7-2006

Extraction of Coherent Relevant Passages using Hidden Markov Models

Jing JIANG
Singapore Management University, jingjiang@smu.edu.sg

ChengXiang ZHAI
University of Illinois at Urbana-Champaign

DOI: https://doi.org/10.1145/1165774.1165775

Citation

JIANG, Jing and ZHAI, ChengXiang. Extraction of Coherent Relevant Passages using Hidden Markov Models. (2006). *ACM Transactions on Information Systems*. 24, (3), 295-319. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/130

Extraction of Coherent Relevant Passages Using Hidden Markov Models

JING JIANG and CHENGXIANG ZHAI
University of Illinois

In information retrieval, retrieving relevant passages, as opposed to whole documents, not only directly benefits the end user by filtering out the irrelevant information within a long relevant document, but also improves retrieval accuracy in general. A critical problem in passage retrieval is to extract *coherent relevant passages* accurately from a document, which we refer to as *passage extraction*. While much work has been done on passage retrieval, the passage extraction problem has not been seriously studied. Most existing work tends to rely on presegmenting documents into fixed-length passages which are unlikely optimal because the length of a relevant passage is presumably highly sensitive to both the query and document.

In this article, we present a new method for accurately detecting coherent relevant passages of variable lengths using hidden Markov models (HMMs). The HMM-based method naturally captures the topical boundaries between passages relevant and nonrelevant to the query. Pseudo-feedback mechanisms can be naturally incorporated into such an HMM-based framework to improve parameter estimation. We show that with appropriate parameter estimation, the HMM method outperforms a number of strong baseline methods on two datasets. We further show how the HMM method can be applied on top of any basic passage extraction method to improve passage boundaries.

Categories and Subject Descriptors: H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms: Algorithms

Additional Key Words and Phrases: Hidden Markov models, passage retrieval

1. INTRODUCTION

Traditional information retrieval systems return a ranked list of whole documents as the answer to a query. However, in many cases, not every part of an entire document is relevant to the query. Thus, it is desirable to retrieve only relevant passages, as opposed to whole documents, which in effect helps to filter out irrelevant information in a long relevant document.

A critical problem in passage retrieval is to accurately locate the boundaries of *coherent relevant passages* in a document, which we refer to as *passage*

This work was supported in part by the National Science Foundation under Award Numbers 0425852, 0347933, and 0428472.

Authors' address: J. Jiang and C. Zhai, Department of Computer Science, University of Illinois at Urbana-Champaign, 201 N. Goodwin Ave., Urbana, IL 61801; email:{jiang4, czhai}@.uiuc.edu.

extraction. Indeed, passage retrieval generally involves two components: passage extraction and passage ranking. Given a query, we can either first rank documents in their entirety, and then extract relevant passages from the retrieved documents, or we can first extract the most likely relevant passages from all documents, and then rank the extracted passages. No matter which order we take, passage extraction is clearly an important element of passage retrieval, and optimal passage retrieval performance requires accurate passage extraction.

In addition to allowing an information retrieval system to precisely point to the most relevant parts of a document, extracting query-specific relevant passages also has the following benefits: (1) It allows us to score a document based on its most relevant topical segment, which presumably is more accurate; indeed, previous work has shown that retrieval performance can be improved by exploiting passage-level evidence [Salton et al. 1993; Callan 1994; Kaszkiel and Zobel 1997, 2001; Liu and Croft 2002]. (2) When the user provides examples of relevant documents, using relevant topical segments in these documents for relevance feedback and query expansion is presumably more accurate than using entire documents, since a whole document may contain nonrelevant information. (3) In some distributed network environments where the bandwidth is limited, such as wireless networks, retrieving relevant topical segments rather than whole relevant documents can significantly reduce the amount of data delivered to the user from the server, and thus the overall communication cost.

Despite its importance, however, the passage extraction problem has not been seriously addressed in existing work. Indeed, to the best of our knowledge, no direct evaluation of the accuracy of passage extraction methods has ever been made. Passage retrieval methods have so far been evaluated for one of the following three tasks: traditional document ranking [Salton et al. 1993; Callan 1994; Mittendorf and Schäuble 1994; Kaszkiel and Zobel 1997, 2001; Denoyer and Zaragoza 2001; Liu and Croft 2002], passage ranking, as in the TREC HARD track [Allan 2003], and question answering [Clarke and Terra 2003; Tellex et al. 2003; Corrada-Emmanuel and Croft 2004]. However, since the ranking is the main component being evaluated in all these tasks, such evaluation does not directly help us understand how effective these methods are for passage extraction. For example, any passage covering the answer to a question would be equally good for question answering, but various passages clearly differ in both coherence and overall coverage of the relevant information from the perspective of passage extraction.

The lack of attention to passage extraction is also reflected in the fact that none of the existing passage retrieval methods was intentionally designed to achieve the goal of extracting passages that are both query-dependent and coherent. For example, methods such as TextTiling [Hearst 1997] segment text into coherent passages by automatically detecting topic shifts, but the passage boundaries identified by these methods are not query-specific. The same problem exists with many window-based passage retrieval methods, which presegment documents into passages of fixed length without considering the specific query. Some other methods do attempt to extract query-specific variable-length passages, but they do not consider their coherence (e.g., Cormack et al. [1998], Kaszkiel and Zobel [2001]). More discussion on previous work is given in Section 7.

In this article , we directly address the passage extraction problem, which we define as detecting the boundaries of the most relevant and coherent passages from relevant documents. While a relevant document may contain multiple relevant passages, we choose to focus first on studying how to accurately extract the single most relevant passage from each document. A good understanding of this simpler case is necessary before we can effectively address more complicated situations. Moreover, once we know how to extract the most relevant passage, we can iteratively apply the same method to extract any additional relevant passages by working on the rest of the document after taking out the most relevant passages. We present a new method that uses hidden Markov models (HMMs) for accurately detecting coherent query-specific relevant passages of variable lengths. We study how to design the structure of the HMMs and propose three different methods for estimating their parameters. Evaluation on two datasets shows that with appropriate parameter estimation, the HMM method outperforms a number of strong baseline methods on both datasets. We further show that the HMM method can be applied on top of any basic passage extraction method to improve passage boundaries.

The rest of the article is organized as follows. In Section 2, we discuss the challenges in the passage extraction problem. We then introduce our HMM-based passage extraction method and basic HMM structure in Section 3. We discuss refinement of the structure and parameter estimation in Section 4. In Sections 5 and 6, we present our experiment design and results. In Section 7, we discuss some related work. Finally, in Section 8, we conclude our work, and envision future research directions.

2. CHALLENGES IN PASSAGE EXTRACTION

As we stated in Section 1, passage extraction aims at extracting coherent and query-specific relevant passages. As such, there are certain challenges in the passage extraction problem that previous passage retrieval methods have not dealt with.

2.1 Address Length Variation

The passages we look for can be of various lengths. Because of the differences between documents in topic, content, style, etc., the length of a coherent passage is document-dependent. Moreover, for different queries, the passage length within the same document can also vary. In Table I, we show two documents in the HARD04 dataset, which is one of two datasets we use and is described in Section 5.2. Both documents contain relevant passages for two topics: "video game crash," and "hand-held electronics." The underlined paragraphs are passages relevant to "video game crash," and those in bold font are relevant to "hand-held electronics." These are the true relevant passages, according to human annotations. We can see that in the first document, the passage relevant to "video game crash" covers its entirety, while the one relevant to "hand-held electronics" covers only the middle part. In the second document, however, the

Table I. Passage Length Variation

APE20030922.0156	APE20030911.0887
Nokia, the world's biggest cell phone maker, said Monday it acquired Sega.com—a subsidiary of Japanese video game maker, Sega Corp.—to improve its online game and services. **The takeover, completed Sept. 16, means that Nokia Nokia will use Sega.com Inc.'s multiplayer technology in its mobile N-Gage game deck that features 3D multiplayer gameplay using Bluetooth wireless technology and GPRS.** **Nokia has slated Oct. 7 for the worldwide launch of its N-Gage mobile phone that combines the features of a cell phone, MP3-player and a gaming deck.** Nokia is the cell phone market leader with about 36 percent of all mobile phones sold worldwide, according to Gartner Dataquest. Last year, Nokia claimed a 38 percent market share.	**Nintendo Co.'s Game Boy Advance hand-held machine now works as a videophone with an attachment that comes with a digital camera, earphone and microphone.** **The 13,000 yen (US$110) Campho Advance from Kyoto-based Digital Act Co., which makes mobile and Internet equipment, slips into the top of the Game Boy Advance just like any video-game cassette.** **When connected to an analog telephone outlet, the display shows live video of the person on the other end of the line, who must also own both the Game Boy Advance and the Campho Advance. Your own image will show up in the corner of the display.** Campho Advance, which goes on sale only in Japan in December, requires no Internet service provider. Developers are working on a broadband device but have no plans to sell the product overseas so far, company spokesman Kazuhisa Saito said Friday. Nintendo has sold more than 10 million Game Boy Advance machines in Japan, and about 34 million worldwide.

passage relevant to "video game crash" covers only a short paragraph, while the passage relevant to "hand-held electronics" covers about two-thirds of the document. This example shows that passage length is both document and query-dependent.

