Passage Retrieval and Evaluation

Courtney Wade and James Allan
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts
Amherst, MA 01003
{cwade, allan}@cs.umass.edu

ABSTRACT

Information retrieval researchers have studied passage retrieval extensively, yet there is no consensus within the community about how to evaluate the results of passage retrieval experiments. This paper describes five character-level passage evaluation measures and tasks for which they may be appropriate. In the second half of the paper we compare several passage retrieval models, including a new generative mixture model that outperforms strong baselines on many of the evaluation measures discussed in part one.

1. INTRODUCTION

Passage retrieval, as described here, is the task of retrieving only the portions of a document that are relevant to a particular information need. It could be useful for limiting the amount of non-relevant material presented to a searcher, or for helping the searcher locate the relevant portions of documents more quickly. Passage retrieval is also often an intermediate step in other information retrieval tasks, like question answering and summarization.

Passage retrieval has been of interest to researchers since the 1970's [25]. However, it was not until the field of information retrieval shifted from abstract retrieval to full-text retrieval in the late 1980's and early 1990's that researchers began to study this problem more extensively [2, 12, 27, 7, 22, 23, 31].

Typically evaluation has been based on the ability of passage retrieval systems to retrieve documents [7, 22, 31, 23, 16, 8, 20]. In some cases, researchers are actually interested in improving document retrieval by using passage retrieval, in which case this type of evaluation is appropriate. However, one of the reasons researchers have used this approach to evaluation in the past is that document-level relevance judgments are much easier to obtain than passage-level judgments. Passage-level judgments require annotators either to read each top-ranked passage and judge it relevant or non-relevant, or to read each top-ranked document and mark only the portions of the text that are relevant to their information needs, a very time-consuming process. Passage retrieval evaluation has been studied more extensively as an intermediate step in question answering (QA) systems [29]. However, passage retrieval evaluation in QA has focused on whether or not retrieved passages contain correct answers, making it inapplicable to general passage retrieval.

The TREC High Accuracy Retrieval from Documents (HARD) track [3, 4], begun in 2003 and continued in 2004, included a passage retrieval component. Instead of indicating just a topic ID and document ID to identify the relevant material, the passage-level relevance judgments provided in the HARD track also indicated a byte offset from the beginning of the document and length in bytes of each relevant passage. However, initial attempts to adopt document evaluation metrics for passage retrieval resulted in a metric with unanticipated undesirable properties. In section 2, we describe some of the problems with this measure and present five character-level passage retrieval evaluation metrics, suitable to different types of passage retrieval applications.

In section 3 we describe seven different passage retrieval models, and in section 4 we compare the retrieval performance of these models using some of the evaluation metrics described in section 2. Among the models described is a new generative mixture model that significantly outperforms TFIDF, query-likelihood, and other retrieval baselines.

2. PASSAGE RETRIEVAL EVALUATION

In all of our experiments we use the data and data format used in the TREC HARD track, where a passage can be any continuous nonempty substring of text from one document. This means that in theory passages can be anywhere from one character to the entire document. The results specify a ranked list of passages for each topic, identified by their document ID, the byte offset of the passage start relative to the beginning of the document, and the length of the passage in bytes. Annotators received a pooled list of top-ranked documents for each of their topics and were asked to read through each of the documents and indicate which portions of each document were relevant, akin to going through with a highlighter and highlighting only the relevant material. The final relevance judgments are indicated in a format similar to that of the ranked results lists.

2.1 Passage R-Precision

Passage R-precision was used as an evaluation metric for the 2003 and 2004 HARD tracks. It is defined as the percent of relevant characters in the first R passages returned that were marked relevant by an annotator, where R is the total

CIIR Technical Report. February 2005.

number of passages marked relevant for the topic. In the HARD track, any character position from a document that is retrieved multiple times is counted as relevant once and non-relevant all of the other times. Another way of putting this is that passage R-precision is equal to the count of relevant characters returned at least once in the top k, divided by k, where k is the number of characters retrieved in the first R passages:

$$\text{psg R-prec} = \frac{|\text{relevant chars. ret.} \geq \text{once in top } k|}{k}. \quad (1)$$

One troubling aspect of this evaluation metric is that the value of R is somewhat arbitrary. In the 2004 HARD relevance judgments there are several cases where an annotator marked all of the text in adjacent passages relevant, but omitted the open and close paragraph tags in the text markup. Because of the way the relevance judgments are specified, this counts as two different passages even though the text is contiguous in the original article.

Let k be the number of characters retrieved in the top R passages returned for a particular topic and j be the total number of characters in the R relevant passages (from the relevance judgments) for that topic. One consequence of this measure is that if $k > j$, it is impossible to achieve a perfect R-precision score of 1.0. On the other hand, if $k < j$, it is possible to achieve a perfect R-precision score of 1.0 without retrieving all of the relevant characters. This indicates that there could be a bias toward systems that return shorter passages, particularly since we are dealing with a ranked list that (we hope) tends to have more relevant characters near the top.

