Spoken document retrieval leveraging unsupervised and supervised topic modeling techniques

Kuan-Yu Chen††, Hsin-Min Wang†† and Berlin Chen††

SUMMARY
This paper describes the application of two attractive categories of topic modeling techniques to the problem of spoken document retrieval (SDR), viz. document topic model (DTM) and word topic model (WTM). Apart from using the conventional unsupervised training strategy, we explore a supervised training strategy for estimating these topic models, imagining a scenario that user query logs along with click-through information of relevant documents can be utilized to build an SDR system. This attempt has the potential to associate relevant documents with queries even if they do not share any of the query words, thereby improving on retrieval quality over the baseline system. Likewise, we also study a novel use of pseudo-supervised training to associate relevant documents with queries through a pseudo-feedback procedure. Moreover, in order to lessen SDR performance degradation caused by imperfect speech recognition, we investigate leveraging different levels of index features for topic modeling, including words, syllable-level units, and their combination. We provide a series of experiments conducted on the TDT (TDT-2 and TDT-3) Chinese SDR collections. The empirical results show that the methods deduced from our proposed modeling framework are very effective when compared with a few existing retrieval approaches.

keywords: spoken document retrieval, topic model, supervised training, pseudo-supervised training, subword-level indexing.

1. Introduction
The field of spoken document retrieval (SDR) has witnessed considerable research activity in the last decade. This is due in large part to the advances in automatic speech recognition (ASR) and the ever-increasing volumes of multimedia data associated with spoken documents made available to the public, such as radio and TV broadcasts, lecture recordings, meetings and telephone conversations, digital archives, among many others [1, 2]. Much work has been devoted towards developing robust indexing (or representation) techniques so as to extract probable spoken terms or phrases inherent in a spoken document that could match the query words (the representation of information need for a user) or phrases literally (the so-called spoken term detection, STD) [3], instead of revolving around the notion of relevance of a spoken document to a query, through the use of existing retrieval models [4, 5].

Most retrieval systems participated in the TREC-SDR evaluations had claimed that speech recognition errors do not seem to cause very significant deterioration of the retrieval quality when merely using imperfect recognition transcripts derived from one-best recognition results [6]. We might attribute such invulnerability to the fact that the TREC-style queries tend to be quite long and contain different words that describe a similar concept and thus help these queries match their relevant spoken documents. Furthermore, a query word (or phrase) may occur repeatedly (more than once) within a relevant spoken document, and the word is not always misrecognized. Although SDR is seemingly a solved problem on account of these reasons, we, however, believe that it would still present a challenge in situations where the queries are relatively short and there exists deviation in word usage between the queries and documents.

More recently, language modeling (LM) for SDR has received great attention due to its inherent neat formulation and clear probabilistic meaning, as well as state-of-the-art performance [7, 8, 9, 10]. In practice, the relevance measure for various LM approaches is usually computed by two distinct matching strategies, namely, literal term matching and concept matching [11]. The unigram language model (ULM) is the most prominent example for literal term matching [7, 8]. In this approach, each document is interpreted as a generative model composed of a mixture of unigram (multinomial) distributions for observing words of the language, while the query is regarded as observations, expressed as a sequence of words (or index terms). Accordingly, documents can be ranked in decreasing order of the probability that each document model generates the query. Yet, there have been a number of studies to further extend ULM in an attempt to account for the dependency between terms based on n-grams of various orders, or some grammar structures, mostly leading to only mild gains or even spoiled results [12, 9].

However, most of the above LM approaches would suffer from the problem of the vocabulary gap, which might make the retrieval performance degrade severely as a given query and its relevant documents are using quite a different set of words. Further, as we known, a document is relevant if it could address the stated information need of the query, not because it just happens to contain all the words in the query. Concept matching tries to explore the latent topic information conveyed in the query and documents, based on which the retrieval is performed; the probabilistic latent semantic analysis (PLSA) [13] and the
latent Dirichlet allocation (LDA) [14] are often considered two primary instantiations of this category. They both treat each document as a document topic model (DTM) and introduce a set of latent topic variables to describe the “word-document” co-occurrence characteristics. The relevance between a query and a document is not computed directly based on the frequency of the query words occurring in the document, but instead based on the frequency of these words in the latent topics as well as the likelihood that the document generates the respective topics, which in fact exhibits some sort of concept matching. Despite the fact that there are many follow-up studies and extensions of LDA and PLSA, empirical evidence in the literature indicates that more sophisticated (or complicated) topic models, such as pachinko Allocation model (PAM), do not necessary offer further retrieval benefits [12, 15]. Interested readers can refer to [12, 16] for comprehensive overviews of a wide spectrum of LM techniques that have been developed and applied to various text information retrieval (IR) tasks.

Taking a step further, we have recently introduced a new perspective on topic modeling for SDR, from which the word topic model (WTM) [11] and its variants [17, 18] were proposed to discover the long-span co-occurrence dependence “between words” through a set of latent topics. Each spoken document in the collection in turn can be represented as a composite WTM model in an efficient way for predicting an observed query. This modeling paradigm also has been applied to speech recognition and summarization with very promising results initially demonstrated.

In this paper, we focus on comparison of the above-mentioned two categories of topic modeling techniques, viz. DTM and WTM, for SDR [19]. In addition to the conventional unsupervised training strategy, we explore a novel use of supervised training for estimating these models, which, by making use of a set of training query exemplars with query-document relevance information, has the merits to associate relevant documents with queries even if they do not share any of the query words. Furthermore, a pseudo-supervised training strategy (in combination with pseudo-feedback procedure) has also been studied, assuming that there is no handcrafted query-document relevance information readily available beyond the query exemplars. Moreover, to alleviate SDR performance degradation caused by imperfect speech recognition, we also utilize different levels of index features for topic modeling, including words, syllable-level units, and their combination. To our knowledge, there is still not much research on leveraging supervised (or pseudo-supervised) topic modeling techniques along with subword index features for SDR, whereas an initial attempt of combining unsupervised topic modeling and subword index features was recently investigated in [20].

