A Lattice-Based Approach to Query-by-Example Spoken Document Retrieval

Tee Kiah Chia† Khe Chai Sim‡
†Department of Computer Science
National University of Singapore
Law Link, Singapore 117590
{chiateek,nght}@comp.nus.edu.sg

Haizhou Li‡ Hwee Tou Ng†
‡Institute for Infocomm Research
21 Heng Mui Keng Terrace
Singapore 119613
{kcsim,hli}@i2r.a-star.edu.sg

ABSTRACT

Recent efforts on the task of spoken document retrieval (SDR) have made use of speech lattices: speech lattices contain information about alternative speech transcription hypotheses other than the 1-best transcripts, and this information can improve retrieval accuracy by overcoming recognition errors present in the 1-best transcription. In this paper, we look at using lattices for the query-by-example spoken document retrieval task – retrieving documents from a speech corpus, where the queries are themselves in the form of complete spoken documents (query exemplars). We extend a previously proposed method for SDR with short queries to the query-by-example task. Specifically, we use a retrieval method based on statistical modeling: we compute expected word counts from document and query lattices, estimate statistical models from these counts, and compute relevance scores as divergences between these models. Experimental results on a speech corpus of conversational English show that the use of statistics from lattices for both documents and query exemplars results in better retrieval accuracy than using only 1-best transcripts for either documents, or queries, or both. In addition, we investigate the effect of stop word removal which further improves retrieval accuracy. To our knowledge, our work is the first to have used a lattice-based approach to query-by-example spoken document retrieval.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*retrieval models*; I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*speech recognition and synthesis*

General Terms

Algorithms, Experimentation, Performance, Theory

SIGIR'08, July 20–24, 2008, Singapore.

1. INTRODUCTION

In a query-by-example information retrieval task, one is given a collection of documents, and a query which is itself a full-fledged document – a query *exemplar* – and the task is to find documents in the collection which are similar in subject matter to this exemplar. When both the query and the document collection are in the form of speech recordings, one obvious way to perform query-by-example retrieval is to run automatic speech recognition (ASR) on the recordings to obtain 1-best transcripts for both queries and documents, and use these transcripts for retrieval. However, this approach suffers from two problems:

- The 1-best transcripts are likely to contain recognition errors, especially when the speech data are of a conversational or noisy nature.
- The query exemplars will contain lots of non-content words which may interfere with the retrieval process.

To overcome the problem of recognition errors, one way is to work not with only one transcription hypothesis for each utterance, but several hypotheses presented in a *lattice* data structure. A lattice is a connected directed acyclic graph in which each edge is labeled with a term hypothesis and a likelihood value[11]; each path through a lattice gives a hypothesis of the sequence of terms spoken in the utterance.

Since the information in a lattice has a statistical interpretation, a retrieval model based on statistical inference, such as the statistical modeling retrieval approach of Song and Croft[28], will seem to be a more natural and more principled approach to lattice-based retrieval. We thus extend the statistical lattice-based retrieval method of Chia et al.[7], which has been shown to work well for retrieving Mandarin Chinese conversational speech with written keyword queries, to the query-by-example task. In our method, we generate a lattice for each speech segment in the collection corpus and the query exemplars, and compute the *expected word count* – the mean number of occurrences of a word given a lattice – for each word in each lattice. Using these expected counts, a statistical language model is estimated for each spoken document and each query, and a document's relevance to a query can then be computed as a Kullback-Leibler divergence between the document model and query model[15]. To mitigate the problem of noise in the retrieval process caused by non-content words in queries, we perform stop word removal, or stopping, which is commonly used in information retrieval (IR) tasks.

The rest of this paper is organized as follows. In Section 2 we review related work in the areas of spoken document

retrieval, IR in general, use of spoken queries, and query by example. Section 3 describes statistical retrieval methods for performing query by example using 1-best transcripts and lattices, the details of the experimental setup for comparing the two methods, and the results. Experiments with stop word removal using different stop lists are described in Section 4. Finally, Section 5 concludes our discussions and outlines our future work.

2. RELATED WORK

2.1 Speech in IR

2.1.1 Lattices for Spoken Document Retrieval

Spoken document retrieval (SDR) with short textual queries is a relatively well-studied task. A straightforward way to perform SDR is to use the 1-best ASR transcripts of spoken documents for retrieval – indeed, in the SDR track of the Ninth Text REtrieval Conference (TREC-9), in which participants were required to perform retrieval from a collection of news broadcasts, it was found that "for each of the participants, retrieval from the transcripts created by their own recognizer was comparable to the retrieval from the human reference transcripts"[30]. Despite this, it is opined that 1-best transcripts are still not of sufficient quality when the speech are of a more challenging nature, and in such cases the WER can be 50% or higher[25, 19]. For such conditions, the use of lattices has been found to be useful for improving retrieval accuracy.

Lattices were first introduced by James and Young[12] as a representation for indexing spoken documents, as part of a method for vocabulary-independent keyword spotting. The lattice representation was later applied to the task of spoken document retrieval by James[11]: James counted how many times each query word occurred in each phone lattice with a sufficiently high normalized log likelihood, and these counts were then used in retrieval under a vector space model with $tf \cdot idf$ weighting. Jones et al.[14] combined retrieval from phone lattices using variations of James' method with retrieval from 1-best word transcripts to achieve better results.