To demonstrate some of the problems with passage R-precision, we took one of our submitted runs from the 2003 HARD track and split each of the top-ranked passages into two parts so that the first half of the first ranked passage is rank one and the second half is rank two, and so on. We repeated this process four more times so that the original passages were one thirty-second of their original size.

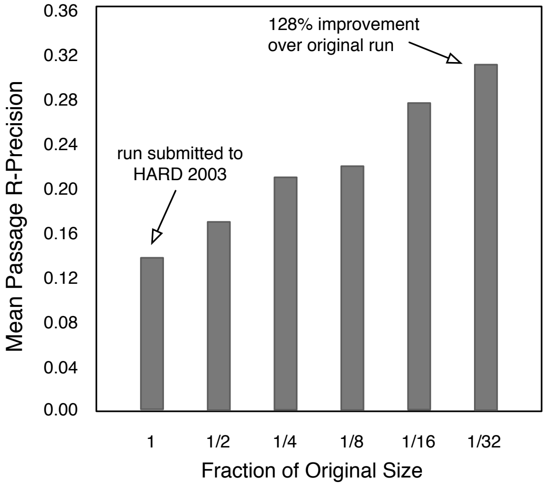

Figure 1: Sensitivity of Passage R-Precision to Passage Size

Figure 1 shows that as the passages get shorter—but the text stays equivalently ranked—mean passage R-precision improves steadily. We are able to improve our results by 128%, simply by cutting passages into smaller pieces. This is clearly an undesirable property of an evaluation metric. One way to fix this would be to require all HARD track participants to retrieve passages of the same size–a particular number of characters or words. However, this would be in opposition to the goal of designing systems that retrieve only the relevant text of a document. Another way to deal with this problem is to use character or word-level metrics. This was the solution adopted in part by the HARD 2004 track.

We used several character-level metrics in our evaluations, described in the following sections. The first two are focused on rewarding high precision at the top of a ranked list, and the other three balance precision and recall.

2.2 Precision-Focused Metrics

Precision at min(N, R) characters is the percent of characters in the first min(N, R) presented in the ranked list that are relevant. Here R is the total number of characters marked relevant in the relevance judgments for the topic being evaluated, and N is a non-negative integer, generally intended to be less than R. Note that when we talk about ranking characters, we act as though the first character of the first-ranked passage is at rank one, the second character is at rank two, and so on. In the HARD track evaluation in 2004, precision at min(N, R) characters was one of the measures used, with N set to 6,000, 12,000, and 24,000 (roughly the number of characters found in 5, 10, or 20 100-word stopped passages for the HARD corpus).

Bpref at min(N, R) characters is the high-precision character-level version of a new evaluation measure, binary preference (bpref), that tracks average precision but is less sensitive to incomplete relevance judgments [6]. To calculate binary preference at $k = \min(\text{N, R})$, each of the first k relevant characters in the ranked list is assigned a score equal to the percent of the first k non-relevant characters it comes ahead of. The overall score for the list is equal to the arithmetic mean of the scores for each of the k relevant characters. The equation for bpref at k = min(N, R) characters is

$$\text{bpref at } k \quad = \quad \frac{1}{k}\sum_{r=1}^{k}\left(1 - \frac{|n \text{ ranked higher than } r|}{\text{k}}\right)(2)$$

where r is a relevant character and n is one of the first min(N, R) non-relevant characters returned. Like with precision at min(N, R), R is the total number of characters marked relevant for the topic being evaluated and N is a non-negative integer.

Bpref at min(N, R) and precision at min(N, R) tend to track each other. One major difference between the two is that bpref rewards systems that rank passages well within the top min(N, R) characters. Because precision is a set-based measure, rank order within the top results is ignored.

2.3 Recall/Precision Balance Metrics

Precision at R characters (also called character R-precision) and *Bpref at R characters* are equal to their high-precision counterparts presented above when N = R. *Non-interpolated character average precision* is the arithmetic mean of the character precision at each point in the ranked list where recall changes (*i.e.* where a relevant character appears). It is the same as document average precision, except that recall and precision are computed at the character level rather than the document level. If we let $G =$ the set of ranks at which each relevant character is retrieved, then character average precision is

$$\text{char avg prec} = \frac{1}{|G|} \sum_{r \in G} \frac{\#\text{ relevant chars w/ rank } \geq r}{r}. \quad (3)$$

In our experiments, we assume that all non-retrieved relevant characters are retrieved at rank infinity.

2.4 Discussion of Metrics

We feel that there are several properties that are desirable in a passage retrieval evaluation metric. First, in measures that involve recall, the value of R should somehow reflect the true amount of relevant material. This was not the case with passage R-precision, but this problem is corrected with the introduction of character R-precision and bpref at R characters. Second, we prefer metrics that reward systems for getting more relevant material closer to the top of the ranked list. For this reason, we prefer rank-based measures like bpref at min(N, R), bpref at R, and character average precision to set-based measures such as precision at min(N, R) and character R-precision. However, we include these measures in our evaluation because they are easy to understand, and have counterparts in document evaluation. Finally, we feel that the appropriate metrics to evaluate passage retrieval depend very much on the specific application of the retrieval system. For example, there are applications where the goal is just to find as much relevant text as possible, but there are others for which we might want to find the most relevant passages of a particular length from each document.