Window-based passages are neither document nor query-dependent. Passages based on topic segmentation, although document-dependent, are not query-dependent. Thus, presegmenting documents into passages, as many existing methods do, cannot address the length variation problem.

2.2 Exploit Coherence

In general, the relevant passages we are to extract will be coherent in content, and the passage boundary is likely to be located where the coherence breaks. To thus improve the accuracy of passage extraction, we should not only look at the passages themselves, but also consider the surrounding text to find such boundaries of coherence. However, to the best of our knowledge, coherence has not been exploited in most existing work on extracting query-dependent relevant passages.

To address the issues of both length variation and coherence we propose a hidden Markov model-based method to extract query-dependent relevant passages. The HMM-based method is designed to naturally extract variable-length passages based on coherence.

3. A BASIC HMM FOR PASSAGE EXTRACTION

The task of passage extraction can be formulated as follows. Given a query q and a document d that is relevant or likely to be relevant to q, find those text segments from d that are coherent and most relevant to q. In this article, we propose a method that is based on unigram language models. We therefore ignore the structural markups in d such as sentence and paragraph boundaries, and represent d as a sequence of words, that is, $d = (w_1, w_2, \ldots, w_n)$. Given query q, we are to find the subsequences of d that are most relevant to q.

In the language modeling approach to information retrieval, a document is often treated as a bag of words, where the words are considered to be a sample drawn from a unigram language model (i.e., a multinomial word distribution). For passage extraction, however, a document should be treated as a sequence, rather than a bag of words, and relevant segments have a different language model than nonrelevant ones. Thus, the document can be modeled as being sequentially generated from two language models, that is, the relevant segments are generated from what we call a relevance language model, while the nonrelevant segments are generated from what we call the background language model. The use of these two language models allows us to naturally model coherence in the text.

There is, however, another stochastic process that determines when the language model switches from the relevance to the background model, and vice versa, during the generation of the document. This stochastic process is hidden from us, but allows us to address the variable length issue in a principled way. Note that this sequence of transitions between the relevance and background model, as generated by the hidden stochastic process, is exactly what we want to discover because this sequence shows where the document shifts between relevant and nonrelevant segments. This doubly embedded stochastic process is essentially a hidden Markov model.

3.1 Hidden Markov Models

Informally, a hidden Markov model is just a "stochastic machine" that can stochastically generate a symbol at each discrete time-point. The symbol is generated from an internal state of the HMM, according to a conditional distribution of the symbol, given the state. As time evolves, an HMM changes its state according to a state-transition distribution. Formally, an HMM is characterized by a set of hidden states, a set of observable output symbols, an initial state-probability distribution, a state-transition probability distribution for each state, and an output-probability distribution for each state. A sequence of output symbols is stochastically generated from a sequence of unobservable states, which itself is generated from the hidden stochastic process that produces state transitions.

Given an observed sequence of output symbols, we often want to find the sequence of hidden states that is the most likely to have generated the observations. The Viterbi algorithm is a dynamic programming algorithm that can efficiently solve this problem. Another common issue is how to estimate the output probabilities and state-transition probabilities, given a sequence of

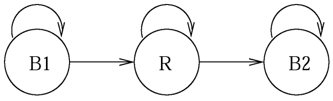

Fig. 1. Basic HMM structure.

output symbols. The Baum-Welch algorithm, essentially an EM algorithm, is a practical solution to this unsupervised learning problem. The Baum-Welch algorithm can also be used to estimate some of the parameters while other parameters are fixed. For example, we can fix the output probabilities and only estimate the transition probabilities from a set of observed sequences by using the Baum-Welch algorithm. Rabiner [1989] gives a good tutorial on HMMs.

3.2 Basic HMM Structure

Figure 1 shows a three-state hidden Markov model constructed to model the generation of a document with a single relevant passage. To force the document to contain only one relevant passage, this linear HMM has a relevance state R between two background states B_1 and B_2. The set of output symbols is the set of words in the document collection. The output distribution at state R is the relevance language model, and the output distributions at states B_1 and B_2 are the background language model. The arrows in Figure 1 indicate nonzero transition probabilities.

Formally, let $\mathcal{B}$ denote the background language model for states B_1 and B_2, and $\mathcal{R}$ denote the relevance language model for state R. Let $p(S_2|S_1)$ denote the transition probability from state S_1 to state S_2, and $p(S_1)$ denote the initial probability of state S_1, where $S_1, S_2 \in \{B_1, B_2, R\}$. Given a document $d = w_1 w_2 \ldots w_n$, we want to find the state sequence S^* that has generated d with the highest probability.

$$S^* = \arg\max_{S=S_1S_2\ldots S_n} p(S_1)p(w_1|S_1)\prod_{i=1}^{n-1} p(S_{i+1}|S_i)p(w_{i+1}|S_{i+1}), \qquad (1)$$

where $S_i \in \{B_1, B_2, R\}$ for $i = 1, \ldots, n$. Because of the structure of the three-state HMM, the state sequence S^* must be of the form $B_1 \ldots B_1 R \ldots R B_2 \ldots B_2$, unless it stops at state R or state B_1, in which case the state sequence is of the form $B_1 \ldots B_1 R \ldots R$ or $B_1 \ldots B_1$. We will show in Section 4 how we handle the case in which the state sequence stops at state R or B_1.

3.3 Parameter Estimation

In order to find the most likely state sequence of observed output symbols, we need to set the various parameters in the HMM structure. We now discuss how we estimate these parameters. The idea is to use fixed output probabilities, estimated from either the document collection or the query, and to train the transition probabilities from single documents.

3.3.1 *Output Probabilities.* The output probabilities at background states B_1 and B_2 are specified by the background language model $\mathcal{B}$. Estimating the background language model is easy. We can simply use the collection language model $\mathcal{C}$ to approximate $\mathcal{B}$. Let $W = \{w_1, w_2, \ldots, w_m\}$ be the set of words in the text collection C. Using maximum-likelihood estimation, we can estimate the background language model as follows:

$$p(w_i|\mathcal{B}) = p(w_i|\mathcal{C}) = \frac{c(w_i, C)}{\sum_{j=1}^{m} c(w_j, C)}, \tag{2}$$

where $c(w_i, C)$ is the number of times word w_i appears in the document collection C.

The output probabilities at state R are specified by the relevance language model. To estimate this model, we start with the query language model. Let $\mathcal{Q}$ denote the query language model. We can estimate $\mathcal{Q}$ using maximum-likelihood estimation, as follows:

$$p(w_i|\mathcal{Q}) = \frac{c(w_i, q)}{\sum_{j=1}^{m} c(w_j, q)}, \tag{3}$$

where $c(w_i, q)$ is the number of times the word w_i appears in query q. Since the relevant passage also contains nonquery words, we need to smooth this query language model with the background language model to construct the relevance language model. One choice is the Jelinek-Mercer smoothing method [Zhai and Lafferty 2001b]:

$$p(w_i|\mathcal{R}) = \lambda p(w_i|\mathcal{Q}) + (1 - \lambda) p(w_i|\mathcal{B}), \tag{4}$$

where λ is a parameter that needs to be tuned empirically.