Since then, a number of different methods for SDR using lattices have been proposed. For instance, Siegler[26] used word lattices instead of phone lattices as the basis of retrieval, and generalized the $tf \cdot idf$ formalism to allow uncertainty in word counts. Saraclar and Sproat[25] performed word-spotting in word lattices by looking for query word occurrences whose expected counts were above a certain threshold; they also computed the expected count of the occurrences of query word pronunciations, when using phone lattices for word-spotting. Chelba and Acero[5] preprocessed lattices into more compact Position Specific Posterior Lattices (PSPL), and computed an aggregate score for each document based on the posterior probability of edges and the proximity of search terms in the document; the PSPL representation was further refined by Zhou et al.[36]. Mamou et al.[19] converted each lattice into a word confusion network[20], and estimated the inverse document frequency (idf) of each word t as the ratio of the total number of words in the document collection to the total number of occurrences of t. Hori et al.[10] combined confusion networks for words and phones to do open-vocabulary word-spotting. Chia et al.[7] performed lattice-based retrieval using a statistical retrieval framework[28], instead of the vector space model with $tf \cdot idf$ weighting.

2.1.2 Lattices of Spoken Queries for IR

The use of lattices of spoken queries for IR was studied by Colineau and Halber[8], who measured the precision and recall for extracting keywords from queries in speech form. In their task, the queries were specified as short natural language sentences, such as "What do you have on John Kennedy?"; these were recognized and transcribed by applying grammar parsing, followed by rescoring with a domain-specific n-gram model, onto a word lattice. While this method is useful for short queries which are relatively formulaic and specify the information need concisely, it is not directly applicable to our task, where the query exemplars comprise unrestricted speech and may contain much non-topical material; a different approach is thus needed in our case.

2.2 General IR Techniques

2.2.1 Retrieval via Statistical Language Modeling

The statistical language modeling approach to retrieval was used by Ponte and Croft[22] for IR with text documents, and it was shown to outperform the $tf \cdot idf$ approach for this task; Song and Croft[28] improved on this method and framed the problem of relevance ranking as a query likelihood computation. Lafferty and Zhai[15] showed that ranking by query likelihood is equivalent to ranking according to the Kullback-Leibler divergence between query and document models.

2.2.2 Effect of Different Stop Word Lists

Stop word removal, being a standard IR technique, is often simply treated as a given. However, studies have been conducted on how different stop lists can impact performance. Sinka and Corne[27] performed classification and clustering of web documents in English using two different stop lists, and also formulated a method for deriving stop lists using word entropies. Carvalho et al.[4] compared the result of stopping using three different stop lists, for the task of Portuguese question answering.

2.3 Query by Example

In the area of query by example, Chen et al.[6] used newswire (text) articles as query exemplars for retrieving news broadcast (speech) recordings; for retrieval, they used the statistical language modeling approach with 1-best transcripts of the collection corpus. The tracking task in the Topic Detection and Tracking (TDT) project has also been viewed as a form of query by example[32]. Efforts to solve this task – for instance, He et al.[9] and Lo and Gauvain[16, 17] – have made use of only ASR transcripts; this is likely because topic tracking, like SDR, can be done accurately even with 1-best transcripts when the rate of transcription errors is low enough[1].

2.4 Contributions of Our Work

The contributions of our work are as follows. First, we extend the use of the statistical model in lattice-based SDR to the query-by-example task. Our work can also be considered an extension of Chen et al.'s work on query-by-example[6] to spoken queries, and an extension of Lafferty and Zhai's formulation of statistical IR as Kullback-Leibler divergence[15]

to the use of expected counts from speech lattices. In addition, we study the effect of stop word removal using different stop lists in the context of query-by-example SDR.

3. RETRIEVAL METHODS

We now describe two retrieval methods for performing query-by-example SDR: a baseline statistical method which works on 1-best transcripts of both queries and documents, and a statistical method which is capable of working with lattice statistics of queries and documents.

3.1 Baseline Statistical Retrieval Method

3.1.1 Retrieval Model

Our baseline retrieval method is motivated by Song and Croft[28], and uses the language smoothing methods of Zhai and Lafferty[35]. This method is used to perform retrieval on the documents' 1-best ASR transcripts and reference human transcripts.

Let $\mathcal{C}$ be the collection of documents to retrieve from. For each document $\mathbf{d}$ contained in $\mathcal{C}$, and each query $\mathbf{q}$, the relevance of $\mathbf{d}$ to $\mathbf{q}$ can be defined as $\Pr(\mathbf{d} \mid \mathbf{q})$. This probability cannot be computed directly, but under the assumption that the prior $\Pr(\mathbf{d})$ is uniform over all documents in $\mathcal{C}$, we see that

$$\Pr(\mathbf{d} \mid \mathbf{q}) = \frac{\Pr(\mathbf{q} \mid \mathbf{d}) \Pr(\mathbf{d})}{\Pr(\mathbf{q})} \propto \Pr(\mathbf{q} \mid \mathbf{d});$$

this means that ranking documents by $\Pr(\mathbf{d} \mid \mathbf{q})$ is equivalent to ranking them by $\Pr(\mathbf{q} \mid \mathbf{d})$, and thus $\Pr(\mathbf{q} \mid \mathbf{d})$ can be used to measure relevance[2]. More precisely, we use as our relevance score the logarithm of $\Pr(\mathbf{q} \mid \mathbf{d})$:

$$\mathrm{Rel}(\mathbf{d}, \mathbf{q}) = \log \Pr(\mathbf{q} \mid \mathbf{d})$$

Now express $\mathbf{q}$ as a series of words drawn from a vocabulary $\mathcal{V} = \{w_1, w_2, \cdots w_V\}$; that is, $\mathbf{q} = q_1 q_2 \cdots q_K$, where K is the number of words in the query, and $q_i \in \mathcal{V}$ for $1 \leq i \leq K$. Then given a unigram model derived from $\mathbf{d}$ which assigns a probability $\Pr(w \mid \mathbf{d})$ to each word w in $\mathcal{V}$, we can compute $\mathrm{Rel}(\mathbf{d}, \mathbf{q})$ as follows:

$$\begin{aligned}\mathrm{Rel}(\mathbf{d}, \mathbf{q}) &= \log \Pr(q_1 q_2 \cdots q_K \mid \mathbf{d}) = \sum_{i=1}^{K} \log \Pr(q_i \mid \mathbf{d}) \\ &= \sum_{\substack{w \in \mathcal{V}, \\ c(w;\mathbf{q})>0}} c(w;\mathbf{q}) \log \Pr(w \mid \mathbf{d}) \qquad (1)\end{aligned}$$

where $c(w;\mathbf{q})$ is the word count of w in $\mathbf{q}$.