The high-precision measures are most useful in applications where a relatively small amount of text will be presented to the user. For example, one can imagine a web search engine where a user types in a query, and gets back a page of relevant text, excerpted from one or more documents. Precision at min(N, R) and bpref at min(N, R), with N equal to roughly the number of characters that can be displayed on this page, would provide a measurement of how relevant this page of text is. In cases where the user was expected to read the entire page of text, the precision measure would suffice. In cases where the user was expected to start reading from the beginning of the page and stop reading once he found enough information, bpref would be more appropriate.

Another high-precision application for passage retrieval is a system where users issue a query and get back a ranked list of document titles, with an excerpt of relevant pieces of text under each title, like in most web search engines. Here we would like to find the most relevant parts of each document, but there is a cap on the maximum number of characters that we can retrieve from any one document. Precision at min(N, R) and bpref at min(N, R) would again be appropriate evaluation metrics, after modifying the retrieval criteria to put a limit on how many characters can be returned from a single document.

The measures that balance precision and recall are more appropriate for applications where the goal is to find all of the relevant text for a topic. One example of this is a document retrieval system where documents are presented with all of the relevant text highlighted so that users can find it faster but still see the context. Which of the three precision/recall measures is best for evaluating this type of system would depend on how important it is to find relevant text early on in higher ranked documents and the expected ratio of relevant to non-relevant text.

All of the measures presented here leave unaddressed the issue of human readability. Because these are all character-level measures, a system that retrieved a lot of relevant material at the top of a ranked list, but presented the characters in jumbled order could still score very high. This is one reason it is important to do sanity checks for any of these measures. In some cases, it might make sense to restrict the retrieval unit boundaries to sentences or phrases.

3. PASSAGE RETRIEVAL MODELS

We compare seven different models for retrieving passages: tfidf, query-likelihood language modeling, a relevance modeling approach for scoring passages or documents, a model based on support vector machines, a new method based on a simple mixture of language models, and a method for retrieving and scoring variable-length passages.

3.1 TFIDF

In the term frequency/inverse document frequency (TFIDF) retrieval model, each passage and query is modeled as a vector in the space of all text in the collection vocabulary. The score of a passage, given a particular query, is equal to the inner product of the vector representing the passage and the vector representing the query. The weight of each dimension in both the passage and query vectors is equal to the term frequency (TF) score of the term that corresponds to that dimension, times the inverse document frequency (IDF) score of that term. The TF score of a term t for a passage p is defined as

$$\text{TF}(t|p) \quad = \quad \frac{\text{count of } t \text{ in } p}{\left(\frac{\text{psg. len. in words}}{2 \times \text{avg. psg. len.}}\right) + 0.5 + \text{count of } t \text{ in } p}. \quad (4)$$

The TF score of a term t for a query q is defined as

$$\text{TF}(t|q) \quad = \quad \frac{\text{count of } t \text{ in } q}{\text{count of } t \text{ in } q + 1}. \quad (5)$$

The IDF score of a term, used for passages and queries, is

$$\text{IDF}(t) \quad = \quad \log\left(\frac{\#\text{ docs. in corpus} + 1}{0.5 + \#\text{ docs. } t \text{ appears in}}\right). \quad (6)$$

This rather *ad hoc* scoring formula is implemented in the Lemur toolkit [5], and arose out of years of experimentation and adjustment of TFIDF models.

3.2 Query Likelihood Language Model (QL)

The use of statistical language modeling in information retrieval was first proposed by Ponte and Croft in 1998 [26] and has been very influential on subsequent information retrieval research. In this model, each document is represented as a probability distribution over the vocabulary, or a language model. They then rank documents by the probability that the query Q was generated from each document model Θ_D.

Ponte and Croft used a multiple-Bernoulli model to estimate $P(Q|\Theta_D)$ which has been largely supplanted by the multinomial model described in several papers that followed shortly after [13, 21, 28]. In this model, words are treated as independent and identically distributed (i.i.d.) samples from the document model. This means they can be ranked according to equation 7.

$$P(Q = q_1, \ldots, q_k) \quad = \quad \prod_{i=1}^{n} P(q_i|\Theta_D) \tag{7}$$

Because maximum-likelihood estimates of the document model will lead to zero probability estimates for most terms, some form of smoothing [32] is generally used to estimate the document model Θ_D. In this case we use Jelinek-Mercer smoothing [14] which is simply the linear interpolation of the maximum likelihood model of the document $\Theta_{D_{ML}}$ and the maximum likelihood model of the entire document corpus $\Theta_{C_{ML}}$,

$$P(w|\Theta_D) = \lambda P(w|\Theta_{C_{ML}}) + (1-\lambda)P(w|\Theta_{D_{ML}}) \tag{8}$$

where $0 \leq \lambda \leq 1$. In order to score passages rather than documents using this model, we simply treat each passage as though it is a short document.

3.3 Relevance Modeling Approaches (RMP and RMD)

Lavrenko and Croft's relevance models [19] provide a language-modeling based approach for estimating a probability for each word in the relevant class of documents, $P(w|\Theta_R)$. They assume that both the query and the associated relevant documents R are samples from the distribution $P(w|\Theta_R)$, though it is not necessarily the case that a particular word w has the same probability in both the query and a relevant document.