It is also noteworthy that in the past decade, the use of training query exemplars and the respective query-document relevance information (or the click-through information that to some extent reflects users’ relative preferences of document relevance) has been extensively studied for supervised training of various machine-learning based retrieval models like SVM (Support Vector Machines) [21]. Furthermore, a recent trend in building retrieval systems is to use the relevance-based language model (RM) and its variants, like simple mixture model (SMM), derived from the initially retrieved documents to enhance the original query representation (or model) for better retrieval effectiveness [22, 23]. These two categories of retrieval models will be extensively evaluated and compared with our proposed models as well.

The rest of this paper is organized as follows. In Section 2, we briefly describe the mathematical formulations of the basic language model and the various topic models studied in this paper, and explain how they can fit into the retrieval purpose. Section 3 sheds light on several improved approaches proposed to work in conjunction with these topic models for SDR. Then, the experimental settings and a series of SDR experiments are presented in Sections 4 and 5, respectively. Finally, we conclude this paper and discuss avenues for future work in Section 6.

2. Language Models for SDR

2.1 Unigram Language Model (ULM)

When applying language modeling (LM) approaches to SDR, a principal realization is to use a probabilistic generative framework for ranking each document $D$ in the collection given a query $Q$, which can be expressed by $P(D|Q)$. Instead of calculating this probability directly, we apply Bayes’ rule and rewrite it as follows:

$$P(D|Q) = \frac{P(Q|D)P(D)}{P(Q)}.$$  

where $P(Q|D)$ is the probability of the query $Q$ being generated by the document $D$, $P(D)$ is the prior probability of the document $D$ being relevant, and $P(Q)$ is the prior probability of the query $Q$. $P(Q)$ in (1) can be eliminated because it is identical for all documents and will not affect the ranking of the documents. Furthermore, because the way to estimate the probability $P(D)$ is still under active study [12, 16], we may simply assume that $P(D)$ is uniformly distributed, or identical for all documents. In this way, the documents can be ranked by means of the probability $P(Q|D)$ instead of using the probability $P(D|Q)$.

Based on this, each document thereby can be treated as a probabilistic language model for generating the query. If the query $Q$ is composed of a sequence of words (or index terms), $Q = w_1 w_2 ... w_l$, where the query words are
assumed to be conditionally independent given the document $D$ and their order is also assumed to be of no importance (i.e., the so-called “bag-of-words” assumption), the relevance measure $P(q|D)$ can be further decomposed as a product of the probabilities of the query words generated by the document:

$$P(q|D) = \prod_{i=1}^{n} P(w_i|D)$$  \hspace{1cm} (2)

The document ranking problem has now been reduced to the problem of how to infer the probability distribution $P(w_i|D)$.

The simplest way to construct $P(w_i|D)$ is based on literal term matching, or using the unigram language model (ULM). Each document $D$ of the collection can respectively offer a unigram distribution for observing a query word, i.e., $P(w_i|M_D)$, where $M_D$ denotes the corresponding document model. We can simply estimate $P(w_i|M_D)$ on the basis of the frequency of the word $w_i$ occurring in the document $D$. It turns out that a document with more query words occurring frequently in it would tend to have a higher probability of generating the query. $P(w_i|M_D)$ can be further smoothed by a unigram distribution estimated from a general collection, i.e., $P(w_i|M_C)$, to model the general properties of the language as well as to avoid the problem of zero probability:

$$P_{ULM}(w_i|D) = \lambda \cdot P(w_i|M_D) + (1-\lambda) \cdot P(w_i|M_C)$$  \hspace{1cm} (3)

where $M_C$ denotes the corresponding collection model and $\lambda$ is a weighting parameter. However, how to strike the balance between these two probability distributions is actually a matter of judgment, or trial and error [12, 16, 24].

2.2 Document Topic Model (DTM)

Rather than probabilistically matching a query and a document through the index term space, the relevance between them can be estimated on the grounds of a set of latent topics. For this idea to work, each document $D$ is taken as a document topic model (DTM), consisting of a set of $K$ shared latent topics $\{T_1, … , T_K\}$ associated with document-specific weights $P(T_k|M_D)$, where each topic $T_k$ in turn offers a unigram distribution $P(w_i|T_k)$ for observing an arbitrary word of the language. For example, in the PLSA model [13], the probability of a word $w_i$ generated by a document $D$ is expressed by

$$P_{PLSA}(w_i|M_D) = \sum_{k=1}^{K} P(w_i|T_k)P(T_k|M_D)$$  \hspace{1cm} (4)

A document is believed to be more relevant to the query if the document has higher weights over some topics and the query words also happen to appear frequently in these topics.

On the other hand, LDA, having a formula analogous to PLSA (cf. (4)) for document ranking, is thought of as a natural extension to PLSA and has enjoyed much empirical success for various text IR tasks [14, 15]. LDA differs from PLSA mainly in the inference of model parameters: PLSA assumes the model parameters are fixed and unknown; while LDA places additional a priori constraints on the model parameters, i.e., thinking of them as random variables that follow some Dirichlet distributions. Since LDA has a more complex form for model optimization, which is hardly solved by exact inference, several approximate inference algorithms, such as the variational Bayes approximation [14], the Gibbs sampling algorithm [25], and the expectation propagation method [26], hence have been proposed to facilitate the estimation of the parameters of LDA according to different training strategies.