3.3.2 *Transition Probabilities.* Once all the output probabilities have been set, we can learn the transition probabilities of this HMM from observed sequences, that is, documents. Because passage length is document-specific, and transition probabilities are important factors for determining passage length, we decide that we should allow the transition probabilities for the HMM to also be document-specific. We do not have labeled training data to learn the transition probabilities because transition probabilities differ from document to document. We can, however, use unsupervised learning. For each document, we first fix the output probabilities at each state of the HMM, as we have described in Section 3.3.1, and then use the document itself as the only observed sequence to learn the transition probabilities, using the Baum-Welch algorithm.

3.4 Finding Relevant Passages

After all the state transition probabilities $p(S_1|S_2)$ ($S_1, S_2 \in \{B_1, B_2, R\}$) and output probabilities $p(w|\mathcal{R})$ and $p(w|\mathcal{B})$ are fully specified in the basic HMM, S^* can then be efficiently found using the Viterbi algorithm. The intuitive reason why the HMM can identify the relevant passage is that the latter usually contains more query words than the nonrelevant passages, and for a query word w_i, $p(w_i|\mathcal{R})$ is higher than $p(w_i|\mathcal{B})$. Thus w_i is more likely to be associated with R than with B_1 or B_2. Meanwhile, a reasonable $p(R|R)$ value ensures that the state sequence stays at state R for a reasonable length of time. With both the output and transition probabilities appropriately set, combining them can automatically adjust the position of the relevant passage.

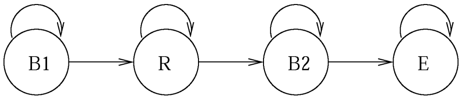

Fig. 2. Improved HMM structure.

Note that we can iteratively apply the same method to extract additional relevant passages from the rest of the document, or to design a more complex HMM structure to extract multiple passages at once. We leave these considerations to our future work and focus on extracting the single most relevant coherent passage from a relevant document in this article.

4. IMPROVED HMMs

The basic HMM method we showed earlier has some limitations. In this section, we discuss a number of improvements to the basic HMM method.

4.1 Structure Refinement

A problem with the three-state HMM is that the state sequence can stop at state R or even state B_1, without violating any restriction of the model. Indeed, during the unsupervised training of the transition probabilities, to maximize the likelihood, there is a tradeoff between having state R generate the query-related words, which increases the likelihood, and going from state B_1 to state R, which has a low transition probability and thus decreases the likelihood. If we know that there is a relevant passage in the document, we can impose this prior knowledge on the unsupervised training by "forcing" the state sequence to go through state R, which will improve the performance. To force a state sequence to contain state R, we extended the three-state HMM to a four-state HMM by adding a fourth state at the end to indicate the termination of a document. The end state E generates no words, but a special symbol ε that marks the end of a document. This symbol ε is also added to the end of each document. Hence, a document originally represented by $(w_1, w_2, \ldots, w_n)$ now becomes $(w_1, w_2, \ldots, w_n, \varepsilon)$. Figure 2 shows this improved HMM. In our preliminary experiments, we found that the four-state model indeed outperformed the three-state model; the F1 measure increased from 0.306 to 0.655 on the DOE dataset (described in Section 5.2).

Another problem with both the three-state and four-state HMM is that the smoothing parameter λ has to be empirically tuned. To automate the smoothing of the relevance language model, we added another background state to the HMM, as shown in Figure 3. Relevant passages can now be generated from both state R and state B_2. Note that in this five-state HMM, the smoothing of the relevant language model is achieved by the transitions between state R and state B_2. To see how, consider a state sequence that has entered state R. Now, to generate a nonquery word, besides switching to state B_3, the state sequence also has the option to switch to state B_2, from which it can switch back to state R later to generate another query word. Therefore, nonquery words in the relevant passage are more likely to be generated from state B_2 than from state B_3, thus state B_2 serves as the background language model in

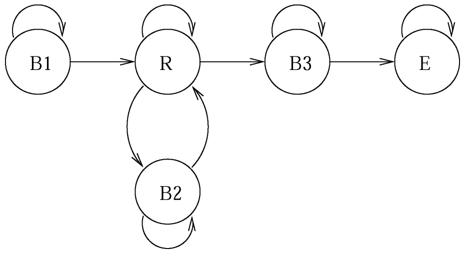

Fig. 3. Final HMM structure.

Table II. Some Statistics of the Transition Probabilities of the Five-State HMM Trained on the HARD04 Dataset

	Statistics of the Probability Sample		
Transition	Mean	Variance	Standard Deviation
$B_1 \to B_1$	0.884	0.081	0.284
$B_1 \to R$	0.129	0.035	0.188
$R \to R$	0.118	0.027	0.165
$R \to B_2$	0.613	0.206	0.454
$R \to B_3$	0.293	0.080	0.283
$B_2 \to R$	0.125	0.052	0.229
$B_2 \to B_2$	0.866	0.052	0.228
$B_3 \to B_3$	0.893	0.111	0.333
$B_3 \to E$	0.130	0.061	0.248
$E \to E$	1.000	0.000	0.000

Equation (4). Note that since the transition probabilities between R and B_2 are also trained, the smoothing parameter is not manually tuned, but learned from the observations. The five-state HMM is the final HMM structure we used in our experiments.

To give a concrete idea of how the five-state HMM looks after training, in Table II, we show some statistics for a sample of the transition probabilities for each pair of states that has a nonzero transition probability. We trained a five-state HMM on each document in the HARD04 dataset, which is one of the two datasets we use and is described in Section 5.2. We then collected the transition probabilities for all the documents, and calculated the sample mean, sample variance, and sample standard deviation of the transition probability for each pair of states. First, as shown in the table, we see that the sample means of the transition probabilities for $B_1 \to B_1$, $B_2 \to B_2$, and $B_3 \to B_3$ are the largest, all between 0.85 and 0.9. These relatively large numbers indicate that most of the transitions between two consecutive words in a document are within either a relevant passage or a nonrelevant passage. The sample mean of the transition probability for $R \to B_2$ is also relatively large, and again, this transition is within a relevant passage. Second, transitions between relevant and nonrelevant passages have lower probabilities, as indicated by the sample means of the transition probabilities for $B_1 \to R$ and $R \to B_3$. Indeed, there should be only one transition from B_1 to R and one from R to B_3 for each document. Third, the sample means of the transition probabilities for $R \to R$ and $B_2 \to R$ are quite small, indicating that only a small number of words in the relevant passages are generated by state R, while most are generated by state

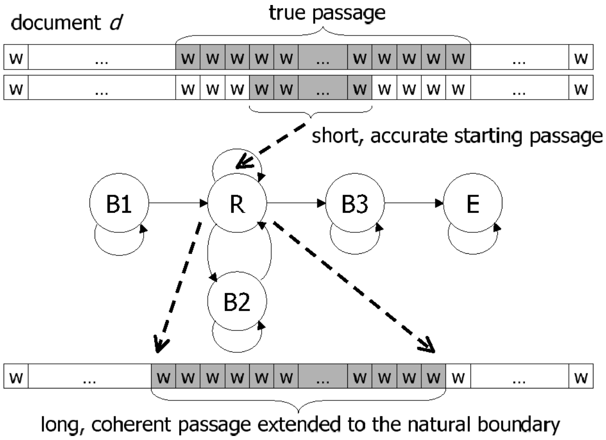
document *d*
true passage
w
...
w w w w w
...
w w w w w
...
w
short, accurate starting passage
B1
R
B3
E
B2
long, coherent passage extended to the natural boundary

Fig. 4. Within-document pseudo-feedback.

B_2, consequently showing that state B_2 serves as a smoothing state for relevant passages. Last, the sample variances show that the transition probabilities vary from document to document. Therefore, it is necessary to train the HMM on each document to obtain the transition probabilities.

4.2 Parameter Estimation Based on Feedback

In the final five-state HMM, the simplest way to estimate the output probabilities at state R is to use the query language model, as in Equation 3. The query language model does not need to be smoothed here because the transitions between R and B_2 essentially implement the smoothing mechanism.

A potentially better way to estimate the relevance language model is to incorporate pseudo-feedback. We present two pseudo-feedback mechanisms here.