Lafferty and Zhai[15] showed that this method of relevance scoring happens to be equivalent to ranking documents by the negative Kullback-Leibler divergence $-\Delta_{\mathrm{KL}}(\mathbf{q}, \mathbf{d})$ of the unsmoothed empirical distribution of words in $\mathbf{q}$ from a (possibly smoothed) distribution of words in $\mathbf{d}$, since

$$-\Delta_{\mathrm{KL}}(\mathbf{q}, \mathbf{d}) = \frac{1}{K} \log \Pr(\mathbf{q} \mid \mathbf{d}) + H_{\mathbf{q}} = \frac{1}{K} \mathrm{Rel}(\mathbf{d}, \mathbf{q}) + H_{\mathbf{q}}$$

where K and $H_{\mathbf{q}}$ are constants which do not depend on $\mathbf{d}$.

3.1.2 Document Model Smoothing

Before using Equation 1, we must estimate a unigram model from $\mathbf{d}$: that is, an assignment of probabilities $\Pr(w \mid \mathbf{d})$ for all $w \in \mathcal{V}$. One way to do this is to use a maximum likelihood estimate (MLE) – an assignment of $\Pr(w \mid \mathbf{d})$ for all w which maximizes the probability of generating $\mathbf{d}$. The MLE is given by the equation

$$\Pr{}_{\mathrm{mle}}(w \mid \mathbf{d}) = \frac{c(w;\mathbf{d})}{|\mathbf{d}|}$$

where $c(w;\mathbf{d})$ is the number of occurrences of w in $\mathbf{d}$, and $|\mathbf{d}|$ is the total number of words in $\mathbf{d}$. However, using this formula means we will get a value of zero for $\Pr(\mathbf{q} \mid \mathbf{d})$ if even a single query word q_i is not found in $\mathbf{d}$. To overcome this problem, we smooth the model by assigning some probability mass to such unseen words. Specifically, we adopt a two-stage smoothing method[35]:

$$\Pr(w \mid \mathbf{d}) = (1-\lambda) \frac{c(w;\mathbf{d}) + \mu \Pr(w \mid \mathcal{C})}{|\mathbf{d}| + \mu} + \lambda \Pr(w \mid \mathcal{U}) \qquad (2)$$

Here, $\mathcal{U}$ denotes a background language model, and $\mu > 0$ and $\lambda \in (0, 1)$ are parameters to the smoothing procedure. This is a combination of Bayesian smoothing using Dirichlet priors[18] and Jelinek-Mercer smoothing[13].

The parameter λ can be set empirically according to the nature of the queries. For the parameter μ, we adopt the estimation procedure of Zhai and Lafferty[35]: we maximize the leave-one-out log likelihood of the document collection, namely

$$\ell_{-1}(\mu \mid \mathcal{C}) = \sum_{\mathbf{d} \in \mathcal{C}} \sum_{w \in \mathcal{V}} c(w;\mathbf{d}) \log \left(\frac{c(w;\mathbf{d}) - 1 + \mu \Pr(w \mid \mathcal{C})}{|\mathbf{d}| - 1 + \mu} \right) \qquad (3)$$

This can be done by using Newton's method to solve the equation

$$\ell'_{-1}(\mu \mid \mathcal{C}) = 0 \qquad (4)$$

With this, we can now compute the relevance of $\mathbf{d}$ to $\mathbf{q}$ according to Equation 1.

3.2 Statistical Lattice-Based Retrieval Method

We now describe our statistical lattice-based method for query-by-example SDR. In contrast to the above baseline method, our proposed method works on the lattice representation of spoken documents, as generated by a speech recognizer. It extends Chia et al.'s[7] SDR method for textual queries, by incorporating uncertainty of query word counts into the retrieval process.

3.2.1 Model Estimation for Documents in Collection

As in the baseline method, we estimate a statistical model for each document $\mathbf{d}$ in the collection corpus $\mathcal{C}$. This is done by generating lattices for the document and computing the expected count of each word in each document.

Lattice Generation.

First, each spoken document in the collection corpus is divided into M short speech segments. Let $\mathbf{o}$ denote the acoustic observations comprising such a speech segment; we then use a speech recognizer to generate a lattice[33] from each $\mathbf{o}$.

We can consider the lattice generation process as proceeding in two steps. First we output a lattice L containing only acoustic likelihood information; we can consider L to comprise a set of nodes S and a set of edges $A \subset S \times S$, where

each edge $a \in A$ is labeled with a word hypothesis $t[a]$ and an acoustic probability $p[a]$. Each path $\pi = a_1 a_2 \cdots a_N$ through the lattice contains a hypothesis of the series of words spoken in this speech segment, $\mathbf{t}[\pi] = t[a_1]t[a_2]\cdots t[a_N]$, along with acoustic probabilities $\Pr(o_1 \mid a_1) = p[a_1]$, $\Pr(o_2 \mid a_2) = p[a_2]$, $\cdots$ $\Pr(o_N \mid a_N) = p[a_N]$; here, o_i denotes the acoustic observations for the time interval corresponding to the edge a_i and the word t_i hypothesized by the speech recognizer, such that $o_1 o_2 \cdots o_N = \mathbf{o}$. We have

$$\Pr(\mathbf{o} \mid \pi) = \prod_{i=1}^{N} \Pr(o_i \mid a_i)$$

We then rescore each lattice with an n-gram language model, to yield an expanded lattice in which paths are weighted by their posterior probabilities $\Pr(\pi \mid \mathbf{o})$ rather than their acoustic likelihoods $\Pr(\mathbf{o} \mid \pi)$.