Traditionally, the DTM models (PLSA and LDA) are trained in an unsupervised way by maximizing the total log-likelihood of the document collection $D$ in terms of the unigram of all document words observed, or more specifically, the total log-likelihood of all documents generated by their own DTM models, using the expectation-maximization (EM) training algorithm [27] or other approximate inference algorithms mentioned above. For example, the PLSA model can be optimized by maximizing the following objective function using the EM algorithm:

$$\log L_D = \sum_{D \in D} \sum_{w \in D} c(w,D) \log P_{PLSA}(w|M_D)$$  \hspace{1cm} (5)

where $c(w,D)$ is the number of times that each distinct word $w$ occurs in $D$.

2.3 Word Topic Model (WTM)

Apart from treating each document in the collection as a document topic model, we can regard each word $w_j$ of the language as a word topic model (WTM) [11, 17]. To get to this point, all words are assumed to share a common set of latent topic distributions but have different weights over these topics. The WTM model of each word $w_j$ for predicting the occurrence of a particular word $w_i$ can be expressed by

$$P_{WTM}(w_i | M_j) = \sum_{k=1}^{K} P(w_i|T_k)P(T_k|M_j)$$  \hspace{1cm} (6)

where $P(w_i|T_k)$ is the probability of a word $w_i$ occurring in a specific latent topic $T_k$ and $P(T_k|M_j)$ is the probability of the topic $T_k$ conditioned on $M_j$. Then, each document can be viewed as a composite WTM, while the relevance measure between a word $w_i$ and a document $D$ can be expressed by
where $D$ is assumed to be represented as a sequence of $J$ words, i.e., $D = w_1, w_2, \ldots, w_J$; and $P(j|D)$ is the probability of randomly picking position $j$ of $D$. $P(j|D)$ is simply set to $1/J$ (i.e., each position is equally likely) in this paper, although the estimation of $P(j|D)$ would be interesting future work. Notably, the expression in (7) allows us to interpret the resulting composite WTM model for $D$ as a kind of language model that translates words $w_j$ in $D$ to $w_i$ in $Q$:

$$P_{\text{WTM}}(Q|M_D) = \prod_{j=1}^{J} P_{\text{WTM}}(w_j|M_j) P(j|D).$$  

(8)

On the other hand, the word vicinity model (WVM), bearing a certain similarity to WTM in its motivation of modeling “word-word” co-occurrences but having a more concise parameterization, has been recently proposed for speech recognition [18]. In this paper, we extend and apply it to SDR. WVM explores the word vicinity information by directly modeling the joint probability of any word pair in the language, rather than modeling the conditional probability of one word given the other word as done by WTM. In this regard, the joint probability of any word pair $(w_i, w_j)$ that describes the associated word vicinity information can be expressed by the following equation, using a set of latent topics:

$$P_{\text{WVM}}(w_i, w_j) = \sum_{k=1}^{K} P(w_i | T_k) P(T_k) P(w_j | T_k)$$  

(9)

where $P(T_k)$ is the prior probability of a given topic $T_k$. It may be noted that the relationships between words, originally expressed in a high-dimensional probability space, are now projected into a low-dimensional probability space characterized by the shared set of topic distributions. During the retrieval process, we can convert (9) into a conditional probability of a document word $w_j$ predicting a query word $w_i$, through simple mathematical manipulation:

$$P_{\text{WVM}}(w_j|M_j) = \frac{\sum_{k=1}^{K} P(w_i | T_k) P(T_k) P(w_j | T_k)}{\sum_{k=1}^{K} P(w_j | T_k) P(T_k)}.$$  

(10)

The composite WVM model for a document $D$, $P_{\text{WVM}}(Q|M_D)$, can be represented in a way similar to (7).

WTM and WVM can be trained in a data-driven manner by concatenating those words occurring within the vicinity of each occurrence of a given word $w_j$ in the document collection, which are postulated to be relevant to $w_j$. To this end, a fixed-size sliding window is placed on each occurrence of $w_j$, and a pseudo-document $O_{w_j}$ associated with such vicinity information of $w_j$ is aggregated consequently. For example, the WVM model of each word can be estimated by maximizing the total log-likelihood of words occurring in their associated “vicinity documents,” using the EM algorithm:

$$\log L_{\text{WVM}} = \sum_{w_j \in V} \sum_{w_j \in O_{w_j}} \log P_{\text{WVM}}(w_j|M_j)$$  

(11)

where $V$ denotes the predefined vocabulary and $c(w_j, O_{w_j})$ is the number of times that word $w_j$ occurs in $O_{w_j}$.

Along a similar vein to LDA, we can also extend WTM and WVM to word Dirichlet topic model (WDTM) and word Dirichlet vicinity model (WDVM), respectively. WDTM and WDVM essentially have the same ranking formula as WTM and WVM, respectively, except that they further assume that the model parameters are governed by some Dirichlet distributions.

### 3. Improved Approaches

#### 3.1 Supervised and Pseudo-supervised Training

In recent years, the use of training query exemplars and the respective query-document relevance information (or the click-through information that to some extent reflects users’ relative preferences of document relevance) has been extensively studied for training various machine-learning based retrieval models, such as those stemming from SVM (Support Vector Machines) [21]. However, to our knowledge, there is not much research on supervised training of the LM-based retrieval models, especially for the topic models. Building on this observation, we investigate supervised training of the various topic models compared in this paper. As an illustration, consider the case where the retrieval model is WVM: Given a training set of query exemplars $Q_{\text{TrainSet}}$ and the associated query-document relevance information, the WVM models can be optimized by looking for the model parameters that maximize the total log-likelihood of the query exemplars in $Q_{\text{TrainSet}}$ generated by their respective relevant documents:

$$\log L_{Q_{\text{TrainSet}}} = \sum_{Q \in Q_{\text{TrainSet}}} \sum_{D \sim D_{R(Q)}} \log P_{\text{WVM}}(Q|M_D),$$  

(12)

where $D_{R(Q)}$ denotes the set of documents that are relevant to a specific training query exemplar $Q$. Such a training approach in essence has the ability to associate a query exemplar with its relevant documents even though they do not share any of the query words. Similar treatments with supervised training can also be applied to the other topic models compared in this paper.