4.2.1 *Within-Document Pseudo-Feedback.* The first pseudo-feedback mechanism considers each document individually, and uses a short, accurate passage extracted from the document as feedback to estimate the relevance language model for the same document. The motivation is as follows. Since a relevant passage within a document is usually a coherent piece of text, if we already have a short passage from the document that is guaranteed to be relevant to the query, we can imagine that the true relevant passage is probably a longer one containing this short starting passage and that it has a similar word distribution. If we use the starting short passage to estimate a relevance language model, and use this language model at state R, then state R should presumably attract similar text surrounding the starting passage. The short starting passage can thus be extended to the true relevant passage that is coherent in content and has a natural topical boundary with the rest of the document. Figure 4 illustrates this basic idea.

Let p be the short starting passage. Using maximum-likelihood estimation, we can estimate the relevance language model as follows:

$$p(w_i|\mathcal{R}) = \frac{c(w_i, p)}{\sum_{j=1}^{m} c(w_j, p)}. \tag{5}$$

Again, we do not need to smooth the relevance language model here because we have background state B_2.

4.2.2 *Cross-Document Pseudo-Feedback.* If short, accurate starting passages can help estimate the relevance language model within the same document, the next research question is whether different passages relevant to the same query can also help estimate the relevance language model across various documents. We present another pseudo-feedback mechanism here to explore this idea.

The idea of cross-document pseudo-feedback is similar to that of query expansion. An assumption here is that documents or passages relevant to the same query are also similar to each other. Thus, if we can expand the query language model based on feedback from different documents or passages that are ranked as highly relevant to this query, and use this expanded query language model as the relevance language model at state R, then presumably the output probabilities at R will be more accurate. In principle, we can apply any language model-based pseudo-feedback methods (e.g., relevance language models [Lavrenko and Croft 2001] and mixture-feedback language models [Zhai and Lafferty 2001a]) to expand the query. We now present a method that uses previously extracted passages.

Suppose for query q, we have extracted its relevant passages from a set of documents we are considering by using some basic passage extraction method. This can be our HMM-based method with direct parameter estimation from the query only, or any other method, such as window-based. We can then construct an expanded query language model based on all or some of these relevant passages. Suppose we are to use passages $p_1, p_2, \ldots, p_l$ for pseudo-feedback. If we assume all the words in l passages are samples drawn independently from a language model $\mathcal{Q}'$, then based on these observed passages and using maximum-likelihood estimation, $\mathcal{Q}'$ can be estimated as follows:

$$p(w_i|\mathcal{Q}') = \frac{\sum_{k=1}^{l} c(w_i, p_k)}{\sum_{k=1}^{l} \sum_{j=1}^{m} c(w_j, p_k)}, \tag{6}$$

where $c(w_i, p_k)$ is the number of times word w_i appears in passage p_k, and m is the total number of words in the vocabulary. This expanded query language model can now be used at state R as the relevance language model. In our experiments, we used all relevant passages removed by the basic passage extraction method from the set of documents that were known to be relevant to the query for cross-document pseudo-feedback.

As we have seen in Sections 4.2.1 and 4.2.2, our final HMM structure can naturally incorporate feedback. An important observation is that there is no restriction on the method used to extract the short, starting passages for feedback, as long as the starting passages are highly relevant to the query. The HMM shown in Figure 3 can be easily built on top of any basic passage extraction method. Thus our HMM method provides a framework for incorporating feedback for passage extraction. In Section 6, we show how the feedback mechanism can also be used on top of two baseline methods to significantly improve performance.

4.3 Summary

To summarize, to extract the most relevant passage from document d with respect to query q, we first use the whole collection to estimate the background language model $\mathcal{B}$. We then estimate the relevance language model $\mathcal{R}$, either directly from the query language model or from some feedback, particularly within-document or cross-document pseudo-feedback. After all the language models have been estimated, we train the transition probabilities of the HMM for each document using the Baum-Welch algorithm and also using the same document as the only observation sequence. Once the transition probabilities are estimated, we use the Viterbi algorithm to find the most likely state sequence for document d, and thus extract the most relevant passage.

5. EXPERIMENT DESIGN

Although passage retrieval methods have previously been evaluated in the context of either document retrieval or question answering, neither operation considers the coherence of passages or the topic shifts at passage boundaries. We evaluated our passage extraction methods by directly looking at the overlap between gold standard and extracted passages. This metric is similar to that used in TREC 2004 HARD track evaluation, except that we do not consider passage ranking.

To evaluate our HMM-based passage extraction method, we tried three different parameter estimation methods. *HMM-q* uses only the original queries to estimate the relevance language model, whereas *HMM-wd* and *HMM-cd* use within and cross-document pseudo-feedback, respectively, to estimate this same model.

5.1 Baseline Methods

We implemented a number of baseline methods for comparison, since there is as of yet no reported performance on this problem that we can find for direct comparison. HARD 2004 has a passage retrieval task, but since passage ranking and extraction are mixed in their evaluation, the results therein are not directly comparable to our problem.

The first method, *BL-s*, is a simple baseline method which returns a passage that starts from the first and ends with the last occurrence of any query word in the document. The second, *BL-win*, is a stronger baseline that is based on fixed-size windows. Given a window size k, *BL-win* examines all passages that are k-word long, and chooses the one that has the most occurrences of the query words to be the most relevant. The other two baseline methods, *BL-cos* and *BL-pivoted*, are also window-based. They differ from *BL-win* in that they employ *TF·IDF* similarity to score passages. *BL-cos* uses the cosine measure defined in Kaszkiel and Zobel [2001] to compute the similarity between a passage p and a query q:

$$\mathrm{sim}(p, q) = \frac{\sum_{t \in p \wedge q}(w_{p,t} \cdot w_{q,t})}{W_p \cdot W_q}, \tag{7}$$

with

$$\begin{aligned} w_{p,t} &= \log_e(f_{p,t}+1), && (8)\\ w_{q,t} &= \log_e(f_{q,t}+1)\cdot\log_e\left(\frac{N}{f_t}+1\right), && (9)\\ W_p &= \sqrt{\sum_{t\in p} w_{p,t}^2}, && (10)\\ W_q &= \sqrt{\sum_{t\in q} w_{q,t}^2}, && (11) \end{aligned}$$

where $f_{x,t}$ is the number of occurrences of term t in x, N is the total number of documents, and f_t is the number of distinct documents containing t. *BL-pivoted* uses a pivoted cosine measure, again defined in Kaszkiel and Zobel [2001], to compute the similarity:

$$\mathrm{sim}(p,q) = \sum_{t\in p\wedge q}\left(\frac{w_{p,t}\cdot w_{q,t}}{W_p}\right), \qquad (12)$$

with

$$\begin{aligned} w_{p,t} &= 1+\log_e(1+\log_e(f_{p,t})), && (13)\\ w_{q,t} &= 1+\log_e(1+\log_e(f_{q,t}))\cdot\log_e\left(\frac{N+1}{f_t}\right), && (14)\\ W_p &= (1-\mathrm{slope})+\mathrm{slope}\cdot\frac{p_{\mathrm{len}}}{\mathrm{avg\_}p_{\mathrm{len}}}, && (15) \end{aligned}$$

where p_{len} is the passage length in words and $\mathrm{avg\_}p_{\mathrm{len}}$ is the average passage length. In our experiments, we used slope $= 0.2$, which was the same as that used in Kaszkiel and Zobel [2001]. Similarly, we followed these authors to set $\mathrm{avg\_}p_{\mathrm{len}}$ to the average of the passage lengths we experimented with (50 to 400), which was 200. *BL-cos* and *BL-pivoted* both choose the passage with the highest similarity score among all passages of length k from the document.

Kaszkiel and Zobel [2001] also proposed variable-length passages, where for each document, the passages of a set of predefined lengths are examined, and the one with either the highest cosine or pivoted cosine similarity with the query is selected. We experimented with this variable-length method, but found its performance much worse than that of the fixed-window-based method, partly because the former often favors short passages. We therefore did not include it as a baseline method.

5.2 Test Collections

Our experiments were carried out on two datasets: a synthetic dataset created from TREC DOE (Department of Energy) abstracts, and a subset of TREC 2004 HARD track data. Using the HARD04 dataset was a natural choice, since it is a reasonably large real dataset with its passage boundaries manually annotated. The reason we also used a synthetic dataset is that this would allow us to control the coherence of relevant passages and to vary the passage lengths, allowing deep understanding of an extraction method's behavior. TREC 2003 QA track also has a passage retrieval task with passage judgment. But as we pointed out in Section 1, the passages we consider are different from those in QA. Therefore we did not consider using QA data for our experiments.

