Passage Retrieval Based On Language Models

Xiaoyong Liu
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
xliu@cs.umass.edu

W. Bruce Croft
Center for Intelligent Information Retrieval
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
croft@cs.umass.edu

ABSTRACT
Previous research has shown that passage-level evidence can bring added benefits to document retrieval when documents are long or span different subject areas. Recent developments in language modeling approach to IR provided a new effective alternative to traditional retrieval models. These two streams of research motivate us to examine the use of passages in a language model framework. This paper reports on experiments using passages in a simple language model and a relevance model, and compares the results with document-based retrieval. Results from the INQUERY search engine, which is not based on a language modeling approach, are also given for comparison. Test data include two heterogeneous and one homogeneous document collections. Our experiments show that passage retrieval is feasible in the language modeling context, and more importantly, it can provide more reliable performance than retrieval based on full documents.

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

General Terms
Theory, Experimentation.

Keywords
Passage Retrieval, Language Model, Information Retrieval

1. INTRODUCTION
Information Retrieval (IR) is the process of locating and retrieving documents relevant to a user's information need from a collection of documents. The user's information need is presented to the IR system as a query which usually consists of a string of words. The IR system uses a matching mechanism to decide how closely a document is related to the query. The matching mechanism is described through retrieval models, and the question of whether the entire document or some portions of it should be used for the matching has been the subject of passage retrieval research. Different passage types include structural [4, 7], semantic [6, 14, 18], window-based [4, 22], and arbitrary [8, 9].

Recently, new retrieval approaches using generative models of documents and queries ("language models") have been introduced to IR [15, 13, 19, 2, 10, 11]. This approach has shown promise as a formal framework for describing a range of retrieval processes, such as query expansion and cross-lingual retrieval, and has produced excellent results using evaluation testbeds such as TREC. Given that the research on language modeling has been entirely document-based, in this paper we address the question of whether passages can be used effectively in this framework. We examine the use of a range of passage types with two language modeling approaches, and compare retrieval results across different test collections taken from TREC.

The remainder of the paper is organized as follows. Section 2 gives a brief review of the language modeling approach to IR and passage retrieval. In section 3, we describe the experimental methods of this study. Section 4 presents the empirical results on three data sets. Conclusions and contributions of this work are summarized in section 5.

CIKM'02, November 4-9, 2002, McLean, Virginia, USA.

2. LANGUAGE MODELS AND PASSAGES
The inspiration and foundation of the present work comes from two streams of research: statistical language modeling and passage retrieval. Since 1980 when the first significant language model was proposed [16], statistical language modeling has become a fundamental component of speech recognition, machine translation, spelling correction, and so forth. It has also proven useful for natural language processing tasks such as natural language generation and summarization. More recently, the language modeling framework has been introduced to information retrieval, and several approaches have been used to adopt this new framework and improve retrieval effectiveness. Unlike the language modeling approach, passage retrieval techniques have been extensively studied and applied to IR. This section gives a brief review of past research in these two areas and highlights those closely related to the present work.

2.1 Language Models
A language model is a probability distribution that captures the statistical regularities of natural language use [15, 16]. The task of language modeling, in general, answers the question: how likely the *i*th word in a sequence would occur given the identities of the preceding *i*-1 words? Applied to information retrieval, language modeling refers to the problem of estimating the

likelihood that a query and a document could have been generated by the same language model, given the language model of the document and with or without a language model of the query.

One language modeling approach to IR is to model the query generation process. The general idea is to build a language model M_d for each document d in the collection, and rank the documents according to how likely the query Q could have been generated from each of these document models, i.e. $P(Q|M_d)$. Different models calculate this probability in a different way. Ponte and Croft [15] treat the query Q as a set of unique terms, and use the product of two probabilities – the probability of producing the query terms and the probability of not producing other terms - to approximate $P(Q|M_d)$. Multiple occurrences of the same term in a query are not considered.

$$P(Q \mid M_d) = \prod_{w \in Q} P(w \mid M_d) \prod_{w \notin Q} (1.0 - P(w \mid M_d)) \qquad (1)$$

where $P(w|M_d)$ is calculated by a non-parametric method that makes use of the average probability of w in documents containing it and a risk factor. For non-occurring terms, the global probability of w in the collection is used instead.