Taking advantage of the assumption that the query $Q = q_1, \ldots, q_k$ is a sample from the model of relevance, they estimate the relevance model according to equation 9.

$$P(w|\Theta_R) \approx P(w|q_1, \ldots, q_k) = \frac{P(w, q_1, \ldots, q_k)}{P(q_1, \ldots, q_k)} \tag{9}$$

Lavrenko and Croft [19] describe two methods for estimating $P(w, q_1, \ldots, q_k)$. We describe only the first because this is the one used in all of our models.

In method 1, the assumption is that w and $q_1, \ldots, q_k$ are i.i.d. samples from a single unigram model Θ_D in some finite space of possible unigram models $\mathcal{R}$. This means they can write the joint distribution as

$$P(w, q_1, \ldots, q_k) = \sum_{\Theta_D \in \mathcal{R}} \left(P(\Theta_D) P(w|\Theta_D) \prod_{i=1}^{k} P(q_i|\Theta_D) \right) \tag{10}$$

They restrict $\mathcal{R}$ to contain only the language models of the top-ranked documents retrieved using the query likelihood model described in section 3.2 to do the initial retrieval. In other words, the assumption is that w and $q_1, \ldots, q_k$ are i.i.d. samples from a model of one of the top-ranked documents.

For computational efficiency, we truncate the relevance model Θ_R to include only the t most probable terms in the model where $P(t) \geq p$. In our experiments, $t = 100$, $p = 0.001$, and $|\mathcal{R}| = 20$. To ensure a valid probability distribution, we normalize the truncated relevance model to sum to 1. To avoid zero probability terms in the relevance model, we smooth it against the document corpus using Jelinek-Mercer smoothing.

Finally, because the relevance model may not place high enough weights on the original query terms, and indeed some query terms may not even appear in the model before the last smoothing step, we take a mixture of the relevance model and the maximum likelihood model of the original query Θ_Q using linear interpolation, to arrive at the final model

$$P(w|\Theta_{R^*}) = \lambda P(w|\Theta_Q) + (1-\lambda)P(w|\Theta_R), \ (0 \leq \lambda \leq 1). \tag{11}$$

We rank passages by the negative KL-divergence between this model $P(w|\Theta_{R^*})$ and the smoothed passage model Θ_P which is equivalent to ranking by query-likelihood, treating Θ_{R^*} as our a representation of our query [18].

$$-D(\Theta_{R^*}||\Theta_P) = -\sum_{w \in \Theta_{R^*}} P(w|\Theta_{R^*}) \log \frac{P(w|\Theta_{R^*})}{P(w|\Theta_P)} \tag{12}$$

We also include relevance model document ranking, referred to as RMD, as a point of comparison. The model is the same, except that we use a smoothed document model in place of the passage model.

3.4 Bootstrap Support Vector Machine (SVM)

A support vector machine (SVM) is a discriminative supervised learning algorithm described by Vapnik [30]. SVMs have been very popular in machine learning for a variety of classification tasks. In the SVM model, labeled training examples are represented by feature vectors and the goal is to find a boundary between the positively and negatively labeled training examples that maximizes the distance to the closest training examples, also known as the margin. The SVM algorithm does this by mapping the training examples, which are represented as points in the original feature space, into a higher dimensional space known as the kernel space where we hope the data are linearly separable. Mapping the data points into a sufficiently high dimensional space guarantees separability as long as the data set is consistent.

The discriminant function that distinguishes positive from negative examples, or relevant documents from non-relevant documents in our case, is

$$g(R|D, Q) = \langle \mathbf{w}, \phi(D, Q) \rangle + b. \tag{13}$$

Here ϕ is the function that maps a feature vector in the input space to a point in the kernel space, $\mathbf{w}$ is a weight vector learned from the training examples, and b is a constant. The SVM is trained so that $g(R|D, Q) \geq 1$ if D is relevant and $g(R|D, Q) \leq -1$ if D is not relevant.

Unfortunately the actual mapping from the feature space to the kernel space can be computationally expensive. Although we won't get into the details here, this can be converted into a dual optimization form that allows us to use a kernel function in place of actually computing a dot product in the feature space. This means that feature functions do not have to be represented explicitly in the higher dimensional space.

One of the issues in applying SVMs to information retrieval problems is that the class of non-relevant training examples tends to overwhelm the class of relevant examples. As a result, we train only on a small set of negative examples. Sometimes these examples are randomly selected as in [24], but AbdulJaleel [1] uses a "bootstrap" method for selecting negative training examples. This method involves a two-step process where the initial classification uses a random sample of negative training instances. In the second classification, the negative examples are those that were false positives in the initial classification. She uses the same features described in [24] to represent the data.

The bootstrap SVM method uses a relevance model to re-weight the query terms and is trained on relevant and non-relevant *documents* from TREC 1 and 2 [1]. It uses a linear kernel to rank passages by decreasing value of the discriminant function (13). The results described here for the bootstrap SVM are identical to those described in [1].