However, in most real-world applications, it is not always the case that the SDR systems have the training query exemplars equipped with correctly labeled relevant
and non-relevant documents available for model training. Alternatively, when such relevance/non-relevance information is not available, we may leverage pseudo-relevance feedback techniques. The top $L$-ranked documents retrieved in response to each training query exemplar are assumed to be relevant to the query exemplar, and hence are selected to accompany the query exemplar for training the various topic models. Hereafter, this kind of training scenario is referred to as pseudo-supervised training.

3.2 Subword-level Index Units

In Mandarin Chinese, there is an unknown number of words, although only some (e.g., 80 thousands, depending on the domain) are commonly used. Each word encompasses one or more characters, each of which is pronounced as a monosyllable and is a morpheme with its own meaning. Consequently, new words are easily generated every day by combining a few characters. Furthermore, Mandarin Chinese is phonologically compact; an inventory of about 400 base syllables provides full phonological coverage of Mandarin audio, if the differences in tones are disregarded. Additionally, an inventory of about 6,000 characters almost provides full textual coverage of written Chinese. There is a many-to-many mapping between characters and syllables. As such, a foreign word can be translated into different Chinese words based on its pronunciation, where different translations usually have some syllables in common, or may have exactly the same syllables.

The characteristics of the Chinese language lead to some special considerations when performing Mandarin Chinese speech recognition; for example, syllable recognition is believed to be a key problem. Mandarin Chinese speech recognition evaluation is usually based on syllable and character accuracy, rather than word accuracy. The characteristics of the Chinese language also lead to some special considerations for SDR. Word-level indexing features possess more semantic information than subword-level features; hence, word-based retrieval enhances precision. On the other hand, subword-level indexing features behave more robustly against the Chinese word tokenization ambiguity, homophone ambiguity, open vocabulary problem, and speech recognition errors; hence, subword-based retrieval enhances recall. Accordingly, there is good reason to fuse the information obtained from indexing the features of different levels [9].

In this paper, we continue this line of research by integrating subword-level information cues into topic modeling for SDR. To do this, syllable pairs are taken as the basic units for indexing besides words. Both the manual transcript and the recognition transcript of each spoken document, in form of a word stream, were automatically converted into a stream of overlapping syllable pairs. Then, all the distinct syllable pairs occurring in the spoken document collection were identified to form a vocabulary of syllable pairs for indexing. We can simply use syllable pairs, in replace of words, to represent the spoken

| Table 1. Statistics for TDT-2 and TDT-3 collections used for spoken document retrieval. |
|---|---|---|---|---|---|---|---|---|
| | TDT-2 (Development Set) | TDT-3 (Evaluation Set) |
| | 1998, 02–06 | 1998, 10–12 |
| # Spoken documents | 2,265 stories, 46.03 hours of audio | 3,371 stories, 98.43 hours of audio |
| # Distinct test queries | 16 Xinhua text stories (Topics 20001–20096) | 47 Xinhua text stories (Topics 30001–30060) |
| # Distinct training queries | 819 Xinhua text stories (Topics 20001–20096) | 731 Xinhua text stories (Topics 30001–30060) |
| Doc. length (in characters) | Min. 23, Max. 4,841, Med. 153, Mean 287.1 | Min. 19, Max. 3,667, Med. 159, Mean 415.1 |
| Length of test short query (in characters) | 8, 27, 13, 14 | 1, 56, 19, 17 |
| Length of test long query (in characters) | 183, 2,623, 329, 532.9 | 98, 1,477, 368, 443.6 |
| # Relevant documents per test query | 2, 95, 13, 29.3 | 3, 89, 12, 20.1 |
| Length of training query (in characters) | 84, 2,980, 412, 510.0 | 49, 4,112, 447, 533.7 |
| # Relevant documents per training query | 2, 95, 87, 74.4 | 3, 60, 13, 20.6 |
documents, and thereby construct the associated probabilistic latent topic distributions for the various topic models.

4. Experimental Setup

We use the Mandarin Chinese collection of the TDT corpora for the retrospective retrieval task [9], such that the statistics for the entire document collection is obtainable. The Chinese news stories (text) from Xinhua News Agency are used as our test queries (or training query exemplars) and training corpus for all topic models (excluding test query set and training query exemplars). More specifically, in the following experiments, we will either use a whole news story as a “long query,” or merely extract the title field from a news story as a “short query.” The Mandarin news stories (audio) from Voice of America news broadcasts are used as the spoken documents. All news stories are exhaustively tagged with event-based topic labels, which serve as the relevance judgments for performance evaluation. Table 1 describes some basic statistics about the corpora used in this paper. The TDT-2 collection is taken as the development set, which forms the basis for tuning the parameters in various retrieval models. The TDT-3 collection is taken as the evaluation set; that is, all the experiments performed on it were conducted following the parameter setting that was optimized based on the TDT-2 development set. Therefore, the experimental results can validate the effectiveness of the proposed approaches on comparable real-world data. The Dragon large-vocabulary continuous speech recognizer provided Chinese word transcripts for our Mandarin audio collections (TDT-2 and TDT-3). To assess the performance level of the recognizer, we spot-checked a fraction of the TDT-2 development set (about 39.90 hours) by comparing the Dragon recognition hypotheses with manual transcripts, and obtained a word error rate (WER) of 35.38%. Spot-checking approximately 76 hours of the TDT-3 test set gave a WER of 36.97%. Since Dragon’s lexicon is not available, we augmented the LDC Mandarin Chinese Lexicon with 24k words extracted from Dragon’s word recognition output, and for computing error rates used the augmented LDC lexicon (about 51,000 words) to tokenize the manual transcripts. We also used this augmented LDC lexicon to tokenize the query sets (including training and test sets) and training corpus in the retrieval experiments.