Hiemstra [5], Miller et al. [13], and Song and Croft [19] treat the query Q as a sequence of independent terms, taking into account possibly multiple occurrences of the same term. Thus the query probability can be obtained by multiplying the individual term probabilities.

$$P(Q \mid M_d) = \prod_{i=1}^{n} P(w_i \mid M_d) \qquad (2)$$

where w_i is the ith term in the query. While through different theoretical derivations, these models all arrived at a similar way of computing $P(w|M_d)$, combining the document model and the collection model by linear interpolation.

$$P(w \mid M_d) = \lambda \frac{tf(w,d)}{dl_d} + (1-\lambda)\frac{cf_w}{cs} \qquad (3)$$

where λ is a weighting parameter between 0 and 1, $tf(w, d)$ is the number of times w occurs in d, dl_d is the document length of d, cf_w is the number of times w occurs in the entire collection, and cs is the total number of tokens in the collection.

We use the preceding formulation as specified in equation (2) and (3) as our simple language model in this study, because it is relatively more commonly used. We report results of this language model with two different smoothing parameters λ: the Dirichlet prior and the Jelinek-Mercer smoothing parameter [21].

Taking a different angle, Berger and Lafferty [2] view a query as a distilment or translation from a document. To determine the relevance of a document to a query, their model estimates the probability that the query would have been generated as a translation of that document. Documents are then ranked according to these probabilities. One notable feature of this model is an inherent query expansion component [11, 10]. However, there are also difficulties with application of this model: the need of a large collection of training data for translation probabilities, and inefficiency for ranking documents [10, 11].

Lafferty and Zhai [10] proposed a new framework that extends the existing language modeling approach to IR to estimating language models not only for documents but also for queries. The similarity between a document and a query is measured by the Kullback-Leibler (KL) divergence between the document model and the query model. This framework bears resemblance to the classical probabilistic retrieval models and can accommodate existing language models proposed by Ponte and Croft [15] and others. They also introduced the idea of estimating expanded query language models for which they used a Markov chain method.

Instead of attempting to model the query generation process, Lavrenko and Croft [11] explicitly model relevance, and developed a novel technique that estimates a relevance model from query alone, with no training data. They assume that, given a collection of documents and a user query Q, there exists an unknown relevance model R that assigns the probabilities $P(w|R)$ to the word occurrence in the relevant documents. The relevant documents are random samples from the distribution $P(w|R)$. Both the query and the documents are samples from R. The essence of their model is to estimate $P(w|R)$. Let $P(w|R)$ denotes the probability that a word sampled at random from a relevant document would be the word w. If we know what documents are relevant, estimation of these probabilities would be straightforward, but in a typical retrieval environment we are not given any examples of relevant documents. Lavrenko and Croft [11] and Lavrenko et al. [12] suggest a reasonable way to approximate $P(w|R)$ by using a joint probability of observing the word w together with query words $q_1, \ldots, q_m$ ($Q = q_1, \ldots, q_m$):

$$P(w \mid R) \approx P(w \mid Q) = \frac{P(w, q_1 \ldots q_m)}{P(q_1 \ldots q_m)} = \frac{P(w, q_1 \ldots q_m)}{\sum_{v \in vocabulary} P(v, q_1 \ldots q_m)} \qquad (4)$$

Two methods of estimating the joint probability $P(w, q_1, \ldots, q_m)$ are described in [11]. Both methods assume the existence of a set U of underlying source distributions from which $w, q_1, \ldots, q_m$ could have been sampled. They differ in their independence assumptions. In this paper, we consider only Method 1 because of its relative simplicity, and its decomposability [12]. Method 1 assumes w and $w, q_1, \ldots, q_m$ to be mutually independent once we pick a source distribution from U. If we assume U to be the set of document models, one for each document in the collection, then we get:

$$P(w, q_1 \ldots q_m) = \sum_{M \in U} P(M) \left(P(w \mid M) \prod_{i=1}^{m} P(q_i \mid M) \right) \qquad (5)$$

Here $P(M)$ denotes some prior distribution over the set U which is usually taken to be uniform, while $P(w|M)$ specifies the probability of observing w if we samples a random word from M. $P(w|M)$ is computed using equation (3). Lavrenko et al. [12] use the KL divergence between the relevance model and the document model to rank documents. Documents with smaller divergence with the relevance model are considered more relevant.

$$KL(R \,||\, M_d) = \sum_{w} P(w \mid R) \log \frac{P(w \mid R)}{P(w \mid M_d)} \qquad (6)$$

In the present study, we choose as basis the formulation specified in equations (5) and (6) for the relevance model later used in the experiments.