3.5 Mixture of Language Models (MM)

It has long been recognized that some combination of information from the text of the passage being modeled and the text of the document it came from may help to improve passage retrieval results [7]. Statistical language modeling has been used for both standard passage retrieval [20, 15] and in passage retrieval systems for question answering [9, 33]. However, we are unaware of any models like ours, which uses a mixture of document and passage language models to do passage retrieval.

In this model, we assume that each word in a passage is generated from a mixture of three multinomial language models: the corpus model Θ_C, the model of the document it came from Θ_D, and the model of the passage Θ_P. All three language models of these are calculated using maximum likelihood estimation.

$$P(w|\Theta_{MM}) = \lambda_1 P(w|\Theta_C) + \lambda_2 P(w|\Theta_D) + \lambda_3 P(w|\Theta_P) \quad (14)$$
$$(\lambda_1 + \lambda_2 + \lambda_3 = 1)$$

In all of our experiments we let $\lambda_1 = 0.8$, $\lambda_2 = 0.1$, and $\lambda_3 = 0.1$. A passage's final score is equal to the negative Kullback-Leibler divergence between the relevance model for the query Θ_{R^*}, given in equation 11, and the mixture model of the passage Θ_{MM}.

$$-D(\Theta_{R^*} \parallel \Theta_{MM}) = -\sum_{w \in \Theta_{R^*}} P(w|\Theta_{R^*}) \log \frac{P(w|\Theta_{R^*})}{P(w|\Theta_{MM})}. \quad (15)$$

Note that when $\lambda_2 = 0.0$, this model is the same as passage relevance modeling (RMP) and when $\lambda_3 = 0.0$ it is the same as document relevance modeling (RMD).

3.6 Query Word Density Model (QWD)

Of the retrieval models presented here, this is the only one that does variable-length passage retrieval. Previous approaches to variable-length passage retrieval include several HMM-based models [22, 17, 11, 15] and models based on query term density [8]. Our term-density QWD model is motivated by the approach used in [10] for scoring each word in a document. In this model, we start with the relevance model in equation 11, which we treat as an expanded query model. We linearly scale the weights of the query terms up to integer values. This leaves each query word t_i with a scaled weight w_i. For each document, we mark every query term t_i and w_i words to either side of it. Once we have done this for every query term, we extract every group of marked words that is longer than a particular threshold as a passage. In our experiments, we extracted only passages that were greater than or equal to 400 characters. Once we've extracted these passages, we rank them using our passage retrieval mixture model in equation 15.

4. EXPERIMENTAL RESULTS

We used a subset of the topics, relevance judgments, and corpus from the 2003 HARD track as our training data, and the topics, relevance judgments, and full corpus from the 2004 HARD track as our test data. The HARD 2003 document corpus consists of 320,380 news stories from the Associated Press, New York Times, and Xinhua News Agency from 1999. It also contains many Congressional Record and Federal Register documents from 1999, which were omitted from the training set because they were less similar to the test corpus. The HARD 2004 document corpus includes 652,309 news and lifestyle articles published in 2003 in eight different sources.

In the training set, we have 23 topics, for which a total of 1,042 documents were judged to have at least one passage relevant to one of these topics. The number of documents containing at least one relevant passage varies a lot by topic. One topic has only 2 documents judged to have a relevant passage, while another has 135. The mean number of documents with a relevant passage for each topic is 45 and the median is 32. In the test set, we have 25 topics, with a total of 1,682 documents judged to have at least one relevant passage. The minimum number of documents with a relevant passage for a topic is 1 and the maximum is 289. The mean and median are 67 and 34, respectively.

We present results for the seven models described above (referred to as TFIDF, QL, RMP, RMD, SVM, MM, and QWD, respectively) for bpref at min(12,000, R) characters, precision at min(12,000, R) characters, character R-precision, bpref at R characters, and non-interpolated character average precision. With the exception of the QWD variable-length passage retrieval model and the RMD whole document model, all retrieval algorithms were applied to an index of stopped 100 word, half-overlapping passages from the HARD 2004 corpus. The results presented are from the 25 test queries for which we have passage-level relevance judgments.

Table 1 shows that the mixture model described in section 3.5 (MM) performs significantly better than TFIDF, QL, SVM, and RMP when evaluated using binary preference at 12,000 characters. Although there was not room for all of the significance tests in this table, MM does *not* significantly outperform the QWD or RMD models. On average, the mixture model does 38.8% better than TFIDF, 35.2% better than query likelihood, 30.2% better than the bootstrap SVM, and 26.4% better than the passage relevance model.

Table 1: Mean bpref at min(12,000, R) characters

Model	**bpref @12K**	2-tail t-test			
		% improvement			
		TFIDF	QL	SVM	RMP
TFIDF	0.1733				
QL	0.1778	0.57			
SVM	0.1846	0.17	0.53		
RMP	0.1902	0.07	0.32	0.61	
QWD	**0.2108**	**0.02**	**0.03**	0.15	0.13
		17.8%	18.5%		
RMD	0.2355	0.09	0.08	0.18	0.18
MM	**0.2404**	**0.03**	**0.05**	**0.05**	**0.05**
		38.8%	35.2%	30.2%	26.4%

The results in table 2 for precision at min(12,000, R) characters are similar to the results for bpref at 12,000 characters. The mixture model performs 34.8% better than TFIDF, 30.3% better than query likelihood and 27.4% better than the SVM. The results are not significantly better than the other three models.