In theory, the rescoring can be done by simply multiplying acoustic probabilities with n-gram probabilities, since $\Pr(\mathbf{t} \mid \mathbf{o}) \propto \Pr(\mathbf{t}, \mathbf{o}) = \Pr(\mathbf{t}) \Pr(\mathbf{o} \mid \mathbf{t})$ for each transcription hypothesis $\mathbf{t}$. In practice, it has been found to be useful[5] to assign a higher weight to the n-gram probabilities by applying a scaling factor $\omega > 1$, and to introduce a word insertion penalty $\rho \leq 0$; thus, the paths in the rescored lattice are weighted by

$$\tilde{\Pr}(\pi, \mathbf{o}) = \Pr(\mathbf{t}[\pi]) \left(\Pr(\mathbf{o} \mid \pi) e^{\rho |\pi|} \right)^{1/\omega}$$

where $|\pi|$ is the length of π.

Lattice Pruning.

The lattice is then pruned, by removing paths in the lattice whose log joint probabilities ($\ln \tilde{\Pr}(\pi, \mathbf{o})$) are not within a threshold Θ_{doc} of the best path's log probability; specifically, by removing nodes and edges which occur only on such low-probability paths[21]. After pruning, we obtain a rescored and pruned lattice L' with nodes S' and edges A'; in this new lattice, each edge a_i is labeled with a joint probability $p'[a_i] = \tilde{\Pr}(o_i, a_i)$ instead of an acoustic likelihood $\Pr(a_i \mid o_i)$.

Expected Count Computation.

Next, we compute the expected count of each word in each document. For each word w and each speech segment $\mathbf{o}$, the expected count of w in $\mathbf{o}$ is

$$\mathrm{E}[c(w; \mathbf{o})] = \sum_{\pi} c(w; \pi) \tilde{\Pr}(\pi \mid \mathbf{o}) \tag{5}$$

where the sum is taken over all lattice paths π, and $c(w; \pi)$ denotes the word count of w in the hypothesized transcript given by π. We can also analogously compute the expected length of $\mathbf{o}$:

$$\mathrm{E}[|\mathbf{o}|] = \sum_{\pi} |\pi| \tilde{\Pr}(\pi \mid \mathbf{o}) \tag{6}$$

A naive implementation of Equations 5 and 6 will be extremely inefficient, since the total number of paths in each lattice is roughly exponential in the length of the speech segment [26]. To compute $\mathrm{E}[c(w; \mathbf{o})]$ and $\mathrm{E}[|\mathbf{o}|]$ efficiently, one practical approach is to use a dynamic programming algorithm, based on the standard forward-backward algorithm[24].

Finally, for each document $\mathbf{d}$ comprised of M speech segments represented by M series of acoustic observations $\mathbf{o}^{(1)}$, $\mathbf{o}^{(2)}$, $\cdots$ $\mathbf{o}^{(M)}$, we can compute the expected count of w in $\mathbf{d}$ and the expected length of $\mathbf{d}$:

$$\mathrm{E}[c(w; \mathbf{d})] = \sum_{j=1}^{M} \mathrm{E}[c(w; \mathbf{o}^{(j)})] \qquad \mathrm{E}[|\mathbf{d}|] = \sum_{j=1}^{M} \mathrm{E}[\left|\mathbf{o}^{(j)}\right|]$$

Model Smoothing.

We now replace $c(w; \mathbf{d})$ and $|\mathbf{d}|$ in Equation 2 with $\mathrm{E}[c(w; \mathbf{d})]$ and $\mathrm{E}[|\mathbf{d}|]$; thus

$$\Pr(w \mid \mathbf{d}) = (1 - \lambda) \frac{\mathrm{E}[c(w; \mathbf{d})] + \mu \Pr(w \mid \mathcal{C})}{\mathrm{E}[|\mathbf{d}|] + \mu} + \lambda \Pr(w \mid \mathcal{U}) \tag{7}$$

In addition, we use an approximate procedure for estimating μ, by considering the leave-one-out log likelihood of a virtual data set with word counts which are close to the expected counts. Specifically, we replace $c(w; \mathbf{d})$ and $|\mathbf{d}|$ in Equation 3 with rounded-off expected counts $\left\lfloor \mathrm{E}[c(w; \mathbf{d})] + \frac{1}{2} \right\rfloor$ and $\sum_{w \in \mathcal{V}} \left\lfloor \mathrm{E}[c(w; \mathbf{d})] + \frac{1}{2} \right\rfloor$ respectively.

3.2.2 Model Estimation for Query Exemplars

We process the query exemplars in a similar way as the documents: each query exemplar $\mathbf{q}$ is divided into short speech segments, lattices are generated from the speech segments, the lattices are then pruned according to a log path probability threshold Θ_{qry}, and the expected count of each word w in $\mathbf{q}$ and the expected length of $\mathbf{q}$ is computed.