The retrieval results, assuming manual transcripts for the spoken documents to be retrieved (denoted TD, text documents, in the tables below) are known, are also shown for reference, compared to the results when only the erroneous transcripts by speech recognition are available (denoted SD, spoken documents, in the tables below). The retrieval results are expressed in terms of non-interpolated mean average precision (mAP) following the TREC evaluation [5], which is computed by the following equation:

\[
mAP = \frac{1}{L} \sum_{i=1}^{L} \frac{1}{N_i} \sum_{j=1}^{r_{i,j}} \frac{P(D_{iwP} | Q_i)}{N_w}
\]

where \( L \) is the number of test queries, \( N_i \) is the total number of documents that are relevant to query \( Q_i \), and \( r_{i,j} \) is the position (rank) of the \( j \)-th document that is relevant to query \( Q_i \), counting down from the top of the ranked list. When topic models are employed in evaluating the relevance between a query word \( w_i \) and a document \( D \) (e.g., \( P_{Topic}(w_i | D) \)), we additionally incorporate the unigram probabilities of \( w_i \) in the document \( P(w_i | M_D) \) and a general text corpus \( P(w_i | M_C) \) for better retrieval performance:

\[
P(w_i | D) = \alpha \left[ \beta \cdot P_{Topic}(w_i | D) + (1 - \beta) \cdot P(w_i | M_D) \right] + (1 - \alpha) \cdot P(w_i | M_C)
\]

This is because words represented in a latent topic space only offer coarse-grained concept cues about the information need at the expense of losing the discriminative power among concept-related words in finer granularity. Similar treatments also have been studied for the PLSA- and LDA-based [28] retrieval models for text document retrieval. Note also that as in our previous studies [20] all the topic models to be compared below have the same number of latent topics which is set to 32.

5. Experimental Results and Analysis

In this section, we begin by comparing the retrieval effectiveness of different topic models from two disparate angles, viz. query styles (long queries vs. short queries) and index terms (words vs. syllable pairs), with the commonly-used unsupervised training approach. Then, we reports on experiments relating to our two proposed training approaches for topic models, viz. supervised training and pseudo-supervised training. At the end, we validate the utility of fusing word- and syllable-level index terms, in conjunction with the various topic models and training approaches.

5.1 Unsupervised Training

We first evaluate the ULM model and the various topic models that are trained in an unsupervised manner without recourse to the training query exemplar set and the corresponding query-document relevance information. The results when using different types of queries (viz. long or short queries) and different kinds of index features (viz. word- or syllable-level index terms) are shown in Tables 2, 3, 4, and 5, respectively. Note that all the training settings,
model complexities and interpolation weights (e.g., $\alpha$ and $\beta$ in (14)) are tuned or optimized by using the TDT-2 development set and tested on both the TDT-2 development set and the TDT-3 evaluation set.

Consulting the results, at first glance, it seems that all topic models are competitive to each other. However, if we have a close look at these results, we notice five particularities. First, although the word error rate (WER) for the spoken document collection is higher than 35%, it does not lead to catastrophic failures probably due to the reason that recognition errors are overshadowed by a large number of spoken words correctly recognized in the documents.

Second, the various topic models cannot always yield improvements over ULM on the evaluation set, which is in part due to the fact that index terms represented in topic models only offer coarse-grained concept cues about a document, which would probably lose the fine-grained discrimination capabilities among concept-related index terms, as already mentioned.

Third, when the test queries are (relatively) too short to sufficiently express the information needs, the syllable-level index features appear to provide the retrieval quality competitive to the word-level index features in most cases (cf. the retrieval results on the evaluation set in Tables 2 and 4). The performance of the former, instead, show the retrieval performance only on par with or even inferior to that of the latter when dealing with the long (verbose) test queries. We conjecture this is because for a given short query, extending the underlying retrieval model with the functionality of partial matching in either the literal-term space or the latent topic space has practical significance in helping retrieve more spoken documents likely to be relevant to the query.

Fourth, WTM and its variants (i.e., WVM, WDTM, and WDVM) seem to yield less pronounced results on the evaluation set as compared to the DTM models (i.e., PLSA and LDA). This may have stemmed from the fact that, the component word topic models or word vicinity models of each spoken document were trained beforehand using the training corpus in this paper; on the contrary, for DTM, the model parameters have to be updated on-the-fly by maximizing the document likelihood. For example, both the probabilities $P(w_j|T_k)$ and $P(T_k|M_f)$ used to construct

---

### Table 2. Retrieval results (in mAP) for short queries with unsupervised training and the word-level index features.

| Unsupervised | ULM | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|-----|------|-----|-----|-----|------|------|
| Dev. Set     | TD  | 0.371| 0.418| 0.401| 0.400| 0.401| 0.403|
| (TDT-2)      | SD  | 0.320| 0.345| 0.341| 0.341| 0.343| 0.345|
| Eval. Set    | TD  | 0.432| 0.439| 0.449| 0.427| 0.432| 0.430|
| (TDT-3)      | SD  | 0.363| 0.413| 0.417| 0.400| 0.401| 0.403|