The synthetic dataset was created from the DOE abstracts in the TREC data collection, Disk 1. We concatenated relevant and nonrelevant abstracts into long documents so that relevant abstracts could be considered relevant passages. We chose the DOE abstracts because we believe that these short abstracts are compact and highly relevant to the queries. We chose 35 topics from Topic 1 to Topic 150. We picked abstracts relevant to these topics, and then randomly concatenated them into long documents, with the only constraint being that each long document contains a single relevant passage (containing one, two, or three DOE topic-relevant abstracts) with respect to one of the topics. This synthetic dataset thus has very clear passage boundaries, and passages of different lengths. There are 1029 documents in this dataset. The average document length within this synthetic set in terms of number of words is 535. On average, 56.3% of a synthetic document in this set is marked as a relevant passage.

For the HARD04 dataset, we extracted only those documents that contain a single passage relevant to some topic, according to the annotations. There are 1152 documents extracted, relevant to 25 topics. The average document length is 475. On average, 65.2% of a document is marked as a relevant passage.

For both datasets, we used the Porter stemmer to perform stemming. In order to test the robustness of our methods, we did not remove any stop-words.

5.3 Evaluation Procedure

For each document in the dataset and the query it is relevant to, we used each passage extraction method to extract the most relevant passage. We then computed the precision, recall, and F1 measures, defined as follows, for the passage. Let N_t be the length (in number of words) of the true passage, as either manually annotated in the HARD04 dataset or explicitly marked in the DOE dataset. Let N_e be the length of the extracted passage. Let o be the overlapping text segment between the true and extracted passage, and let N_o be the length of o. Then,

$$\mathrm{P} = \frac{N_o}{N_e}, \qquad \mathrm{R} = \frac{N_o}{N_t}, \qquad \mathrm{F1} = \frac{2 \times \mathrm{P} \times \mathrm{R}}{\mathrm{P} + \mathrm{R}}.$$

Since we do not consider passage ranking, each extracted passage is treated equally in our experiments. We average the precision, recall, and F1 measures over all documents as the final performance measures.

6. EXPERIMENT RESULTS

In this section, we discuss our experiment results.

6.1 HMM Versus Baseline Methods

We first compare the three HMM methods with the four baseline methods on both datasets in Table III. Stars in the table indicate the best performance

Table III. HMM Methods vs. Baseline Methods

Collection		Precision	Recall	F1
DOE	*BL-s*	0.869	0.591	0.632
	BL-win	0.779	0.777	0.730
	BL-cos	0.764	0.763	0.717
	BL-pivoted	0.749	0.745	0.701
	HMM-q	0.940	0.500	0.561
	HMM-wd	0.932	0.630	0.659
	HMM-cd	0.941*	0.858*	0.862*
HARD04	*BL-s*	0.670	0.909	0.666
	BL-win	0.668	0.759	0.621
	BL-cos	0.671	0.781	0.628
	BL-pivoted	0.672	0.783	0.629
	HMM-q	0.709*	0.726	0.585
	HMM-wd	0.686	0.877	0.656
	HMM-cd	0.671	0.969*	0.706*

Stars indicate the best performance among all methods on the same dataset. *HMM-cd* outperformed all other methods on both datasets (in terms of F1 measure).

figures among all methods on the same dataset. For *BL-win*, *BL-cos*, and *BL-pivoted*, the window size k is set to approximately the average relevant passage length for that data collection, which is 250 for DOE data, and 300 for HARD04. Without any knowledge about the length of the relevant passage in each particular document, the best the system can do is to choose one that can perform well on average. We believe the true average relevant passage length is a good approximation to this optimal value, and is thus the best the system can choose for window-based methods. Actually, this gives these window-based baseline methods an unrealistic advantage, as relevance judgments are used to tune the window size. Thus *BL-win*, *BL-cos*, and *BL-pivoted* can be regarded as very strong baselines. Note that we do not tune the HMMs with any relevance judgment information about the passage boundaries or lengths. For *HMM-wd* and *HMM-cd*, the starting passages used for pseudo-feedback are extracted by *HMM-q*.

We can see from Table III that *HMM-cd* performed the best among all methods if we used F1 as the performance measure. A Wilcoxon signed-rank test showed that using F1 as the performance measure, *HMM-cd* performed significantly better than the best baseline method (*BL-win* for DOE and *BL-s* for HARD04), at $p = 0.001$. This shows that with good parameter estimation from pseudo-feedback, the HMM-based method can outperform all baselines.

We also see that *HMM-q*, the HMM method with parameter estimation taken directly from the queries, did not perform as well as the baseline methods in terms of the F1 measure. On both datasets, *HMM-q* achieved high precision among all methods, but with very low recall. The reason is that in *HMM-q*, the relevance language model is estimated using only the original query. The relevant passage that *HMM-q* extracts is therefore very conservative: the density of query words in the extracted passage must be high. Although occurrences of query words often indicate relevance to the query, the true relevant passage

Table IV. Effect of Pseudo-Feedback on DOE Data—**F1**

		BL-win (with different *k*)							
	BL-s	50	100	150	200	250	300	350	400
BL	0.632	0.344	0.529	0.636	0.696	0.730	0.749	0.756	0.755
HMM-wd	0.705	0.712	0.749	0.771	0.772	0.775	0.774	0.766	0.754
Improv.	**+12%**	**+107%**	**+42%**	**+21%**	**+11%**	**+6%**	**+3%**	**+1%**	**−0%**
HMM-cd	0.874	0.845	0.872	0.876	0.860	0.842	0.826	0.809	0.791
Improv.	**+38%**	**+146%**	**+65%**	**+38%**	**+24%**	**+15%**	**+10%**	**+7%**	**+5%**

does not need to contain query words everywhere. Thus, although *HMM-q* can achieve a high precision, its recall is lower than other methods.

However, when the relevance language model is estimated using pseudo-feedback, as in *HMM-wd* and *HMM-cd*, we can see that recall increased substantially compared with *HMM-q*, while precision did not decrease much. As a result, the overall performance of *HMM-wd* and *HMM-cd*, measured by F1, was an improvement over *HMM-q*. This agrees with our hypothesis that if we use short, accurate passages as article pseudo-feedback to estimate the relevance language model, the HMM method can automatically extend passages to the natural topical boundaries, and thus improve recall.

The baseline methods performed differently on the two datasets. *BL-s* achieved high precision and low recall on DOE data, but average precision and high recall on HARD04 data. *BL-win* outperformed both *BL-cos* and *BL-pivoted* on DOE data, but performed worse than these two on HARD04 data. This difference in the performance of the baseline methods between the datasets suggests that the datasets have different characteristics. On the other hand, it also suggests that no baseline method can perform consistently well on different datasets, whereas we see that *HMM-cd* consistently performed better than baseline methods on both.

6.2 The Effect of Feedback

As we have discussed in Section 4.2.2, the HMM method provides a framework for incorporating feedback from any basic passage extraction method. In this section, we show the results of applying the HMM method using pseudo-feedback from some of the baseline methods. First, we applied *HMM-wd* and *HMM-cd* on top of *BL-s*. Since *BL-s* achieved high precision on DOE data, passages extracted by *BL-s* on the DOE dataset are presumably a good choice as starting passages. We also applied *HMM-wd* and *HMM-cd* on top of one of the window-based baseline methods. Since the three window-based baseline methods did not differ significantly we picked *BL-win*. We used passages of different lengths extracted by *BL-win* as pseudo-feedback. The passage length *k* allowed us to control the accuracy of the starting passages and to study the effectiveness of the HMM method when passages of different accuracy are used for feedback. Tables IV, V, and VI show the F1, precision, and recall values, respectively, of applying *HMM-wd* and *HMM-cd* on top of baseline methods on DOE data, and Tables VII, VIII, and IX show the same on HARD04 data.