We can then estimate a probability model for each $\mathbf{q}$ by normalizing the expected word counts:

$$\Pr(w \mid \mathbf{q}) = \frac{\mathrm{E}[c(w; \mathbf{q})]}{\mathrm{E}[|\mathbf{q}|]}$$

3.2.3 Relevance Score Computation

In Section 3.1.1, we stated that the use of the query likelihood for relevance scoring is equivalent to ranking documents by the negative Kullback-Leibler divergence; we use the same measure for lattice-based retrieval. Given probability models for the distribution of words in both $\mathbf{d}$ and $\mathbf{q}$, the negative Kullback-Leibler divergence is

$$-\Delta_{\text{KL}}(\mathbf{q}, \mathbf{d}) = \sum_{w \in \mathcal{V}} \Pr(w \mid \mathbf{q}) \log \Pr(w \mid \mathbf{d}) + H_{\mathbf{q}}$$

where again $H_{\mathbf{q}}$ is a constant. Thus we can compute the relevance of $\mathbf{d}$ to $\mathbf{q}$ as

$$\begin{aligned} \mathrm{Rel}_{\text{lat}}(\mathbf{d}, \mathbf{q}) &= \sum_{w \in \mathcal{V}} \Pr(w \mid \mathbf{q}) \log \Pr(w \mid \mathbf{d}) \\ &= \frac{1}{\mathrm{E}[|\mathbf{q}|]} \sum_{\substack{w \in \mathcal{V}, \\ \mathrm{E}[c(w;\mathbf{q})] > 0}} \mathrm{E}[c(w; \mathbf{q})] \log \Pr(w \mid \mathbf{d}) \end{aligned}$$

where $\Pr(w \mid \mathbf{d})$ is as estimated in Equation 7.

3.3 Experiments

To evaluate our proposed retrieval method, we performed experiments using the Fisher English Training corpus released by the Linguistic Data Consortium (LDC2004S13 and LDC2005S13). This is a conversational telephone speech corpus comprising 11,699 recorded conversations each taking up to 10 minutes, for a total of more than 1,920 hours, corresponding to approximately 109Mb of transcribed text.

ENG01. *Professional Sports on TV.* Do either of you have a favorite TV sport? How many hours per week do you spend watching it and other sporting events on TV?

Figure 1: An example of a topic specification in the Fisher corpus

The conversations were sampled at 8kHz, and have been broken up into speech segments of up to 30 seconds each.

Each conversation was initiated by a topic statement chosen from a list of 40 topics, and speakers mostly adhered to the suggested topics. Each topic was specified with a topic identifier, a topic title, and a verbose topic description; an example of a topic specification is given in Figure 1.

We divided the speech corpus into three portions:

- 1,600 conversations for training acoustic models ($\mathcal{S}_1$);
- 5,005 conversations for training an n-gram language model for lattice rescoring ($\mathcal{S}_2$); and
- 5,094 conversations for the collection corpus and query exemplars ($\mathcal{S}_3$).

3.3.1 Task Setup

As each telephone conversation pertains to a single topic, we decided to simply treat each conversation in $\mathcal{S}_3$ as a possible unit of retrieval – a "document". Each document comprises multiple consecutive speech segments.

To obtain query exemplars for our task setup, we decided to select 40 conversations as exemplars to represent the 40 topics. To do this, we performed a preliminary round of text retrieval on the *reference* transcripts of the conversations in $\mathcal{S}_3$, using the titles and verbose descriptions of the 40 topic specifications as queries; for each topic, we then selected the highest-ranked conversation as an exemplar. (We used this procedure instead of randomly selecting exemplars, as the latter might cause us to select exemplars which are unrepresentative of their topics, and retrieval results obtained with them would not be useful for our subsequent analysis.)

We then assigned 8 of the query exemplars to be development queries, and the remaining 32 as test queries. The exemplars, along with the Fisher topics they represent, are listed in Table 1. The remaining 5,054 conversations in $\mathcal{S}_3$ were designated as the collection corpus ($\mathcal{C}$) for our experimental setup.

To obtain ground truth relevance judgements, we adopted the following procedure: a document in $\mathcal{C}$ was deemed to be relevant to a query if and only if the document and query exemplar were about the same Fisher topic, according to the topic assignment tables provided by the LDC with the corpus. An examination of a small sample of the corpus showed that the conversations did tend to stay with their initial topics; thus we had reason to believe that the LDC's topic annotations are good indicators of the various documents' subject matter.

3.3.2 Preprocessing of Documents and Queries

To process the conversations in the corpus, we used the HTK speech recognition engine[34] for acoustic model training, lattice generation, and lattice rescoring. We divided each speech segment into 25-millisecond speech frames with a frame shift of 10 milliseconds, and coded each frame as a 39-dimensional feature vector consisting of 12 mel frequency cepstral coefficients (MFCCs) and normalized energy, and their first and second order derivatives; speaker-based cepstral mean and variance normalization were also applied.

Table 1: List of query exemplars

Development queries

Topic	Exemplar	Topic	Exemplar	Topic	Exemplar
ENG04	fe_03_03609	ENG20	fe_03_10205	ENG29	fe_03_08738
ENG05	fe_03_03916	ENG22	fe_03_08256	ENG39	fe_03_09447
ENG09	fe_03_09859	ENG23	fe_03_08390		

Test queries

Topic	Exemplar	Topic	Exemplar	Topic	Exemplar
ENG01	fe_03_02783	ENG15	fe_03_06120	ENG30	fe_03_07041
ENG02	fe_03_03138	ENG16	fe_03_04700	ENG31	fe_03_05822
ENG03	fe_03_09686	ENG17	fe_03_06378	ENG32	fe_03_08970
ENG06	fe_03_00329	ENG18	fe_03_06419	ENG33	fe_03_09033
ENG07	fe_03_07846	ENG19	fe_03_05482	ENG34	fe_03_02223
ENG08	fe_03_04071	ENG21	fe_03_04938	ENG35	fe_03_02282
ENG10	fe_03_04364	ENG24	fe_03_00267	ENG36	fe_03_05254
ENG11	fe_03_05948	ENG25	fe_03_05055	ENG37	fe_03_00075
ENG12	fe_03_01242	ENG26	fe_03_00279	ENG38	fe_03_02628
ENG13	fe_03_10001	ENG27	fe_03_08587	ENG40	fe_03_10613
ENG14	fe_03_08034	ENG28	fe_03_11639		

Using the acoustic modeling data in $\mathcal{S}_1$, we trained a set of 33,830 tied-state cross-word triphone models for 39 English phonemes, and two silence models. Each triphone was modeled as a left-to-right 3-state HMM with each state having 8 Gaussian mixture components. Pronunciations of words were obtained from the CMU Pronouncing Dictionary[1].