### Table 3. Retrieval results (in mAP) for long queries with unsupervised training and the word-level index features.

| Unsupervised | ULM | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|-----|------|-----|-----|-----|------|------|
| Dev. Set     | TD  | 0.634| 0.645| 0.643| 0.644| 0.649| 0.644|
| (TDT-2)      | SD  | 0.563| 0.582| 0.581| 0.572| 0.583| 0.576|
| Eval. Set    | TD  | 0.653| 0.660| 0.658| 0.652| 0.654| 0.650|
| (TDT-3)      | SD  | 0.631| 0.634| 0.632| 0.628| 0.614| 0.625|

### Table 4. Retrieval results (in mAP) for short queries with unsupervised training and the syllable-level index features.

| Unsupervised | ULM | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|-----|------|-----|-----|-----|------|------|
| Dev. Set     | TD  | 0.352| 0.438| 0.442| 0.400| 0.397| 0.403|
| (TDT-2)      | SD  | 0.329| 0.419| 0.413| 0.357| 0.357| 0.360|
| Eval. Set    | TD  | 0.452| 0.474| 0.466| 0.458| 0.464| 0.459|
| (TDT-3)      | SD  | 0.391| 0.450| 0.444| 0.425| 0.428| 0.427|

### Table 5. Retrieval results (in mAP) for long queries with unsupervised training and the syllable-level index features.

| Unsupervised | ULM | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|-----|------|-----|-----|-----|------|------|
| Dev. Set     | TD  | 0.593| 0.635| 0.635| 0.616| 0.618| 0.615|
| (TDT-2)      | SD  | 0.570| 0.613| 0.615| 0.590| 0.591| 0.591|
| Eval. Set    | TD  | 0.682| 0.677| 0.648| 0.652| 0.652| 0.653|
| (TDT-3)      | SD  | 0.667| 0.627| 0.626| 0.640| 0.642| 0.643|
WTM for the evaluation set were directly adopted from that of the development set, as opposed to PLSA where $D_kTP^M$ for the evaluation set had to be re-estimated using the EM algorithm.

A final observation is that, despite that LDA has some known theoretical advantages over PLSA, they tend to perform on par with each other for the SDR task studied here, which is in line with the recent results reported in [29] for text IR. An analogous reasoning also holds for WTM vs. WVM and WDTM vs. WDVM. From now on, unless otherwise stated, the retrieval results reported were obtained by using the syllable-level index terms.

### 5.2 Supervised Training

Next, we perform experiments that simulate a scenario in which a set of training query exemplars and the corresponding query-document relevance information (or the click-through information that to some extent reflects users’ relative preferences of document relevance) can be utilized to boost the performance of the various topic models (cf. Sec. 3.1). 819 training query exemplars with the corresponding query-document relevance information are compiled for the development set, while 731 training query exemplars with the corresponding query-document relevance information were collected for the evaluation set. From the retrieval results shown in Tables 6 and 7, several observations can be made. First, supervised training can significantly boost the performance of various topic models as compared to the results of unsupervisedly trained topic models and SVM shown in Tables 6 and 7. Second, WTM and its variants (i.e., WVM, WDTM, and WDVM) seem to perform better than or at least as well as the DTM models (i.e., PLSA and LDA) in the supervised training case. The performance gap between the WTM and DTM models tends to increase when going from short queries to long queries. Third, we found that WTM (and its variants) had larger values of $\alpha$ and $\beta$ (cf. equation (14)) compared to PLSA and LDA in both the unsupervised and supervised training settings. It seems to indicate that document ranking relies more on the topic cues provided by WTM (and its variants) than that provided by PLSA and LDA when

| Table 6. Retrieval results (in mAP) for short queries with supervised training and the syllable-level index features. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Supervised     | PLSA | LDA | WTM | WVM | WDTM | WDVM | SVM |
| Dev. Set       |       |       |     |     |     |     |     |
| (TDT-2)        | TD   | 0.546| 0.540| 0.624| 0.667| 0.625| 0.623| 0.427|
|                | SD   | 0.534| 0.519| 0.605| 0.650| 0.629| 0.615| 0.407|
| Eval. Set      |       |       |     |     |     |     |     |
| (TDT-3)        | TD   | 0.522| 0.516| 0.549| 0.528| 0.509| 0.579| 0.468|
|                | SD   | 0.494| 0.499| 0.538| 0.489| 0.464| 0.529| 0.457|

| Table 7. Retrieval results (in mAP) for long queries with supervised training and the syllable-level index features. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Supervised     | PLSA | LDA | WTM | WVM | WDTM | WDVM | SVM |
| Dev. Set       |       |       |     |     |     |     |     |
| (TDT-2)        | TD   | 0.701| 0.695| 0.828| 0.844| 0.848| 0.843| 0.621|
|                | SD   | 0.693| 0.682| 0.806| 0.831| 0.831| 0.828| 0.601|
| Eval. Set      |       |       |     |     |     |     |     |
| (TDT-3)        | TD   | 0.720| 0.714| 0.803| 0.810| 0.802| 0.793| 0.655|
|                | SD   | 0.727| 0.709| 0.782| 0.786| 0.782| 0.781| 0.631|