2.2 Passage Retrieval

Language modeling is a new framework for IR, and to learn more about this framework it is important to study how well-known IR techniques can be implemented, and whether there are differences in performance from what has previously been observed. Passage retrieval techniques have been extensively used in standard IR settings, and have proven effective for document retrieval when documents are long or when there are topic changes within a document, thus making it an appealing candidate for the present work. Second, from an IR system user's standpoint, it may be more desirable that the relevant section of a document is presented to the user than the entire document.

Passages can be defined based on the document structure [4, 7, 17]. This entails using author-provided marking (e.g. period, indentation, empty line, etc.) as passage boundaries. Examples of such passages include paragraphs, sections, or sentences. Passages can also be defined according to subject or content of the text. The main idea is to divide documents into coherent units with each unit corresponding to a subtopic. A well-known algorithm for deriving such passages is TextTiling [6, 7]. Other algorithms have been reported in [17, 14, 18]. The third type of passage is window, which consists of a fixed number of words or bytes. Passages in this category may or may not take logical structure of the document into account. Overlapped windows as used in [4] and non-overlapped windows as used in [9] do not depend on text, whereas pages in [22] and bounded paragraphs in [4] make use of paragraph boundary information and restrict windows to some minimum length. A more dynamic alternative to windows is arbitrary passages proposed by [8, 9]. The word "arbitrary" means that a passage can start at any word in the document. Two subclasses are further defined. Fixed-length arbitrary passages resemble overlapped windows but with an arbitrary starting point. Variable-length arbitrary passages can be of any length. Unlike structural, topical, and window passages which are typically predefined (defined before or at indexing time), arbitrary passages are defined at query time. A survey of passages can be found in [9].

We test on two kinds of passages in this study: half-overlapped windows and arbitrary passages. We define half-overlapped windows in a similar way to that of [4] with slight variation: the first window starts at the first term in a document, and subsequent windows start at the middle of the previous one. The definition of arbitrary passages follows that in [9]. Half-overlapped windows stand as an example of predefined passages and they have been shown in standard IR setting to be at least as effective as and more efficient than other pre-defined passages [4, 1, 3]. In addition, using this type of passages makes our experimental results directly comparable with those from INQUERY, a probabilistic retrieval system based on the inference net model [20], as it uses the same passage type for retrieval. Arbitrary passages provide a case of query-time passages and have demonstrated, in standard IR setting, improved retrieval effectiveness over predefined passages [9]. The experimental methods are discussed in the next section.

3. METHODOLOGY

3.1 Experimental Methods

We use the simple language model (LM) and the relevance model (RM) described in section 2.1 for retrieval. In order to address the research questions, we experiment with a few ways to retrieve against passages. In the context of the simple language model, passage retrieval is done using method L1. All the documents are segmented into passages and a language model is built for each passage. Passages are then ranked according to the probability that the query could have been generated by each of them. Documents are finally ranked based on the score of their best passage.

In the context of the relevance model, we have developed the following 3 methods for passage retrieval. Method R1 breaks documents into passages and builds a language model for each passage. It then builds a relevance model using the query and passages. A KL divergence score is computed between each passage model and the relevance model, and is used as the basis for the ranking of passages. Documents are ranked based on the score of their best passage.

Both Method R2 and R3 are a variation of R1. Method R2 differs from R1 in two steps. Unlike R1 which constructs language models only for passages, method R2 builds a language model for each passage and one for each document. Furthermore, the relevance model in R2 is constructed using documents instead of passages. Method R3 is similar to method R1 in that it also builds a language model for each passage and the relevance model is constructed using passages. In addition to these, R3 also builds a language model for every document. Document language models, instead of passage models, are used for computing the KL divergence score, and are therefore used directly in the ranking step. Table 1 gives a summary of the retrieval steps of the aforementioned methods, and illustrates the differences between them.

3.2 Data Set and Experiments

The test data consists of three sets: TREC queries 51-150 over collection AP, TREC queries 51-100 over collection FR-12, and TREC queries 301-400 over collection TREC-45. AP corresponds to Associated Press newswire 1988, 1989, and 1990, from TREC disk 2, 1, and 3 respectively. It is chosen as a homogenous collection. FR-12 refers to Federal Register 1988 and 1989 from TREC disk 2 and 1 respectively. It is selected as a collection of long documents, with a large variance in the document length and often shifts in topic. TREC-45 is a heterogeneous collection that is composed of all data from TREC disk 4 and 5. Statistics of the test collections are given in table 2.

Queries are taken from the "title" field of TREC topics. They range from 1 word to as many as 11 words. Relevance judgments are obtained from the judged pool of top retrieved documents by various participating retrieval systems. Table 3 summarizes information about queries and relevance judgments on different collections.