Table 2: Mean precision at min(12,000, R) characters

Model	**prec @12K**	2-tail t-test			
		% improvement			
		TFIDF	QL	SVM	RMP
TFIDF	0.1972				
QL	0.2040	0.42			
SVM	0.2088	0.38	0.72		
RMP	0.2181	0.12	0.34	0.46	
QWD	**0.2427**	**0.01**	**0.02**	0.09	0.09
		23.1%	19.0%		
RMD	0.2654	0.06	0.08	0.15	0.19
MM	**0.2659**	**0.02**	**0.04**	**0.05**	0.07
		34.8%	30.3%	27.4%	

Table 3: R-precision for fixed-length passage retrieval

Model	**char R-prec**	2-tail t-test			
		% improvement			
		QL	TFIDF	QWD	SVM
QL	0.1705				
TFIDF	0.1718	0.78			
QWD	0.1720	0.90	0.99		
SVM	0.1807	0.11	0.17	0.48	
RMP	0.1846	0.17	0.15	0.36	0.69
RMD	0.2018	0.22	0.24	0.30	0.36
MM	**0.2168**	**0.04**	**0.04**	0.10	0.08
		27.2%	26.1%		

Table 3 presents the results for character-level R-precision. Although we should note that all of these models were trained to optimize the high-precision metrics, and were not retrained to optimize for the measures that balance precision and recall, the relative performance of the seven models is still similar. Here, the mixture model still performs well, but is only significantly better than query likelihood and TFIDF, by 27.2% and 26.1%, respectively.

Tables 4 and 5 show the results for bpref at R characters and non-interpolated character average precision (CAP), which exhibit trends similar to the other tables. One notable exception is that the query word density (QWD) model seems to do quite a bit worse on the measures that balance precision and recall, than on the high-precision measures.

The mixture model performs well compared to the other models, and is robust across the various measures, but all of the scores are still quite low overall. This indicates that finding only the relevant portions of a document is a difficult problem, at least for TREC-style topics.

One interesting observation about these results is that the relevance model document retrieval algorithm performs almost as well as our best passage retrieval model, for all of these measures.[1] In fact, several 2004 HARD track participants working on passage retrieval made the same observations about their results [1, 15].

Table 4: Binary precision at R characters

Model	**bpref @R**	2-tail t-test			
		% improvement			
		QL	TFIDF	QWD	SVM
QL	0.1284				
TFIDF	0.1309	0.60			
QWD	0.1357	0.40	0.60		
SVM	**0.1413**	**0.04**	**0.03**	0.54	
		10.1%	8.0%		
RMP	**0.1447**	**0.04**	**0.03**	0.33	0.57
		12.7%	10.6%		
RMD	0.1723	0.06	0.09	0.16	0.14
MM	**0.1798**	**0.03**	**0.05**	0.11	0.08
		40.1%	37.4%		

Table 5: Non-interpolated character average precision (CAP)

Model	**CAP**	2-tail t-test			
		% improvement			
		QWD	TFIDF	QL	SVM
QWD	0.1077				
TFIDF	0.1139	0.56			
QL	0.1165	0.39	0.76		
SVM	**0.1310**	**0.05**	**0.05**	**0.01**	
		21.6%	15.0%	12.4%	
RMP	**0.1332**	**0.03**	**0.04**	**0.00**	0.53
		23.7%	16.9%	14.3%	
RMD	**0.1710**	**0.02**	**0.04**	**0.01**	**0.04**
		58.8%	50.1%	46.8%	30.6%
MM	**0.1718**	**0.02**	**0.05**	**0.02**	0.06
		59.5%	50.8%	47.5%	

We investigated several hypotheses about why document retrieval algorithms perform as well as the best passage retrieval algorithms. One observation is that the documents in the HARD 2004 corpus are quite short in general; they average 2,507 characters, which is generally equal to a passage containing about 200 non-stop words. Past research has found that passage retrieval performance (evaluated using document retrieval) is fairly consistent for passages of

[1] One property of all of the passage runs presented here is that they can never retrieve the characters in the beginning of all documents in tags like KEYWORD and DATE_TIME, which are guaranteed to be non-relevant. This gives all of the passage runs an advantage over the document runs from the start because document runs must return entire documents, which include all of this non-relevant text. Even worse, this text tends to occur more at the beginning of documents, further disadvantaging document runs on the measures, like bpref and average precision, where character ranking matters. It seems likely that removing these parts of the text from the documents returned could remove any advantage that the mixture model (MM) has over the document retrieval relevance model (RMD).

150-300 words [7]. In addition, in the documents with at least one passage marked relevant by HARD annotators, an average of 51% percent of the characters were marked relevant. Considering that every document includes some markup that is automatically non-relevant, this means that on average, annotators marked substantially more than half of the document text relevant.