| Table 8. Retrieval results (in mAP) for short queries with pseudo-supervised training and the syllable-level index features. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Pseudo-supervised | PLSA | LDA | WTM | WVM | WDTM | WDVM | RM | SMM |
| Dev. Set       |       |       |     |     |     |     |    |    |
| (TDT-2)        | TD   | 0.458| 0.460| 0.426| 0.430| 0.435| 0.429| 0.415| 0.403|
|                | SD   | 0.442| 0.432| 0.403| 0.396| 0.405| 0.398| 0.381| 0.377|
| Eval. Set      |       |       |     |     |     |     |    |    |
| (TDT-3)        | TD   | 0.481| 0.477| 0.460| 0.462| 0.471| 0.467| 0.524| 0.460|
|                | SD   | 0.465| 0.456| 0.444| 0.431| 0.442| 0.433| 0.508| 0.413|

| Table 9. Retrieval results (in mAP) for long queries with pseudo-supervised training and the syllable-level index features. |
|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Pseudo-supervised | PLSA | LDA | WTM | WVM | WDTM | WDVM | RM | SMM |
| Dev. Set       |       |       |     |     |     |     |    |    |
| (TDT-2)        | TD   | 0.636| 0.639| 0.628| 0.629| 0.632| 0.627| 0.630| 0.630|
|                | SD   | 0.623| 0.633| 0.601| 0.601| 0.604| 0.601| 0.579| 0.584|
| Eval. Set      |       |       |     |     |     |     |    |    |
| (TDT-3)        | TD   | 0.665| 0.664| 0.672| 0.676| 0.671| 0.683| 0.677| 0.692|
|                | SD   | 0.667| 0.668| 0.666| 0.667| 0.665| 0.669| 0.668| 0.679|
performing topic matching between the query and the spoken documents.

5.3 Pseudo-supervised Training

In the third set of experiments, we investigate pseudo-supervised training of topic models. Here we assume the query-document relevance information of the training query exemplars is not readily available. A natural solution to overcome this limitation is to conduct a run of retrieval and take the top-ranked documents in response to each training query exemplar as the pseudo-relevant documents of the query for training the topic models subsequently. By doing so, it is expected that we can steadily refine a deployed SDR system without user interference.

Some conventional IR models, such as relevance-based language model (RM) and simple mixture model (SMM) [22, 23], also employ an initial retrieval to expand the query representation. Both RM and SMM treat each test query as a query model (instead of an observation consisting of a sequence of index terms) and compute the

| Unsupervised | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|------|-----|-----|-----|------|------|
| Dev. Set TD  | 0.468| 0.461| 0.434| 0.435| 0.435| 0.432|
| (TDT-2) SD   | 0.465| 0.453| 0.400| 0.388| 0.393| 0.389|
| Eval. Set TD | 0.490| 0.496| 0.468| 0.473| 0.467| 0.470|
| (TDT-3) SD   | 0.469| 0.471| 0.449| 0.448| 0.444| 0.449|

| Supervised   | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|------|-----|-----|-----|------|------|
| Dev. Set TD  | 0.592| 0.579| 0.709| 0.733| 0.736| 0.740|
| (TDT-2) SD   | 0.547| 0.528| 0.609| 0.655| 0.639| 0.641|
| Eval. Set TD | 0.539| 0.538| 0.569| 0.596| 0.592| 0.556|
| (TDT-3) SD   | 0.539| 0.533| 0.541| 0.513| 0.556| 0.491|

| Supervised   | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|--------------|------|-----|-----|-----|------|------|
| Dev. Set TD  | 0.716| 0.714| 0.884| 0.898| 0.906| 0.906|
| (TDT-2) SD   | 0.704| 0.693| 0.844| 0.866| 0.876| 0.868|
| Eval. Set TD | 0.719| 0.729| 0.801| 0.808| 0.798| 0.805|
| (TDT-3) SD   | 0.740| 0.732| 0.790| 0.789| 0.794| 0.790|

| Pseudo-supervised | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|-------------------|------|-----|-----|-----|------|------|
| Dev. Set TD       | 0.480| 0.484| 0.466| 0.468| 0.480| 0.468|
| (TDT-2) SD        | 0.458| 0.455| 0.429| 0.430| 0.436| 0.427|
| Eval. Set TD      | 0.501| 0.501| 0.475| 0.478| 0.484| 0.478|
| (TDT-3) SD        | 0.482| 0.477| 0.452| 0.450| 0.456| 0.453|

| Pseudo-supervised | PLSA | LDA | WTM | WVM | WDTM | WDVM |
|-------------------|------|-----|-----|-----|------|------|
| Dev. Set TD       | 0.665| 0.668| 0.674| 0.674| 0.675| 0.673|
| (TDT-2) SD        | 0.628| 0.633| 0.623| 0.630| 0.627| 0.633|
| Eval. Set TD      | 0.679| 0.679| 0.666| 0.658| 0.666| 0.666|
| (TDT-3) SD        | 0.679| 0.670| 0.635| 0.663| 0.635| 0.643|
probabilistic model distance between the query and document models for document ranking [12]. At query time, the original query model (rather than the document models) is automatically modified based on the statistical evidence of the top-ranked documents returned by the first run of retrieval. The retrieval performance of RM and SMM in the second retrieval is anticipated to be improved. From this vein, pseudo-supervised training of topic models can be considered kind of document model adaptation for retrieval.

The retrieval results of the topic models trained in the pseudo-supervised manner are shown in Tables 8 and 9. The results for RM and SMM are also listed for reference. For all topic models, we selected the top 3 retrieved documents by the ULM model for pseudo-supervised training. The number of pseudo-relevant documents retrieved from the local feedback-like procedure (through ULM) for RM and SMM is 15. Comparing Tables 8 and 9 to Tables 4 and 5, we can find that, in most cases, the topic models trained in the pseudo-supervised manner outperform their counterpart models trained in the unsupervised manner. The DTM models seem to benefit more than the WTM models when using pseudo-supervised training. Further, the results show the superiority of the topic models over RM and SMM for SD case for the development set, whereas RM and SMM perform better than the topic models on the evaluation set for most cases. Further investigation is needed to determine the underlying cause.