Four sets of experiments are performed. The first set of experiments investigates whether passage retrieval is feasible in the context of the simple language model. We experiment with two smoothing parameter settings - the Dirichlet prior and the Jelinek-Mercer smoothing parameter. LM(dir-1000) refers to the experiment with the Dirichlet prior set to 1000. LM(lin-0.5) corresponds to the experiment with the Jelinek-Mercer smoothing

Table 1. Relationship between the experimental methods to passage retrieval

Retrieval Steps	LM	RM		
	L1	R1	R2	R3
1. Break each document into passages	*	*	*	*
2a. Build a language model for each passage	*	*	*	*
2b. Build a language model for each document			*	*
3a. Build a relevance model using the query and passages		*		*
3b. Build a relevance model using the query and documents			*	
4a. Compute the KL divergence score between each passage model and the relevance model		*	*	
4b. Compute the KL divergence score between each document model and the relevance model				*
5a. Rank passages according to the probability that the query could have been generated by each of those passages	*			
5b. Rank passages according to their KL divergence scores		*	*	
6a. Rank documents according to their KL divergence scores				*
6b. Rank documents using the score of the best passage	*	*	*	

Table 2. Statistics of test collections

Collection	# of Docs	Size	Average # of Words/Doc[1]	Std Dev. of Doc length	Contents
AP	242,918	0.73 Gb	273.3	132.72	Associated Press newswire 1988-90 (from TREC disk 1-3).
FR-12	45,820	0.47 Gb	873.9	2514.16	Federal Register 1988-89 (from TREC disk 1-2).
TREC-45	556,077	2.13 Gb	305.3	775.78	The Financial Times 1991-94, Federal Register 1994, Congressional Record 1993, Foreign Broadcast Information Service, the LA Times.

Table 3. Query set information.

Collection	Queries	# of Queries with Relevant Docs	Average Query Length[2]	Std. Dev. of Query Length	# of Relevant Documents			
					Total	Per Query (for queries with rel. docs)		
						Avg	Min	Max
AP	TREC topics 51-150 (title only)	99	4.3	2.22	21819	220.4	2	1142
FR-12	TREC topics 51-100 (title only)	21	4.2	2.37	502	23.9	1	118
TREC-45	TREC topics 301-400 (title only)	100	2.5	0.71	9285	92.9	3	474

[1] After the application of stemming and stopword removal.

[2] Only the queries with at least one relevant document are used.

Table 4. Results for passage retrieval using simple language model with half-overlapped windows. % chg is based on full-length document retrieval. FR-12, AP, and TREC-45 collections.

		Doc	LM with Half-Overlapped Windows					
			Psg-50	% chg	Psg-150	% chg	Psg-350	% chg
FR-12	LM (dir-1000)	0.2875	0.3065	+6.6	0.2613	-9.1	0.2894	+0.7
	LM (lin-0.5)	0.2204	0.3075	+39.5	0.2751	+24.8	0.2684	+21.8
AP	LM (dir-1000)	0.2187	0.1795	-17.9	0.2067	-5.5	0.2159	-1.3
	LM (lin-0.5)	0.2043	0.1661	-18.7	0.1765	-13.6	0.1823	-10.8
TREC-45	LM (dir-1000)	0.2011	-	-	0.1903	-5.37	0.1977	-1.69
	LM (lin-0.5)	0.1949	-	-	0.1781	-8.62	0.1884	-3.34

parameter set to 0.5. Half-overlapped windows of size 50, 150, and 350 are used for retrieval and results are compared to full-length document retrieval results by this model. The second set of experiments investigates the applicability of passage retrieval in the context of the relevance model. We use the three methods as described in section 3.1 to retrieve against half-overlapped windows. Again, windows of size 50, 150, and 350 are used and results are compared to full-length document retrieval results. The third set of experiments is designed for investigating whether a different type of passages would yield drastically different results. Both fixed-length and variable-length arbitrary passages are evaluated. Finally, for comparison with results produced by a standard IR system, we perform both document and passage retrieval experiments using INQUERY. The 11-point average precision is used as the basis of evaluation throughout this study.

4. EMPIRICAL RESULTS

As we mentioned earlier, four sets of experiments are carried out in this study. In all experiments, both the queries and documents are stemmed. Stopwords are removed based on the standard INQUERY stoplist of 418 words [11].