In light of these properties of the corpus, we performed some analysis, designed to give an indication of the relative performance of passage and document retrieval algorithms on a corpus with more multi-topic documents. Table 6 shows how each of the seven retrieval methods performs on each of the five evaluation measures, averaged over the 14 topics where the average percentage of characters marked relevant in documents with relevant passages was $\leq 35\%$. The cells shaded gray indicate the two best-performing models for each measure. Although we want to be careful not to draw too many conclusions from an experiment on only 14 topics, it is worth noting that the RMD document retrieval model is one of the worst-performing models for measures except character average precision (CAP). For the CAP measure, however, RMD is the second best model.

Table 6: Evaluation using topics where an average of $\leq 35\%$ of characters were marked relevant in documents with a relevant passage. Gray-shaded boxes indicate the best performing measures.

	Bpref12K	Prec12K	RPrec	BprefR	CAP
RMD	0.0811	0.1115	0.0980	0.0732	0.0862
QL	0.0949	0.1284	0.1192	0.0794	0.0790
TFIDF	0.0970	0.1327	0.1170	0.0782	0.0809
RMP	0.0979	0.0950	0.1138	0.0800	0.0850
SVM	0.1094	0.1428	0.1206	0.0819	0.0860
MM	0.1142	0.1482	0.1238	0.0888	0.0959
QWD	0.1175	0.1540	0.1225	0.0872	0.0748

For our next analysis, we removed every document that had more than 50% of its total number of characters marked relevant from each ranked result list and from the relevance judgments. The results are shown in table 7.[2] Here we have to be even more careful about drawing conclusions because we have removed a great deal of relevant material from the judgments. This means relevant material is very sparse, scores are even lower, and small differences or anomalies could have a big effect. However, we note once again that the document retrieval model RMD is the worst performer on four of the five measures. For the CAP measure, RMD is somewhere in the middle.

Table 7: Results after removing all documents with more than 50% of characters marked relevant from the results and from the relevance judgments. Gray-shaded cells indicate the two best-performing models for each evaluation measure.

	Bpref12K	Prec12K	RPrec	BprefR	CAP
RMD	0.0576	0.0803	0.0879	0.0593	0.0640
QL	0.0703	0.1061	0.0905	0.0651	0.0612
SVM	0.0723	0.0994	0.0954	0.0658	0.0642
TFIDF	0.0769	0.1076	0.0913	0.0655	0.0623
MM	0.0710	0.0977	0.0970	0.0684	0.0744
RMP	0.0805	0.1010	0.0920	0.0701	0.0692
QWD	0.0808	0.1177	0.0969	0.0687	0.0571

5. CONCLUSION

We have demonstrated some of the problems that can arrive in passage retrieval evaluation and presented five retrieval methods that correct some of these problems. Using these metrics, we compared the retrieval performance of seven different passage retrieval algorithms. We found that the algorithms that tended to do well on these evaluation metrics did well on all of them. Of particular note is our new relevance modeling-based mixture model that combines document and passage information to score passages. This model performed the best on all five evaluation metrics. We saw that the one document retrieval algorithm we used performed very well. However, our last two experiments suggest that this may be due to the short documents in the corpus and the presence of many documents with a lot of relevant text. The relatively low scores for all of the retrieval methods suggest that we have a long way to go in improving passage retrieval. We feel that future passage retrieval researchers should choose evaluation measures with a specific task in mind, and that passage retrieval corpora should contain longer, multi-topic documents.

6. ACKNOWLEDGEMENTS

We would like to thank the other members of the UMass TREC HARD 2004 group for their contributions for this work: Nasreen Abdul-Jaleel, Fernando Diaz, Mark Smucker, and Leah Larkey. Nasreen first applied the SVM model to passage retrieval and pointed out many of the problems with passage R-precision as an evaluation metric. Fernando was instrumental in extending the Lemur toolkit to make all of these experiments possible. Mark came up with the idea of character-level passage retrieval evaluation.

This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-02-1-8903. Any opinions, findings and conclusions or recommendations expressed in this material are the author(s) and do not necessarily reflect those of the sponsor.

7. REFERENCES

[1] N. AbdulJaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, D. Metzler, M. D. Smucker, T. Strohman, H. Turtle, and C. Wade. UMass at TREC 2004: Notebook. In *Text REtrieval Conference (TREC 2004)*, 2004.

[2] S. Al-Hawamdeh and P. Willett. Paragraph-based searching in full-text documents. *Electronic Publishing*, 2(4):179–192, 1988.

[3] J. Allan. HARD track overview in TREC 2003 high accuracy retrieval from documents. In *NIST Special*

[2] For the models using some kind of pseudo-relevance feedback (RMD, MM, RMP, and QWD), we did *not* remove documents $\geq 50\%$ relevant from the modified query, as we should have to make this a more fair experiment.

Publication 500-255:The Twelfth Text REtrieval Conference (TREC 2003), pages 24–37, 2004.

[4] J. Allan. HARD track overview in TREC 2004: High accuracy retrieval from documents. In *TREC 2004*, 2005.