From these tables, we can draw a number of conclusions: (1) The HMM pseudo-feedback method is highly effective. Indeed, from Tables III and VI, we

Table V. Effect of Pseudo-Feedback on DOE Data—**Precision**

		BL-win (with different *k*)							
	BL-s	50	100	150	200	250	300	350	400
BL	0.869	0.913	0.888	0.851	0.815	0.779	0.747	0.716	0.688
HMM-wd	0.906	0.898	0.864	0.828	0.785	0.750	0.722	0.693	0.668
Improv.	**+4%**	**−2%**	**−3%**	**−3%**	**−4%**	**−4%**	**−3%**	**−3%**	**−3%**
HMM-cd	0.913	0.934	0.902	0.875	0.835	0.799	0.767	0.738	0.708
Improv.	**+5%**	**+2%**	**+2%**	**+3%**	**+2%**	**+3%**	**+3%**	**+3%**	**+3%**

Table VI. Effect of Pseudo-Feedback on DOE Data—**Recall**

		BL-win (with different *k*)							
	BL-s	50	100	150	200	250	300	350	400
BL	0.591	0.234	0.425	0.573	0.687	0.777	0.847	0.899	0.937
HMM-wd	0.703	0.646	0.721	0.789	0.841	0.890	0.926	0.950	0.966
Improv.	**+19%**	**+176%**	**+70%**	**+38%**	**+22%**	**+15%**	**+9%**	**+6%**	**+3%**
HMM-cd	0.893	0.824	0.891	0.924	0.943	0.957	0.968	0.978	0.984
Improv.	**+51%**	**+252%**	**+110%**	**+61%**	**+37%**	**+23%**	**+14%**	**+9%**	**+5%**

Table VII. Effect of Pseudo-Feedback on HARD04 Data—**F1**

		BL-win (with different *k*)					
	BL-s	50	100	200	300	400	500
BL	0.666	0.280	0.411	0.547	0.621	0.658	0.679
HMM-wd	0.683	0.665	0.688	0.698	0.703	0.703	0.703
Improv.	**+3%**	**+138%**	**+67%**	**+28%**	**+13%**	**+7%**	**+4%**
HMM-cd	0.709	0.694	0.701	0.707	0.708	0.707	0.708
Improv.	**+6%**	**+148%**	**+71%**	**+29%**	**+14%**	**+7%**	**+4%**

Table VIII. Effect of Pseudo-Feedback on HARD04 Data—**Precision**

		BL-win (with different *k*)					
	BL-s	50	100	200	300	400	500
BL	0.670	0.709	0.696	0.674	0.668	0.662	0.657
HMM-wd	0.666	0.671	0.667	0.661	0.659	0.657	0.659
Improv.	**−1%**	**−5%**	**−4%**	**−2%**	**−1%**	**−1%**	**+0%**
HMM-cd	0.664	0.670	0.666	0.664	0.660	0.658	0.657
Improv.	**−1%**	**−6%**	**−4%**	**−1%**	**−1%**	**−1%**	**0%**

Table IX. Effect of Pseudo-Feedback on HARD04 Data—**Recall**

		BL-win (with different *k*)					
	BL-s	50	100	200	300	400	500
BL-	0.909	0.219	0.374	0.606	0.759	0.846	0.907
HMM-wd	0.946	0.859	0.910	0.947	0.967	0.970	0.966
Improv.	**+4%**	**+292%**	**+143%**	**+56%**	**+27%**	**+15%**	**+7%**
HMM-cd	0.984	0.949	0.961	0.981	0.986	0.987	0.991
Improv.	**+8%**	**+333%**	**+157%**	**+62%**	**+30%**	**+17%**	**+9%**

can see that when applied on top of a baseline method, both *HMM-wd* and *HMM-cd* perform better than the latter in all cases by our primary measure F1, and substantially better in most cases. (2) The relative improvement of the HMM feedback method over that of the baseline is greater when we start with short, accurate passages rather than long, inaccurate passages. For example, if we look at Tables IV and V, we can see that when $k = 50$, although the F1 measure of the starting passages extracted by *BL-win* is low, their precision is high. Using these accurate starting passages, *HMM-cd* achieved the largest relative improvement over *BL-win* in terms of F1. As k increases, the accuracy of the starting passages decreases, as does the F1 of *HMM-cd* and the relative improvement of *HMM-cd* over *BL-win*. The optimal starting passages to use should therefore be of medium size so that their F1 measure is not too low, and their precision remains sufficiently good for feedback. (3) The HMM feedback method can increase recall while keeping precision at a similar level. This increase of recall shows that the HMM-method can indeed extend short passages based on the language models constructed from short starting passages. The small change in precision shows that the extended portion of newly extracted passages is also relevant to the query. (4) *HMM-cd* performed better than *HMM-wd*, which means cross-document pseudo-feedback is more effective than within-document pseudo-feedback, at least for these two datasets. One possible explanation is that the relevant passages from different documents for the same query tend to be similar. Hence, cross-document feedback can mutually reinforce and amplify the feedback effect. In the case where passages relevant to the same query are dissimilar, they will generally remain quite different from the nonrelevant background surrounding the relevant passages. Thus cross-document feedback is also not likely to hurt performance.

6.3 Sensitivity of Performance to Passage Length

As we pointed out in Section 1, window-based passage extraction methods do not consider query-specific passage boundaries, and therefore cannot handle variable-length passages well. The HMM-based method, on the other hand, detects query-specific passage boundaries based on a relevance language model that is sensitive to both the query and the coherence of passages. We now look at how the HMM methods *HMM-wd* and *HMM-cd* as well as the window-based baseline method *BL-win* performed over passages of different lengths. For comparison, we also include *BL-s* because it is a baseline method that allows variable passage lengths. To see the sensitivity of performance to passage length, we grouped the documents in each dataset into a number of buckets, according to their true passage lengths. We then plotted the average performance measures over the passages in each bucket for *BL-s*, *BL-win*, *HMM-wd*, and *HMM-cd*. Figures 5, 6 and 7 show the precision, recall, and F1 measures on the DOE dataset, respectively. Figures 8, 9 and 10 show the same performance measures on the HARD04 dataset. For each dataset, k is set to the average passage length of that dataset in *BL-win*. *HMM-cd* and *HMM-wd* are based on the passages extracted by *BL-win* for pseudo-feedback.

We can see from Figures 5 and 8 that for documents with different true passage lengths, the precision of both *HMM-wd* and *HMM-cd* is similar to that of *BL-win*. For these three methods, the precision is low for documents with short true passages because *BL-win* uses the average passage length, which is longer than these short true passages. *BL-s* gives high precision for short passages because *BL-s* does not impose a fixed passage length. From

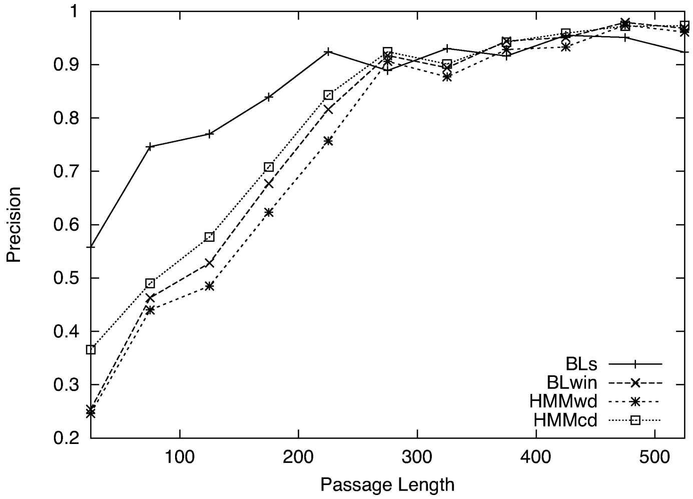

Fig. 5. Precision vs. passage length on DOE data.

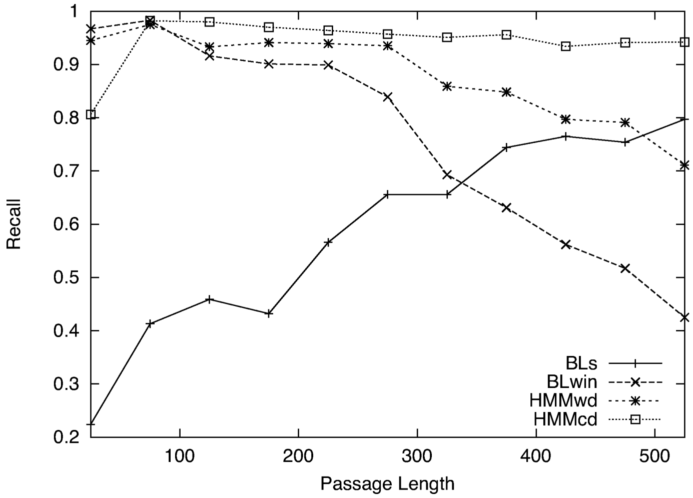

Fig. 6. Recall vs. passage length on DOE data.