4.1 Simple Language Model with Half-Overlapped Windows

This set of experiments compares the results of retrieving against half-overlapped windows with retrieving against full-length documents, using the simple language model. Table 4 summarizes the results of these experiments and shows the average precision score and the percentage increase or decrease over full-length document retrieval for half-overlapped windows of size 50, 150, and 350 words.

The first experiment is performed using TREC title queries 51-100 on FR-12 collection. From Table 4, we observe that passage retrieval using different smoothing parameters yields consistent results while full-length document retrieval with different parameters differ considerably in retrieval performance. For example, the document-level performance of LM(dir-1000) and LM(lin-0.5) changes from 0.2875 to 0.2204 in average precision, but the passage-level performance of the two stay very close. The best results are obtained by using windows of size 50. Table 5 reports the results for the simple language model with Jelinek-Mercer smoothing parameter. There are significant improvements at many recall levels over full-length document retrieval using the same parameter settings.

Table 5. Simple language model (Jelinek-Mercer smoothing) with half-overlapped windows. FR-12 collection, queries 51-100.

	LM (lin-0.5) on FR-12						
	Doc	Psg-50	% chg	Psg-150	% chg	Psg-350	% chg
Rel.	502	502	-	502	-	502	-
Rel. Retr.	249	265	+6.4	282	+13.3	287	+15.3
Prec.							
0.00	0.4184	0.4758	+13.7	0.4250	+1.6	0.4566	+9.1
0.10	0.3114	0.4361	+40.0	0.3826	+22.9	0.3763	+20.8
0.20	0.2817	0.3896	+38.3	0.3743	+32.9	0.3567	+26.6
0.30	0.2524	0.3465	+37.3	0.3360	+33.1	0.3195	+26.6
0.40	0.2374	0.2940	+23.8	0.2794	+17.7	0.2649	+11.6
0.50	0.2320	0.2937	+26.6	0.2708	+16.7	0.2593	+11.8
0.60	0.1947	0.2834	+45.6	0.2536	+30.3	0.2386	+22.6
0.70	0.1835	0.2757	+50.3	0.2425	+32.2	0.2330	+27.0
0.80	0.1619	0.2549	+57.4	0.2179	+34.6	0.2067	+27.7
0.90	0.1519	0.2318	+52.6	0.1779	+17.1	0.1837	+20.9
1.00	0.1131	0.2156	+90.6	0.1619	+43.6	0.1569	+38.7
Avg	0.2204	0.3075	+39.5	0.2751	+24.8	0.2684	+21.8

The results for TREC title queries 51-150 on the AP collection and TREC title queries 301-400 on the TREC-45 collection are also given in Table 4. In the experiment on the AP collection, passage retrieval produces losses ranging from 1.28% to 18.7%. We also observe that with an increase in passage size from 50 to 350, losses decrease. Similar results are found in the experiment on the TREC-45 collection.

4.2 Relevance Model with Half-Overlapped Windows

Our next set of experiments compares the results of RM with different passage retrieval methods as described in section 3.1. Again, full-length document retrieval results are used as baseline for comparison. Similar to LM experiments reported in section 4.1, RM experiments are also performed on three data sets. In experiments on AP and TREC-45, we see from Table 7 that,

amongst three passage retrieval methods, method R3 is as good as full-length document retrieval, and method R1 and R2 have virtually no difference in performance. Windows of size 150 produce the best results, which are comparable with those of document-level retrieval. On FR-12, however, significant improvements over document-level results are found for almost all experiments except one. The results for windows of size 50 are shown in Table 6. Method R1 and R2 have significant improvements over full-length document retrieval at all recall levels, while results of method R3 stays close to document-level results. Method R1 consistently produces better results than the other two on all three window sizes.

Table 6. Relevance model with half-overlapped windows. Window size = 50 words. FR-12 collection, queries 51-100.

	Doc	Psg-50					
		Method R1	% chg	Method R2	% chg	Method R3	% chg
Rel.	502	502	-	502	-	502	-
Rel. Retr.	288	279	-3.1	279	-3.1	288	0.0
Prec.							
0.00	0.2463	0.4095	+66.3	0.3621	+47.0	0.2478	+0.6
0.10	0.2378	0.3997	+68.1	0.3523	+48.2	0.2393	+0.6
0.20	0.2251	0.3815	+69.5	0.3341	+48.4	0.2266	+0.7
0.30	0.1995	0.3511	+76.0	0.3037	+52.2	0.2010	+0.8
0.40	0.1788	0.3297	+84.4	0.2822	+57.8	0.1803	+0.8
0.50	0.1746	0.3179	+82.1	0.2705	+54.9	0.1761	+0.9
0.60	0.1088	0.3053	+180.6	0.2578	+137.0	0.1103	+1.4
0.70	0.1062	0.2880	+171.2	0.2405	+126.5	0.1077	+1.4
0.80	0.0866	0.2694	+211.1	0.222	+156.4	0.0881	+1.7
0.90	0.0825	0.2516	+205.0	0.2042	+147.5	0.0840	+1.8
1.00	0.0468	0.2367	+405.8	0.1893	+304.5	0.0483	+3.2
Avg	0.1486	0.3177	+113.8	0.2702	+81.8	0.1501	+1.0