[5] J. Allan, J. Callan, K. Collins-Thompson, B. Croft, F. Feng, D. Fisher, J. Lafferty, L. Larkey, T. N. Truong, P. Ogilvie, L. Si, T. Strohman, H. Turtle, and C. Zhai. The lemur toolkit for language modeling and information retrieval. http://www.cs.cmu.edu/∼lemur/, 2003.

[6] C. Buckley and E. M. Voorhees. Retrieval evaluation with incomplete information. In *SIGIR '04*, pages 25–32. ACM Press, 2004.

[7] J. P. Callan. Passage-level evidence in document retrieval. In *SIGIR '94*, pages 302–310. Springer-Verlag New York, Inc., 1994.

[8] C. L. A. Clarke and G. V. Cormack. Interactive substring retrieval: Multitext experiments for TREC5. In E. M. Voorhees and D. K. Harman, editors, *NIST Special Publication 500-238: The Fifth Text REtrieval Conference (TREC-5)*, pages 267–278, 1996.

[9] A. Corrada-Emmanuel, W. B. Croft, and V. Murdock. Answer passage retrieval for question answering. CIIR Technical Report, 2003.

[10] O. de Kretser and A. Moffat. Effective document presentation with a locality-based similarity heuristic. In *SIGIR '99*, pages 113–120. ACM Press, 1999.

[11] L. Denoyer, H. Zaragoza, and P. Gallinari. HMM-based passage models for document classification and ranking. In *Proceedings of ECIR-01, 23rd European Colloquium on Information Retrieval Research*, pages 126–135, Darmstadt, DE, 2001.

[12] M. A. Hearst and C. Plaunt. Subtopic structuring for full-length document access. In *SIGIR '93*, pages 59–68. ACM Press, 1993.

[13] D. Hiemstra and W. Kraaij. Twenty-one at TREC-7: ad-hoc and cross-language track. In E. M. Voorhees and D. K. Harman, editors, *NIST Special Publication 500-242: The Seventh Text REtrieval Conference (TREC-7)*, pages 227–238, 1998.

[14] F. Jelinek and R. L. Mercer. Interpolated estimation of Markov source parameters from sparse data. In *Proceedings of the Workshop on Pattern Recognition in Practice*, Amsterdam, The Netherlands: North-Holland, May 1980.

[15] J. Jiang and C. Zhai. UIUC in HARD 2004–passage retrieval using HMMs. In *Text REtrieval Conference*, 2004.

[16] M. Kaszkiel and J. Zobel. Passage retrieval revisited. In *SIGIR '97*, pages 178–185. ACM Press, 1997.

[17] D. Knaus, E. Mittendorf, P. Schäuble, and P. Sheridan. Highlighting relevant passages for users of the interactive spider retrieval system. In D. K. Harman, editor, *NIST Special Publication 500-236: The Fourth Text REtrieval Conference (TREC-4)*, pages 233–244, 1996.

[18] J. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *SIGIR '01*, pages 111–119. ACM Press, 2001.

[19] V. Lavrenko and W. B. Croft. Relevance-based language models. In *SIGIR '01*, pages 120–127. ACM Press, 2001.

[20] X. Liu and W. B. Croft. Passage retrieval based on language models. In *Proceedings of the Eleventh International Conference on Information and Knowledge Management*, pages 375–382. ACM Press, 2002.

[21] D. R. H. Miller, T. Leek, and R. M. Schwartz. A hidden markov model information retrieval system. In *SIGIR '99*, pages 214–221. ACM Press, 1999.

[22] E. Mittendorf and P. Schäuble. Document and passage retrieval based on hidden markov models. In *SIGIR '94*, pages 318–327. Springer-Verlag New York, Inc., 1994.

[23] A. Moffat, R. Sacks-Davis, R. Wilkinson, and J. Zobel. Retrieval of partial documents. In D. K. Harman, editor, *NIST Special Publication 500-215: The Second Text REtrieval Conference (TREC-2)*, pages 181–190, 1994.

[24] R. Nallapati. Discriminative models for information retrieval. In *SIGIR '04*, pages 64–71. ACM Press, 2004.

[25] J. O'Connor. Retrieval of answer-sentences and answer-figures from papers by text searching. *Information Processing and Management*, 11(5/7):155–164, 1975.

[26] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98*, pages 275–281. ACM Press, 1998.

[27] G. Salton, J. Allan, and C. Buckley. Approaches to passage retrieval in full text information systems. In *SIGIR '93*, pages 49–58. ACM Press, 1993.

[28] F. Song and W. B. Croft. A general language model for information retrieval. In *CIKM '99: Proceedings of the eighth international conference on Information and knowledge management*, pages 316–321. ACM Press, 1999.

[29] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question a nswering. In *SIGIR '03*, pages 41–47. ACM Press, 2003.

[30] V. Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag, 1995.

[31] R. Wilkinson. Effective retrieval of structured documents. In *SIGIR '94*, pages 311–317. Springer-Verlag New York, Inc., 1994.

[32] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, 2004.

[33] D. Zhang and W. S. Lee. A language modeling approach to passage question answering. In *NIST Special Publication 500-255:The Twelfth Text REtrieval Conference (TREC 2003)*, 2004.