We then used HTK to generate lattices for the speech segments in the collection corpus $\mathcal{C}$ and the query exemplars. The lattices were then rescored with a trigram model trained from the reference transcripts of the conversations in $\mathcal{S}_2$; the size of this model training data is approximately 65Mb. 1-best ASR transcripts were decoded by finding the highest-probability hypothesis from each of the rescored lattices. We computed the word error rate of the 1-best transcripts of the collection and queries: the WER was found to be 48.1%. All words in the collection corpus and the queries were then stemmed using the Porter stemming algorithm[23].

Besides HTK, several other tools were used to help with the preprocessing. Pruning of the rescored lattices was done using the AT&T FSM library[21] to prune the rescored lattices, and expected count computation was done using the SRILM toolkit[29]. To make retrieval efficient, we stored expected word counts in B-tree index structures via library routines in the CMU Lemur toolkit[2].

3.3.3 Retrieval and Evaluation

We then performed retrieval on the document collection using the algorithms in Section 3, using the reference transcripts, the 1-best ASR transcripts, and expected counts from lattices of the query exemplars and the collection corpus. For the retrieval parameters, we set $\lambda = 0.7$, which was suggested by Zhai and Lafferty[35] to give good retrieval performance for verbose queries; values of μ (obtained by

[1] ftp://ftp.cs.cmu.edu/project/speech/dict/cmudict.0.6

[2] http://www.lemurproject.org/

Table 2: Summary of experimental results without stop word removal

System	Retrieval source		Pruning parameters $(\Theta_{\text{qry}}, \Theta_{\text{doc}})$	Mean average precision	
	Queries	Documents		For devel. queries	For test queries
Ref → Ref	Exemplar reference	Reference	–	0.7941	0.7468
1-best → 1-best	Exemplar 1-best	1-best	–	0.7580	0.6958
1-best → Lat	Exemplar 1-best	Lattices	$(-, 120)$	0.7613	0.7009
Lat → 1-best	Exemplar lattices	1-best	$(240, -)$	0.7669	0.7023
Lat → Lat	Exemplar lattices	Lattices	$(240, 120)$	0.7740	0.7079
Top → Ref	Topic specifications	Reference	–	0.8325	0.8149
Top → 1-best	Topic specifications	1-best	–	0.7922	0.7613
Top → Lat	Topic specifications	Lattices	$(-, 160)$	0.8023	0.7723

solving Equation 4) were found to range between 1,300 and 2,600. For the background language model $\mathcal{U}$, we used the language model derived from the collection corpus $\mathcal{C}$.

The results of retrieval were checked against the ground truth relevance judgements, and evaluated in terms of the non-interpolated mean average precision (MAP):

$$\text{MAP} = \frac{1}{L}\sum_{i=1}^{L}\left(\frac{1}{R_i}\sum_{j=1}^{R_i}\frac{j}{r_{i,j}}\right)$$

where L denotes the total number of queries, R_i the total number of documents relevant to the ith query, and $r_{i,j}$ the position of the jth relevant document in the ranked list output by the retrieval method for query i.

For the lattice-based retrieval method, we performed retrieval with the development queries using different values of the query lattice pruning threshold Θ_{qry} from 20 to 300, and values of the document lattice pruning threshold Θ_{doc} from 20 to 200; we then used the values of Θ_{qry} and Θ_{doc} with the best MAP to do retrieval with the test queries.

As a further comparison, we also performed retrieval on the collection corpus using the original Fisher topic specifications (as in Figure 1) as queries.

3.4 Experimental Results

The results of our experiments are summarized in Table 2. We see that when performing retrieval using lattices (Lat → Lat), the MAP of the development queries was highest at $\Theta_{\text{qry}} = 240$ and $\Theta_{\text{doc}} = 120$, at which point the MAP of the test queries was 0.7079. A one-tailed paired Student's t-test with 31 degrees of freedom between the 1-best retrieval results (1-best → 1-best) and the lattice-based retrieval results yields $t = 3.58$, and a one-tailed Wilcoxon signed-rank test[31] yields $w_+ = 440$; these indicate that the improvement due to lattice-based retrieval in this case was significant at the 99.95% confidence level.

We also found that using expected counts of lattices from both queries and documents (Lat → Lat) resulted in better retrieval performance than using lattices for documents only (1-best → Lat), or using lattices for queries only (Lat → 1-best). This shows that the alternative hypotheses contained in query and document lattices were able to reinforce one another to yield better retrieval accuracy.

However, when compared against the performance obtainable by using short queries – namely, using the original Fisher topic specifications as queries – our lattice-based query-by-example method still fell short. In fact, even when the reference transcripts of documents and query exemplars are used in query-by-example retrieval (Ref → Ref), we found that the MAP achieved on the test queries is only 0.7468, which is still lower than the MAP of 0.7613 obtainable using topic specifications, even when working merely with 1-best transcripts of the collection corpus (Top → 1-best). This suggests that the very nature of the query exemplars presents difficulties in the way of accurate retrieval.

Table 3: Most frequent words (after stemming) in the topic specification for `ENG01`, and in the reference transcripts of the corresponding exemplar `fe_03_02783`

`ENG01` topic spec.		`fe_03_02783`	
Word	Frequency	Word	Frequency
v.	3	i	108
t.	3	the	82
sport	3	and	66
you	2	thei	54
on	2	to	53
do	2	it	41
week	1	yeah	39
watch	1	on	37
spend	1	you	36
profession	1	like	36

4. STOP WORD REMOVAL

To bridge the accuracy gap between query by example and SDR with short queries, we tried applying stop word removal on both the document collection and the query exemplars. As illustrated in Table 3, the distribution of words is different in the two types of queries; the short topic specification contains stop words but still has a high concentration of content words specifying the topic, while the most frequent words in the query exemplar are all stop words (such as "the") or filler words (such as "yeah"); thus by filtering away stop words, we expect to be able to reduce the difference between the nature of the two types of queries.