4.3 Experiments with Arbitrary Passages

In order to investigate whether a different type of passage would yield dramatically different results, we also used arbitrary passages as described in [9] for retrieval. In the experimental runs, we followed the example of [9] by using passages starting at every 25th word instead of every word in a document, to limit the cost of ranking passages. Previous research showed that arbitrary passages starting at 25-word intervals were as effective as those that start at any word [8]. In the case of fixed-length arbitrary passages, passage size is set before query time. We experimented with passage size ranging from 50 to 600 words, in increments of 50. In the case of variable-length arbitrary passages, passages of different sizes between 50 and 600 (in increments of 50) are used at the same time. This set of experiments was performed using the AP data set. Results for variable-length arbitrary passages, as well as fixed-length arbitrary passages with size 50, 150, and 350, are shown in Table 8. The performance of the relevance model with fixed-length arbitrary passages is as good as and sometimes better than that of the relevance model with half-overlapped windows, at equivalent passage sizes. Relevance model with variable-length arbitrary passages gives the best result as compared with RM on half-overlapped windows and fixed-length arbitrary passages. It also outperforms the document-level retrieval on the same data set.

Kaszkiel and Zobel [9] observed, in standard IR setting, that the effectiveness of fixed-length arbitrary passages was not particularly sensitive to passage length. Our results confirm this, albeit in a different context. While there is a mild climb between passage size 50 and 100, performance stabilizes after size 200 (results are not shown).

4.4 Overall Comparison

The last set of experiments compares results of language models with those of INQUERY. Results are shown in Table 9. We observe that, on AP and TREC-45 data set, passage retrieval using INQUERY does not have a noticeable advantage over full-length document retrieval. On FR-12 data set, passage retrieval improves retrieval performance significantly. Experiments with LM and RM produced similar results in the language modeling context. Our study also confirms the results from previous research [4, 9] that, on collections of medium length or mixed documents such as AP and TREC-45, passage retrieval performance is comparable with and sometimes a little worse than full-length document retrieval performance; and, on collections of long documents or documents that span different subject areas, as exemplified by FR-12, passage retrieval gives significant improvements over full-length document retrieval. We also observe that passage retrieval in the language modeling framework sometimes provides more consistent performance than that of full-length document retrieval. For instance, on FR-12 collection, full-length document retrieval using RM results in a poor average precision of 0.1486, much worse than that produced by INQUERY. Passage retrieval is able to correct this and produces comparable performance with that of INQUERY. It is more difficult to obtain consistent results with the simple language model than with the relevance model. Except for the AP collection, the relevance model with half-overlapped windows of size 150 always gives as good a performance as and sometimes a better one than full-length document retrieval. Results from using arbitrary passages (Table 8) and using half-overlapped windows on the AP collection are similar, and they are slightly in favor of the former when RM is used. The relevance model with fixed-length arbitrary passages is as effective as and sometimes better than the relevance model with half-overlapped windows at the same passage sizes. RM with variable-length arbitrary passages yields the best result on AP, better than both the document-level retrieval and the passage retrieval using half-overlapped windows and fixed-length arbitrary passages. In the context of LM, variable-length arbitrary passages are as effective as fixed-length ones. Among the three approaches to passage retrieval with RM, the best result on FR-12 is provided by R1, with TREC-45, it is given by R1 and R2, and the best result on AP is produced by R3. If one had to choose a single consistently good method, R1 is the best candidate. In addition, Table 9 shows that, in general, INQUERY's performance improves with increasing window size from 50 to 350 words. INQUERY consistently achieves its best passage retrieval results with window size 350. On AP and TREC-45, LM behaves similarly to INQUERY while RM achieves its best results with window size 150. However, on FR-12, the best results for both LM and RM are obtained with a much smaller window size – 50 words.