Figures 6 and 9, we can see that *BL-s* gives low recall because it only returns the passage between the first and last query words, which is probably only a segment of the true passage. *BL-win* only achieved high recall for documents with true passage lengths approximately equal to or less than the average passage length. However, *HMM-wd* and *HMM-cd* achieved consistently high recall over different true passage lengths. As a result, we see from Figures 7 and 10 that both *HMM-wd* and *HMM-cd* achieved consistently good F1 values for different passage lengths.

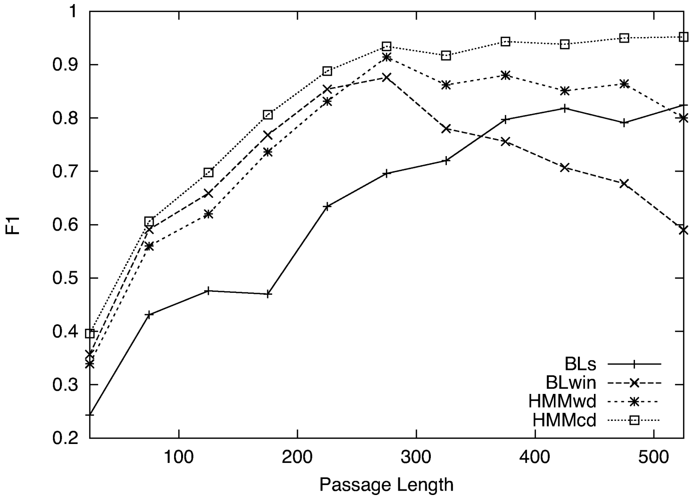

Fig. 7. F1 vs. passage length on DOE data.

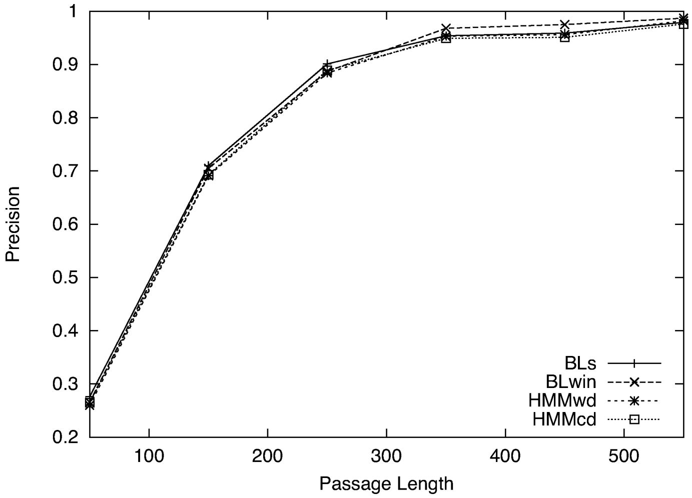

Fig. 8. Precision vs. passage length on HARD04 data.

7. RELATED WORK

Passage retrieval is an important component of question answering systems. Tellex et al. [2003] evaluated a number of passage retrieval methods in QA systems. However, passage retrieval in QA systems is very different from the passage extraction problem we address here—the former looks for passages that contain answers to very specific questions, hence often only a few sentences or even a single sentence in length, while the latter looks for coherent passages that contain complete pieces of information about more general topics, which are generally longer and have clearer topical boundaries with the rest of the documents.

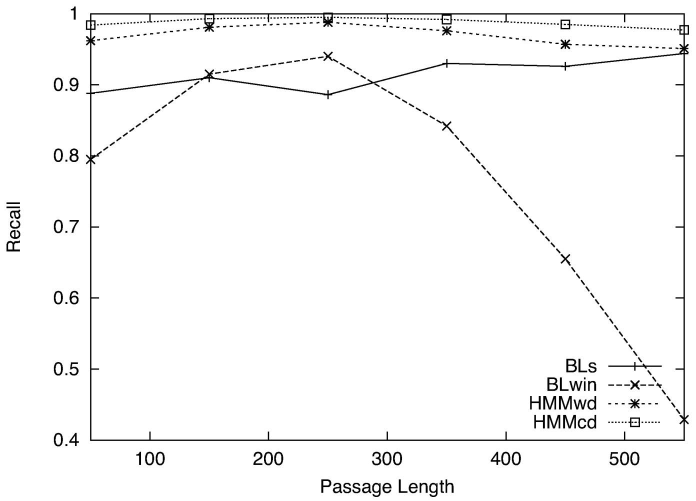

Fig. 9. Recall vs. passage length on HARD04 data.

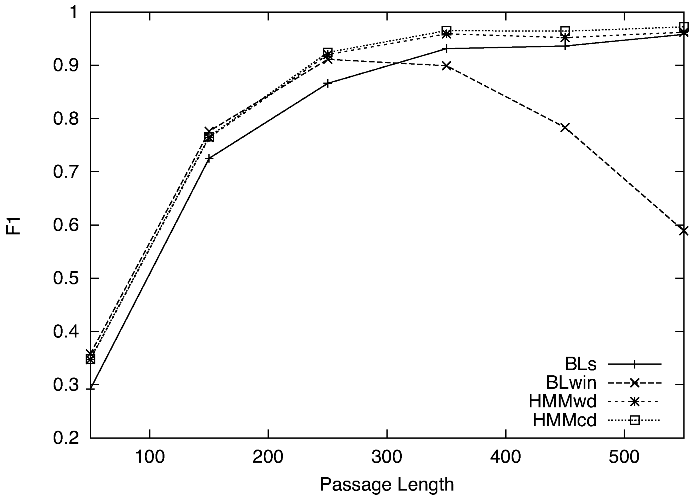

Fig. 10. F1 vs. passage length on HARD04 data.

Some previous passage retrieval methods have been based on structural markups such as paragraph and section boundaries. A problem with this approach is that these structural boundaries may be inconsistent among different authors. Callan [1994] observed that the structure of a document might be irrelevant to its content, but merely for presentation purposes. Another approach is to automatically identify semantic boundaries, for example, as in Hearst's [1997] TextTiling method. A problem with this approach is that these presegmented passages are not query-specific. A third approach that is commonly used is window-based, where the number of sentences or words in a passage is fixed. Although this approach has been shown to be effective for document reranking, it does not consider passage coherence and our experimental results also show that it does not perform well over varying passage lengths. Thus, a window-based approach is not good for the purpose of passage extraction. Researchers have also tried to identify arbitrary passages. In the MultiText system [Cormack et al. 1998], heuristic rules and scoring functions are used to rank arbitrary passages. The variable-length arbitrary passages proposed by Kaszkiel and Zobel [2001] are chosen from a set of fixedlength passages with a set of predefined different window sizes. However, none of these lines of work has considered coherence, whereas the HMM method proposed in this article can model coherence boundaries in a principled way.

The idea of applying HMMs for passage retrieval is not new [Mittendorf and Schäuble 1994; Knaus et al. 1996; Denoyer and Zaragoza 2001; He et al. 2004], but our method differs from previous approaches. Our method was first proposed in Jiang and Zhai [2004]. In this article, we refine the model, and propose a cross-document pseudo-feedback mechanism. Compared with previous HMM-based methods, ours has three major differences. First of all, some previous work uses larger building blocks, such as sentences [Knaus et al. 1996] or paragraphs [He et al. 2004], to build the hidden Markov models. In those cases, a mapping function is needed to transform a single text segment (a sentence or paragraph) into a value that encodes the similarity between the text segment and query. For example, in He et al. [2004], the authors employed a set of similarity measures to map a paragraph into a set of scalar values, and then linearly combined them into a single scalar value. They built a two-state HMM wherein one state generates relevant paragraphs and the other generates nonrelevant paragraphs. The output probabilities in each state follow a Gaussian distribution. In Mittendorf and Schäuble [1994], even though the building blocks are single words, the authors employed a similarity measure to map the words into scalar values, and used these as outputs from the hidden states in HMMs. In these methods, the issue of how to define such a mapping function becomes important, but their HMM-based model provides no guidance on this, since the mapping function is outside the model. By contrast, our method is based on unigram language models, so it does not require a similarity mapping function, but naturally captures the similarity between queries and passages. Second, we trained the transition probabilities for each individual document, whereas in some previous methods, the transition probabilities were fixed manually. Third, unlike using passages for document retrieval [Mittendorf and Schäuble 1994] or classification [Denoyer and Zaragoza 2001], we addressed the passage extraction problem, in which the evaluation focuses on the accuracy of passage boundaries.