For reference and 1-best transcripts, stopping can be done by excluding stop words from the vocabulary $\mathcal{V}$, and omitting such words from the transcripts of $\mathbf{d}$ and $\mathbf{q}$. For lattice-based retrieval, stop word removal can be achieved by treating all stop words in documents and queries as having an expected count of zero, and also leaving them out of the computation of $\mathrm{E}[|\mathbf{d}|]$ and $\mathrm{E}[|\mathbf{q}|]$.

Table 4: Summary of experimental results with stop word removal

System	Retrieval source		Stop word list	Pruning parameters $(\Theta_{\text{qry}}, \Theta_{\text{doc}})$	Mean average precision	
	Queries	Documents			For devel. queries	For test queries
Ref $\xrightarrow{\texttt{gla}}$ Ref	Exemplar reference	Reference	`gla`	–	0.7884	0.7630
1-best $\xrightarrow{\texttt{gla}}$ 1-best	Exemplar 1-best	1-best	`gla`	–	0.7699	0.7193
1-best $\xrightarrow{\texttt{gla}}$ Lat	Exemplar 1-best	Lattices	`gla`	$(-, 140)$	0.7753	0.7283
Lat $\xrightarrow{\texttt{gla}}$ 1-best	Exemplar lattices	1-best	`gla`	$(240, -)$	0.7801	0.7285
Lat $\xrightarrow{\texttt{gla}}$ Lat	Exemplar lattices	Lattices	`gla`	$(240, 120)$	0.7868	0.7364
Ref $\xrightarrow{\texttt{smart}}$ Ref	Exemplar reference	Reference	`smart`	–	0.8363	0.7781
1-best $\xrightarrow{\texttt{smart}}$ 1-best	Exemplar 1-best	1-best	`smart`	–	0.8271	0.7406
1-best $\xrightarrow{\texttt{smart}}$ Lat	Exemplar 1-best	Lattices	`smart`	$(-, 140)$	0.8321	0.7499
Lat $\xrightarrow{\texttt{smart}}$ 1-best	Exemplar lattices	1-best	`smart`	$(240, -)$	0.8355	0.7487
Lat $\xrightarrow{\texttt{smart}}$ Lat	Exemplar lattices	Lattices	`smart`	$(240, 160)$	0.8421	0.7569

4.1 Experiments and Results

We tested the effect of stop word removal on the Fisher corpus retrieval task described in Section 3.3. We experimented with using two different stop lists:

- `gla` – a 319-word stop list[3] maintained by the University of Glasgow;
- `smart` – a 571-word stop list[4] which was used in the classical SMART information retrieval system[3].

The results are shown in Table 4. When performing retrieval using lattices with stopping using the `gla` stop list (Lat $\xrightarrow{\texttt{gla}}$ Lat), the MAP of the development queries was highest at $\Theta_{\text{qry}} = 240$ and $\Theta_{\text{doc}} = 120$, at which point the MAP of the test queries was 0.7364; a t-test shows that the improvement over retrieval using 1-best transcripts (1-best $\xrightarrow{\texttt{gla}}$ 1-best) was significant at the 99.99% confidence level ($t = 4.24$, $w_+ = 483$). When the `smart` stop list was used, the MAP of lattice-based retrieval (Lat $\xrightarrow{\texttt{smart}}$ Lat) was also found to be significantly better than 1-best retrieval (1-best $\xrightarrow{\texttt{smart}}$ 1-best) at the 99.95% confidence level under the t-test ($t = 3.76$), and at the 99.99% confidence level under the Wilcoxon test ($w_+ = 472$). Thus, the use of lattices for query-by-example SDR still produces better performance than query-by-example SDR with 1-best transcripts, even with the use of stopping.

When compared to lattice-based retrieval without stop word removal (Lat → Lat), the results were also better. The MAP increased from 0.7079 to 0.7364 with the use of the `gla` stop list (Lat $\xrightarrow{\texttt{gla}}$ Lat), while using `smart` (Lat $\xrightarrow{\texttt{smart}}$ Lat) caused the MAP to increase to 0.7569. We therefore see that stop word removal can help to boost retrieval accuracy for the query-by-example task.

Also, we found a significant difference between the effects of the two different stop lists: for lattice-based retrieval, switching from the `gla` stop list to the `smart` stop list resulted in a MAP increase which was significant at the 97.5% confidence level ($t = 2.02$, $w_+ = 379$). This suggests that the precise choice of stop list can also have an impact on query-by-example performance, and thus this issue merits attention.

[3] `http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words`

[4] `http://members.unine.ch/jacques.savoy/clef/englishST.txt`

5. CONCLUSIONS AND FUTURE WORK

We have presented a method for performing query-by-example SDR using lattices of documents and spoken queries, based on a statistical language modeling retrieval framework. Results show that our new method can significantly improve the retrieval MAP compared to using only the 1-best ASR transcripts.

Furthermore, retrieval results using stop word removal with two different stop word lists showed that using lattices gave better retrieval accuracy than using 1-best transcripts for both stop word lists; thus it can be seen that lattice-based retrieval yields a consistent improvement over 1-best retrieval across a variety of retrieval setups.

For future work, we would like to extend our statistical lattice-based retrieval framework to other speech processing tasks, such as spoken document classification.

6. REFERENCES

[1] J. Allan. Robust techniques for organizing and retrieving spoken documents. *EURASIP Journal on Applied Signal Processing*, 2003(1):103–114, 2003.

[2] A. Berger and J. Lafferty. Information retrieval as statistical translation. In *Proceedings of SIGIR 1999*, pages 222–229, New York, NY, USA, 1999. ACM Press.