5.4 Fusion of Different Levels of Indexing Features

As a final set of experiments, we explore how the word- and syllable-level index features complement each other. The results are shown in Tables 10 to 15, as a function of different types of queries (viz. long or short queries) and different kinds of training approaches (viz. unsupervised, supervised, or pseudo-supervised training) being used. As can be seen from these tables, the results have consistent trends with that of the previous experiments. First, not surprisingly, compared to the results of using either the word- or syllable-level index features alone, the fusion of different levels of index features can integrate their advantages to achieve better performance. Second, supervised training outperforms pseudo-supervised training and unsupervised training. Third, although the results show that WTM and its variants are worse than PLSA and LDA in the unsupervised training case, WTM still has its merit of efficient model construction for an unseen document than PLSA and LDA for a retrieval task. From the results, we can conclude that fusion works well for both the TD and SD cases, in all different training settings, and for different query types.

6. Conclusions

In this paper, we have conducted a series of comparisons among several topic models for SDR. The experimental results have shown the good potential of leveraging the supervised and pseudo-supervised training strategies for topic modeling. Furthermore, the various WTM models have also been demonstrated to work effectively in the SDR task. As to future work, we envisage the following three directions: 1) utilizing speech summarization techniques to help better estimate the query and document models [26], 2) integrating the document model with other more elaborate representations of the speech recognition output [10], and 3) further confirming our observations on larger-scale experiments.

Acknowledgements

This work was sponsored in part by "Aim for the Top University Plan" of National Taiwan Normal University and Ministry of Education, Taiwan, and the National Science Council, Taiwan, under Grants NSC 100-2515-S-003-003, NSC 99-2221-E-003-017-MY3, NSC 99-2631-S-003-002 and NSC 98-2221-E-003-011-MY3.

References

[1] L. S. Lee and B. Chen, “Spoken document understanding and organization,” IEEE Signal Processing Magazine, 22(5), pp. 42–60, 2005.
[2] M. Ostendorf, “Speech technology and information access,” IEEE Signal Processing Magazine, 25 (3), pp. 150–152, 2008.
[3] C. Chelba, T. J. Hazen, and Murat Saraclar, “Retrieval and browsing of spoken content,” IEEE Signal Processing Magazine, 25(3), pp. 39–49, 2008.
[4] C. D. Manning, P. Raghavan, and H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008.
[5] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval: The Concepts and Technology behind Search, ACM Press, 2011.
[6] J. Garofolo, G. Auzanne, , and E. Voorhees, “The TREC spoken document retrieval track: A success story,” in Proc. the 8th Text Retrieval Conference, NIST, pp. 107–129, 2000.
[7] J. M. Ponte and W. B. Croft, “A language modeling approach to information retrieval,” in Proc. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 275–281, 1998.
[8] D. R. H. Miller, T. Leek, and R. M. Schwartz, “A hidden Markov model information retrieval System,” in Proc. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 214–221, 1999.
[9] B. Chen, H. M. Wang, and L. S. Lee, “A discriminative HMM/n-gram-based retrieval approach for Mandarin spoken documents,”
11

ACM Transactions on Asian Language Information Processing, 3(2), pp. 128–145, 2004.

[10] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, “Statistical lattice-based spoken document retrieval,” ACM Transactions on Information Systems, 28(1), pp. 2:1–2:30, 2010.

[11] B. Chen, “Word topic models for spoken document retrieval and transcription,” ACM Transactions on Asian Language Information Processing, 8(1), pp. 2:1–2:27, March 2009.

[12] C. X. Zhai, “Statistical language models for information retrieval: A critical review,” Foundations and Trends in Information Retrieval, 2(3), 137–213, 2008.

[13] T. Hoffmann, “Unsupervised learning by probabilistic latent semantic analysis,” Machine Learning, 42, pp. 177–196, 2001.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet allocation,” Journal of Machine Learning Research, 3, pp. 993–1022, January 2003.

[15] D. M. Blei and J. Lafferty, “Topic models,” in A. Srivastava and M. Sahami, (eds.), Text Mining: Theory and Applications. Taylor and Francis, 2009.

[16] W. B. Croft and J. Lafferty (eds.), Language Models for Information Retrieval, Kluwer International Series on Information Retrieval, Volume 13, Kluwer Academic Publishers, 2002.

[17] B. Chen, “Latent topic modeling of word co-occurrence information for spoken document retrieval,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 3961–3964, 2009.

[18] K. Y. Chen, H. S. Chiu, and B. Chen, “Latent topic modeling of word vicinity information for speech recognition,” in Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 5394–5397, 2010.

[19] K. Y. Chen and B. Chen, “A study of topic modeling techniques for spoken document retrieval,” in Proc. AFISPA Annual Summit and Conference, pp. 237–242, 2010.

[20] S. H. Lin and B. Chen, “Topic modeling for spoken document retrieval using word- and syllable-level information,” in Proc. the workshop on Searching Spontaneous Conversational Speech, pp. 3–10, 2009.

[21] R. Nallapati, “Discriminative models for information retrieval,” in Proc. ACM Conference on Research and Development in Information Retrieval, pp. 64–71, 2004.

[22] Y. Lv and C. X. Zhai, “A comparative study of methods for estimating query language models with pseudo feedback,” in Proc. ACM conference on Information and knowledge management, pp. 1895–1898, 2009.

[23] P. N. Chen, K. Y. Chen, B. Chen, “Leveraging relevance cues for improved spoken document retrieval,” in Proc. the Annual Conference of the International Speech Communication Association, 2011.