There has been some work in automatic text summarization that employs hidden Markov models [Conroy and O'Leary 2001; Fung et al. 2003; Zajic et al. 2005]. Although our HMM-based passage extraction method bears some similarity to these summarization methods, there are at least three major differences. First, Conroy and O'Leary [2001], Fung et al. [2003], and Zajic et al. [2005] all address the extractive summarization problem, and wherein, the goal is to select a set of sentences (or a set of words, as in headline generation) that can summarize the original document(s). Thus, the selected sentences are usually not contiguous in the original document(s), and there is no consideration of their coherence. When the method decides whether a given sentence should be included in the summary, besides its content words, its length and position in the document are often also considered. On the other hand, in passage extraction, the goal is to select coherent, query-dependent relevant passages that consist of contiguous sentences from the original documents. The task focuses more on the coherence of the extracted passages. Second, Conroy and O'Leary [2001], Fung et al. [2003], and Zajic et al. [2005] all address the generic summarization problem, in which there is no notion of user queries. The goal is to capture the essential contents of the document(s). By contrast, in passage extraction, the extracted passage should be relevant to the user query. Third, most summarization work, such as Conroy and O'Leary [2001] and Fung et al. [2003], extracts sentences, so the building blocks of the hidden Markov models they use are thus sentences. Therefore, similar to other work in passage retrieval using HMMs, they need a mapping function to transform the sentences into some similarity measures, and again, how to define such a mapping function becomes an important issue. In our HMM-based passage extraction method, we use unigram language models to naturally model the similarity between the passage and the user query. In Zajic et al. [2005], although their building blocks for the HMMs are also words, because they are generating headlines, the words they select are not contiguous in the original documents, which is a complete departure from passage extraction.

HMMs have also been successfully applied to information extraction [Freitag and McCallum 2000], where the HMM structure contains target and background states, which is a little similar to our HMM structure. However, theirs is used to address a completely different problem.

8. CONCLUSION AND FUTURE WORK

Passage extraction is an essential component of passage retrieval, and can benefit an information retrieval system in many respects. Unlike previous work, which tends to mix passage extraction with a ranking component in evaluation, our work aims at directly studying the problem of passage extraction. We proposed an HMM-based method for passage extraction which can naturally exploit the coherence in the text to accurately identify coherent relevant passages of variable lengths. We studied the design of the HMM structure, and presented three different methods to estimate the parameters. With its refined structure and appropriate parameter estimation, our HMM method outperformed a number of baseline methods. We also showed that the HMM method naturally provides a framework for incorporating feedback from any basic passage extraction method, which can improve passage boundaries and hence, the overall performance. Moreover, the HMM method can perform consistently well over different passage lengths.

Currently, our HMM method only extracts a single relevant passage from a given document. It can be extended to handle multiple relevant passages per document, however, care must be taken in the design. If we simply allow a transition from B_3 to B_1, that is, to allow a loop in the HMM so that multiple relevant passages can be generated, then without any constraint on the transition probabilities, the unsupervised training would assign very high transition probabilities from R to B_3. This is because having R generate the query words and B_1 and B_3 generate most of the background words could result in a larger likelihood. The resultant extracted passages would consist of a short fragments surrounding the query words in a document. One possibility is to force the transition probabilities between B_1 and B_3 to be small. Currently, we have not found a principled good solution to this problem.We will further study the problem in the future.

We did not make use of the markups in the documents when we used HMMs to extract passages because our method is based on unigram language models. Recently, many documents in XML format have become available, providing both good datasets on which to explore the usage of markup information for passage extraction, and ground-truth passage boundaries on which to evaluate our method. We will consider using these datasets in the future for the purpose of improving our method.

Additional interesting problems for further study are the issues of how to better estimate the relevance language model and how to exploit extracted passages to improve document ranking.

REFERENCES

Allan, J. 2003. Hard track overview in trec 2003: High accuracy retrieval from documents. In *Proceedings of the 12th Text REtrieval Conference*. 24–37.

Callan, J. P. 1994. Passage-level evidence in document retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Dublin. Springer Verlag, New York, NY, 302–310.

Clarke, C. L. A. and Terra, E. L. 2003. Passage retrieval vs. document retrieval for factoid question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Toronto, Canada. ACM Press, New York, NY, 427–428.

Conroy, J. and O'Leary, D. P. 2001. Text summarization via hidden Markov models and pivoted QR matrix decomposition. Tech. Rep., University of Maryland, College Park.

Cormack, G. V., Clarke, C. L. A., Palmer, C. R., and To, S. S. L. 1998. Passage-based refinement (MultiText experiments for TREC-6). In *Proceedings of the 6th Text REtrieval Conference*. 303–320.

Corrada-Emmanuel, A. and Croft, W. B. 2004. Answer models for question answering passage retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Sheffield, UK. ACM Press, New York, NY, 516–517.

Denoyer, L. and Zaragoza, H. 2001. HMM-based passage models for document classification and ranking. In *Proceedings of the 23rd BCS European Annual Colloquium on Information Retrieval*.

Freitag, D. and McCallum, A. 2000. Information extraction with HMM structure learned by stochastic optimization. In *Proceedings of the 18th Conference on Artifitical Intelligence (AAAI)*. 584–589.

Fung, P., Ngai, G., and Cheung, P. 2003. Combining optimal clustering and hidden Markov models for extractive summarization. In *Proceedings of the ACL Workshop on Multilingual Summarization*.

He, D., Demner-Fushman, D., Oard, D. W., Karakos, D., and Khudanpur, S. 2004. Improving passage retrieval using interactive elicitation and statistical modeling. In *Proceedings of the 13th Text REtrieval Conference*.

Hearst, M. A. 1997. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist. 23*, 1, 33–64.

Jiang, J. and Zhai, C. 2004. UIUC in HARD 2004–Passage retrieval using HMMs. In *Proceedings of the 13th Text REtrieval Conference*.

KASZKIEL, M. AND ZOBEL, J. 1997. Passage retrieval revisited. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Philadelphia, PA. ACM Press, New York, NY, 178–185.

KASZKIEL, M. AND ZOBEL, J. 2001. Effective ranking with arbitrary passages. *J. American Society Inf. Sci. 52*, 4, 344–364.

KNAUS, D., MITTENDORF, E., SCHÄUBLE, P., AND SHERIDAN, P. 1996. Highlighting relevant passages for users of the interactive SPIDER retrieval system. In *Proceedings of the 4th Text REtrieval Conference*.

LAVRENKO, V. AND CROFT, W. B. 2001. Relevance-based language models. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New Orleans, LA. ACM Press, New York, NY, 120–127.

LIU, X. AND CROFT, W. B. 2002. Passage retrieval based on language models. In *Proceedings of the 11th International Conference on Information and Knowledge Management*. McLean, VA. ACM Press, New York, NY, 375–382.

MITTENDORF, E. AND SCHÄUBLE, P. 1994. Document and passage retrieval based on hidden Markov models. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Dublin. Springer Verlag, New York, 318–327.

RABINER, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE 77*, 2, 257–286.

SALTON, G., ALLAN, J., AND BUCKLEY, C. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Pittsburgh, PA. ACM Press, New York, NY, 49–58.

TELLEX, S., KATZ, B., LIN, J., FERNANDES, A., AND MARTON, G. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Toronto, Canada. ACM Press, New York, NY, 41–47.

ZAJIC, D., DORR, B., AND SCHWARTZ, R. 2005. Headline generation for written and broadcast news. Tech. Rep. LAMP-TR-120,CS-TR-4698,UMIACS-TR-2005-07, University of Maryland, College Park.

ZHAI, C. AND LAFFERTY, J. 2001a. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management*. Atlanta, GA. ACM Press, New York, NY, 403–410.

ZHAI, C. AND LAFFERTY, J. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New Orleans, LA, 334–342.

Received October 2005; revised March 2006; accepted June 2006