5. CONCLUSION

This paper presented an examination of the applicability of passage retrieval within the language-modeling framework. The experiments were conducted with one relatively homogeneous collectios (AP), and two more heterogeneous collections (FR-12 and TREC-45). The experiments were intended to help understand whether passage retrieval can be applied in the language modeling context, how they can be applied, and what value they can add. We can draw the following conclusions based on the results reported here. First, passage retrieval can be successfully implemented in a language modeling environment. We tried various approaches to passage retrieval, all of which have produced comparable results with and sometimes significant improvements over full-length document retrieval. Second, passage retrieval can provide more reliable performance than full-length document retrieval in the language modeling context, especially when using relevance models which are a form of query expansion.

6. ACKNOWLEDGMENTS

We thank Victor Lavrenko for helpful discussions on this work. This work was supported in part by the Center for Intelligent Information Retrieval and in part by SPAWARSYSCEN-SD grant number N66001-99-1-8912. Any opinions, findings and conclusions or recommendations expressed in this material are the authors and do not necessarily reflect those of the sponsors.

7. REFERENCES

[1] Allan, J. (1995). Relevance feedback with too much data. In E. A. Fox, P. Ingwersen, & R. Fidel (Eds.), *Proceedings of the 18th annual international ACM-SIGIR conference on research and development in information retrieval*, Seattle, WA, July (pp. 337-343).

[2] Berger, A. and Lafferty, J. (1999). Information retrieval as statistical translation. In *Proceedings on the 22nd annual international ACM SIGIR conference*, pp. 222-229.

[3] Buckley, C., Salton, G., Allan, J., and Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. In *Third Text Retrieval Conference (TREC-3) proceedings*.

[4] Callan, J.P. (1994). Passage-level evidence in document retrieval. In B.W. Croft & C.J. van Rijsbergen (Eds.), *Proceedings of the 17th annual international ACM-SIGIR conference on research and developments in information retrieval*, Dublin, Ireland, July (pp. 302-310), New York: ACM.

[5] Hiemstra, D. (1998). A linguistically motivated probabilistic model of information retrieval. In *Proceedings of the Second European Conference on Research and Advance Technology for Digital Libraries (ECDL)*, pp. 569-584.

[6] Hearst, M. A. (1993). TextTiling, a quantitative approach to discourse segmentation. Technical Report 93/24 Sequoia 2000 Technical Report, University of California, Berkeley.

[7] Hearst, M.A., & Plaunt, C. (1993). Subtopic structuring for full-length document access. In R. Korfhage, E. Rasmussen, & P. Willet (Eds.), *Proceedings of the 16th annual international ACM-SIGIR conference on research and development in information retrieval*, Pittsburgh, PA (pp.59-68), New York: ACM.

[8] Kaszkiel, M. and Zobel, J. (1997). Passage retrieval revisited. In N. J. Belkin, D. Narasimhalu, & P. Willett (Eds.), *Proceedings of the 20th annual international ACM-SIGIR conference on research and development in information retrieval*, Philadelphia, PA (pp. 178-185).

[9] Kaszkiel, M. and Zobel, J. (2001). Effective ranking with arbitrary passages. *Journal of the American Society For Information Science and Technology*, 52(4):344-364.

[10] Lafferty, J. and Zhai, C. (2001). Document language models, query models, and risk minimization for information retrieval. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), *Proceedings of the 24th annual international ACM-SIGIR conference on research and development in information retrieval*, New Orleans, Louisiana (pp.111-119), New York: ACM.

[11] Lavrenko, V. and Croft, W.B. (2001). Relevance-based language models. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), *Proceedings of the 24th annual international ACM-SIGIR conference on research and development in information retrieval*, New Orleans, Louisiana (pp.120-127), New York: ACM.

[12] Lavrenko, V., Choquette, M., and Croft, W.B. (2002). Cross-lingual relevance models. To appear in *Proceedings of the 25th annual international ACM-SIGIR conference on research and development in information retrieval*.

[13] Miller, D., Leek, T., and Schwartz, R. (1999). A hidden Markov model information retrieval system. In *Proceedings of the 22nd annual international ACM SIGIR conference*, pp. 214-221.

[14] Ponte, J., and Croft, W.B. (1997). Text segmentation by topic. In *Proceedings of the 1st European conference on research and advanced technology for digital libraries* (pp. 113-125).

[15] Ponte, J., and Croft, W.B. (1998). A language modelling approach to information retrieval. In *Proceedings of the 21st annual international ACM-SIGIR conference on research and development in information retrieval* (pp.275-281), New York: ACM.

[16] Rosenfeld, R. (2000). Two decades of statistical language modelling: where do we go from here? In *Proceedings of the IEEE*, 88(8), 2000.

