot: An experimental speech-based search engine for multimedia content on the web," *IEEE Trans. on Multimedia*, vol. 4, no. 1, pp. 88–96, March 2002.

[10] S. Whittaker, J. Hirschberg, B. Amento, L. Stark, M. Bacchiani, P. Isenhour, L. Stead, G. Zamchick, and A. Rosenberg, "SCANMail: A voicemail interface that makes speech browsable, readable and searchable," in *Proc. SIGCHI Conference on Human Factors in Computing Systems*, 2002, pp. 275–282.

[11] D. W. Oard, D. Soergel, D. Doermann, X. Huang, G. C. Murray, J. Wang, B. Ramabhadran, M. Franz, S. Gustman, J. Mayfield, L. Kharevych, and S. Strassel, "Building an information retrieval test collection for spontaneous conversational speech," in *Proc. SIGIR*, 2004, pp. 41–48.

[12] J. Hansen, R. Huang, B. Zhou, M. Seadle, J. Deller, A. Gurijala, M. Kurimo, and P. Angkititrakul, "SpeechFind: Advances in spoken document retrieval for a national gallery of the spoken word," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 712–730, September 2005.

[13] J. Glass, T. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," in *Proc. Interspeech*, 2007, pp. 2553–2556.

[14] P. Woodland, S. Johnson, P. Jourlin, and K. Jones, "Effects of out of vocabulary words in spoken document retrieval," in *Proc. of SIGIR*, 2000, pp. 372–374.

[15] M. Siegler, "Integration of continuous speech recognition and information retrieval for mutually optimal performance," Ph.D. dissertation, Carnegie Mellon University, 1999.

[16] B. Logan, J. V. Thong, and P. Moreno, "Approaches to reduce the effects of OOV queries on indexed spoken audio," *IEEE Trans. on Multimedia*, vol. 7, no. 5, pp. 899–906, October 2005.

[17] P. Yu, K. Chen, C. Ma, and F. Seide, "Vocabulary-independent indexing of spontaneous speech," *IEEE Trans. on Speech and Audio Processing*, vol. 13, no. 5, pp. 635–643, September 2005.

[18] M. Saraclar and R. Sproat, "Lattice-based search for spoken utterance retrieval," in *Proc. HLT-NAACL*, 2004, pp. 129–136.

[19] C. Chelba, J. Silva, and A. Acero, "Soft indexing of speech content for search in spoken documents," *Computer Speech and Language*, vol. 21, no. 3, pp. 458–478, July 2007.

[20] D. Vergyri, I. Shafran, A. Stolcke, R. Gadde, M. Akbacak, B. Roark, and W. Wang, "The SRI/OGI 2006 Spoken Term Detection System," in *Proc. Interspeech*, 2007, pp. 2393–2396.

[21] J. Mamou, B. Ramabhadran, and O. Siohan, "Vocabulary independent spoken term detection," in *Proc. SIGIR*, 2007.

[22] D. R. H. Miller, M. Kleber, C. Kao, O. Kimball, T. Colthurst, S. Lowe, R. Schwartz, and H. Gish, "Rapid and accurate spoken term detection," in *Proc. Interspeech*, 2007.

[23] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. New York: Addison Wesley, 1999, ch. 2, pp. 27–30.

[24] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[25] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: Word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373–400, October 2000.

[26] C. Allauzen, M. Mohri, and M. Saraclar, "General indexation of weighted automata - application to spoken utterance retrieval," in *Proc. HLT-NAACL 2004 Workshop on Interdiciplinary Approaches to Speech Indexing and Retrieval*, 2004.

[27] V. Turunen and M. Kurimo, "Indexing confusion networks for morph-based spoken document retrieval," in *Proc. SIGIR*, 2007.

[28] S. Dharanipragada and S. Roukos, "A multistage algorithm for spotting new words in speech," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 8, pp. 542–550, November 2002.

[29] S. Parlak and M. Saraclar, "Spoken term detection for Turkish broadcast news," 2008, submitted to ICASSP.

[30] K. Bain, S. Basson, A. Faisman, and D. Kanevsky, "Accessibility, transcription and access everywhere," *IBM Systems Journal*, vol. 44, no. 3, pp. 589–603, 2005.