[3] C. Buckley. Implementation of the SMART information retrieval system. Technical Report TR85-686, Cornell University, Ithaca, NY, USA, 1985.

[4] G. Carvalho, D. M. de Matos, and V. Rocio. Document retrieval for question answering: a quantitative evaluation of text preprocessing. In *Proceedings of the ACM First Ph. D. Workshop in CIKM*, pages 125–130, New York, NY, USA, 2007. ACM.

[5] C. Chelba and A. Acero. Position specific posterior lattices for indexing speech. In *Proceedings of ACL 2005*, pages 443–450, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

[6] B. Chen, H.-M. Wang, and L.-S. Lee. A discriminative HMM/N-gram-based retrieval approach for Mandarin spoken documents. *ACM Transactions on Asian Language Information Processing*, 3(2):128–145, 2004.

[7] T. K. Chia, H. Li, and H. T. Ng. A statistical language modeling approach to lattice-based spoken document retrieval. In *Proceedings of EMNLP-CoNLL 2007*, pages 810–818, 2007.

[8] N. Colineau and A. Halber. A hybrid approach to spoken query processing in document retrieval system. In *Proceedings of the ESCA ETRW Workshop: Accessing information in spoken audio*, pages 31–36, 1999.

[9] D. He, H. R. Park, G. C. Murray, M. Subotin, and D. W. Oard. TDT-2002 topic tracking at Maryland: First experiments with the Lemur toolkit. Technical Report LAMP-TR-099, CS-TR-4454, UMIACS-TR-2003-24, University of Maryland, College Park, Feb. 2003.

[10] T. Hori, I. L. Hetherington, T. J. Hazen, and J. R. Glass. Open-vocabulary spoken utterance retrieval using confusion networks. In *Proceedings of IEEE ICASSP*, pages 73–76, Honolulu, Hawaii, 2007.

[11] D. A. James. *The Application of Classical Information Retrieval Techniques to Spoken Documents*. PhD thesis, University of Cambridge, 1995.

[12] D. A. James and S. J. Young. A fast lattice-based approach to vocabulary independent wordspotting. In *Proceedings of IEEE ICASSP*, pages 377–380, Adelaide, Australia, 1994.

[13] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, pages 381–397, 1980.

[14] G. J. F. Jones, J. T. Foote, K. S. Jones, and S. J. Young. Retrieving spoken documents by combining multiple index sources. In *Proceedings of SIGIR 1996*, pages 30–38, New York, NY, USA, 1996. ACM Press.

[15] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR 2001*, pages 111–119, New York, NY, USA, 2001. ACM.

[16] Y.-Y. Lo and J.-L. Gauvain. The LIMSI topic tracking system for TDT2002. In *Proceedings of DARPA Topic Detection and Tracking Workshop*, Gaithersburg, Nov 2002.

[17] Y.-Y. Lo and J.-L. Gauvain. Tracking topics in broadcast news data. In *Proceedings of ISCA Workshop on Multilingual Spoken Document Retrieval*, Hong Kong, April 2003.

[18] D. J. C. Mackay and L. C. B. Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):1–19, 1994.

[19] J. Mamou, D. Carmel, and R. Hoory. Spoken document retrieval from call-center conversations. In *Proceedings of SIGIR 2006*, pages 51–58, New York, NY, USA, 2006. ACM Press.

[20] 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*, 14(4):373–400, 2000.

[21] M. Mohri, F. Pereira, and M. Riley. The design principles of a weighted finite-state transducer library. *Theoretical Computer Science*, 231:17–32, 2000.

[22] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR 1998*, pages 275–281, New York, NY, USA, 1998. ACM Press.

[23] M. F. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.

[24] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[25] M. Saraclar and R. Sproat. Lattice-based search for spoken utterance retrieval. In *Proceedings of HLT-NAACL 2004*, pages 129–136, Boston, Massachusetts, USA, May 2004. Association for Computational Linguistics.

[26] M. A. Siegler. *Integration of Continuous Speech Recognition and Information Retrieval for Mutually Optimal Performance*. PhD thesis, Carnegie Mellon University, 1999.

[27] M. P. Sinka and D. W. Corne. Towards modernised and web-specific stoplists for web document analysis. *Proceedings of the IEEE/WIC International Conference on Web Intelligence*, pages 396–402, 2003.

[28] F. Song and W. B. Croft. A general language model for information retrieval. In *Proceedings of CIKM 1999*, pages 316–321, New York, NY, USA, 1999. ACM Press.

[29] A. Stolcke. SRILM – an extensible language modeling toolkit. In *Proceedings of ICSLP 2002*, volume 2, pages 901–904, Denver, CO, USA, 2002.

[30] E. M. Voorhees and D. Harman. Overview of the Ninth Text REtrieval Conference (TREC-9). In *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 1–14, 2000.

[31] R. E. Walpole and R. H. Myers. *Probability and Statistics for Engineers and Scientists*. Macmillan, Inc., New York, 4th edition, 1989.

[32] C. L. Wayne. Topic detection and tracking in English and Chinese. In *Proceedings of IRAL 2000*, pages 165–172, New York, NY, USA, 2000. ACM.

[33] F. Weng, A. Stolcke, and A. Sankar. Efficient lattice representation and generation. In *Proceedings of ICSLP 1998*, volume 6, pages 2531–2534, Sydney, Australia, 1998.

[34] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book (for HTK Version 3.4)*. Cambridge University Press, Cambridge, UK, 2006.

[35] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.

[36] Z.-Y. Zhou, P. Yu, C. Chelba, and F. Seide. Towards spoken-document retrieval for the internet: Lattice indexing for large-scale web-search architectures. In *Proceedings of HLT-NAACL 2006*, pages 415–422, New York City, USA, June 2006. Association for Computational Linguistics.