[17] Salton, G., Allan, J., & Buckley, C. (1993). Approaches to passage retrieval in full text information systems. In R. Korfhage, E. Rasmussen, & P. Willet (Eds.), *Proceedings of the 16th annual international ACM-SIGIR conference on research and development in information retrieval*, Pittsburgh, PA (pp.49-58), New York: ACM.

[18] Salton, G., Allan, J., and Singhal, A.K. (1996). Automatic text decomposition and structuring. *Information Processing and Management*, 32(2), 127-138.

[19] Song, F., & Croft, W.B. (1999). A general language model for information retrieval. In *Proceedings of the 22nd annual international ACM-SIGIR conference on research and development in information retrieval* (pp.279-280), New York: ACM.

[20] Turtle, H.R. (1990). Inference networks for document retrieval. Ph. D. thesis, University of Massachusetts, Amherst.

[21] Zhai, C. and Lafferty, J. (2001). A study of smoothing methods for language models applied to ad hoc information retrieval. In W.B. Croft, D.J. Harper, D.H. Kraft, & J. Zobel (Eds.), *Proceedings of the 24th annual international ACM-SIGIR conference on research and development in information retrieval*, New Orleans, Louisiana (pp. 334-342), New York: ACM.

[22] Zobel, J., Moffat, A., Wilkinson, R., and Sacks-Davis, R. (1995). Efficient retrieval of partial documents. *Information Processing and Management*, 31(3), 361-377.

Table 7. Results for passage retrieval using relevance model with half-overlapped windows. % chg is compared to full-length document retrieval.

		RM with Half-Overlapped Windows						
		Doc	**Psg-50**	**% chg**	**Psg-150**	**% chg**	**Psg-350**	**% chg**
FR-12	Method R1	0.1486	0.3177	+113.8	0.2789	+87.7	0.3071	+106.7
	Method R2		0.2702	+81.8	0.2314	+55.7	0.2596	+74.7
	Method R3		0.1501	+1.0	0.1960	+31.9	0.1960	+31.9
AP	Method R1	0.2754	0.2462	-10.6	0.2681	-2.7	0.2658	-3.5
	Method R2		0.2458	-10.8	0.2680	-2.7	0.2657	-3.5
	Method R3		0.2756	+0.1	0.2755	0.0	0.2755	0.0
TREC-45	Method R1	0.2267	-	-	0.2305	+1.7	0.2208	-2.6
	Method R2		-	-	0.2307	+1.8	0.2210	-2.5
	Method R3		-	-	0.2265	-0.1	0.2205	-2.7

Table 8. Experiments with arbitrary passages. AP collection, queries 51-150. % chg is based on full-length document retrieval.

	Passage Size	**LM (dir-1000)**	**% chg**	**LM (lin-0.5)**	**% chg**	**RM (Method R2)**	**% chg**
Fixed Length	50 words	0.1750	-20.0	0.1658	-18.8	0.2468	-10.4
	150 words	0.2021	-7.6	0.1658	-18.8	0.2678	-2.8
	350 words	0.2079	-4.9	0.1556	-23.8	0.2703	-1.9
Variable Length	-	0.2072	-5.3	0.1695	-17.0	0.2759	+0.2

Table 9. Comparison between LM, RM, and INQUERY, using half-overlapped windows. FR-12, AP, and TREC-45 collections.

		INQUERY	**LM (dir-1000)**	**LM (lin-0.5)**	**RM (Method R1)**	**RM (Method R2)**	**RM (Method R3)**
FR-12	**Document**	0.2289	0.2875	0.2204	0.1486	0.1486	0.1486
	Psg-50	0.2689	0.3065	0.3075	0.3177	0.2702	0.1501
	Psg-150	0.3156	0.2603	0.2751	0.2789	0.2314	0.1960
	Psg-350	0.3191	0.2894	0.2684	0.3071	0.2596	0.1960
AP	**Document**	0.2279	0.2187	0.2043	0.2754	0.2754	0.2754
	Psg-50	0.2036	0.1795	0.1661	0.2462	0.2458	0.2756
	Psg-150	0.2213	0.2067	0.1765	0.2681	0.2680	0.2755
	Psg-350	0.2302	0.2159	0.1823	0.2658	0.2657	0.2755
TREC-45	**Document**	0.1809	0.2011	0.1949	0.2267	0.2267	0.2267
	Psg-150	0.1811	0.1903	0.1781	0.2305	0.2307	0.2265
	Psg-350	0.1828	0.1977	0.1884	0.2208	0.2210	0.2205