Minimal Span Weighting Retrieval for Question Answering

Christof Monz
Institute for Advanced Computer Studies (UMIACS)
University of Maryland
College Park, MD 20742, USA
christof@umiacs.umd.edu

ABSTRACT

Current question answering systems rely on document retrieval as a means of providing documents which are likely to contain an answer to a user's question. Recent research has shown that taking into account the proximity between question terms is helpful in determining whether a document contains an answer to a question. In this paper, we propose a novel proximity-based approach to document retrieval, which combines full-document retrieval with proximity information. Experimental results show that it leads to significant improvements when compared to full document retrieval. Our approach also proves to be useful for extracting short text segments from a document, which contain an answer to the question. This allows answer selection to be focused on smaller segments instead of full documents, and experimental results confirm that it leads to improvements in an existing question answering system.

Categories and Subject Descriptors

H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.4 [**H.3.4 Systems and Software**]: Question-answering (fact retrieval) systems

1. INTRODUCTION

One of the reasons passage-based retrieval is widely used as a pre-fetch in current question answering systems, is the intuition that the answers to most questions can be found in rather short text segments, occupying only a sentence or two. Of course, this depends on the type of question, as some types, e.g., procedural questions such as *How do I make spaghetti alla carbonara?*, require more extensive answers. The fact that most answers are expressed rather locally in a document has two consequences for retrieval as a pre-fetch to a question answering system. First, the retrieval method should take into account the proximity between query terms and rank documents where query terms occur close to each other higher than documents where this is not the case. Second, the retrieval method should return segments of the document which exhibit a high proximity between the query terms instead of full documents.

Both requirements are met by passage-based retrieval. However, the experiments discussed in the literature do not show significant improvements of passage-based retrieval over full-document retrieval when used as a pre-fetch to a QA system, [4, 10, 11, 13]. On the contrary, in most cases it lead to a significant decrease in performance. Although we are reluctant to say that passage-based retrieval is indeed harmful in the context of question answering, it can be concluded that the parameters controlling passage-based retrieval, such as passage size, degree of overlap between passages, fixed length vs. variable length, etc., have to be carefully chosen, and might be highly collection and query dependent.

An alternative to passage-based retrieval that meets the two requirements mentioned above, is proximity-based retrieval. Other than for passage-based retrieval, parameters such as passage size, degree of overlap between passages, etc., do not need to be fixed. In passage-based retrieval, the proximity between a number of terms is determined by checking whether they occur in the same passage, which depends on the passage size. In proximity-based retrieval, proximity is expressed as the distance between terms, i.e., the number of words occurring between them. Defining the proximity between two terms is trivial, but several approaches are possible if more than two words are involved.

The remainder of this paper is organized as follows: The next section provides an overview on previous approaches to proximity-based retrieval for question answering. Section 3 introduces our novel retrieval approach based on minimal spans. Section 4 provides the details of the experimental setting and the evaluation itself. Section 5 discusses the integration of minimal span weighting into an existing QA system and its comparative evaluation. In Section 6, we draw some conclusions and give an outlook on future research.

2. RELATED WORK

Numerous approaches to proximity-based retrieval have been proposed in the literature. The intuition that the proximity between query terms in a document affects relevance dates back to 1958, to the work of Luhn [9].

Clark et al. [3, 2] use proximity-based retrieval as a method for pre-fetching and excerpt extraction in question answering. If a document contains n query terms, [3] consider all spans that contain $m \leq n$ terms. Since proximity-based retrieval is used to identify text excerpts that are likely to contain answer to a question, the retrieval system returns a ranked list of spans instead of documents.

Kwok et al. [8] use proximity-based retrieval as a pre-fetch in their question answering system, but applied it only if all terms from the question did occur in a document, i.e., partial spans where not considered. This very strong restriction requires the retrieval query to be formulated very carefully.

Although proximity-based retrieval is used in many question answering systems there are few experimental evaluations of its effectiveness as a pre-fetch for question answering. Cormack et al. [5] use a proximity-based retrieval system as the question answering system for their participation in the TREC-8 250-byte task. They did not apply any question analysis nor answer selection. Nevertheless, their top five responses contained a correct answer for 63% of the questions. Unfortunately, the reliability of the TREC-8 data set is questionable, because the questions were mostly back-formulations of sentences in the document collection which contained a correct answer [20]. This resulted in a large word overlap between questions and answer sentences, which distorts many findings based on this data set.

Tellex et al. [17] compare the impact of eight passage-based and locality-based retrieval strategies that were used by TREC participants. The different approaches are compared with respect to the overall performance of a version of the MIT question answering system. They show that the choice of the retrieval approach that is used for pre-fetching does have a significant impact on the overall performance of a question answering system. In their evaluation, algorithms that take the proximity between terms into account perform best.

Clarke and Terra [4] use their own locality-based retrieval algorithm and integrate it into a passage-based retrieval and a full-document retrieval using an implementation of the Okapi retrieval system [14]. Their results indicate that full-document retrieval returns more documents that contain a correct answer, but that passage-based retrieval might still be useful in the context of question answering as it returns shorter excerpts that might ease the process of identifying an actual answer.

3. MINIMAL SPAN WEIGHTING

In this section, we introduce a new proximity-based approach to document retrieval, which is based on the minimal size of a text excerpt that covers all terms that are common between the document and the query, the number of common terms vs. the number of query terms, and the global similarity between the document and the query. The advantage of this approach over previous approaches to proximity-based retrieval, lies in the number of aspects that are taken into account, namely full-document similarity, ratio of matching terms, and the proximity of matching terms, and the parameterized way in which the different aspects are combined to compute the final document similarity score.

3.1 Definition of Minimal Span Weighting

Minimal span weighting takes the positions of matching terms into account, but does so in a more flexible way than passage-based retrieval. Intuitively, a minimal matching span is the smallest text excerpt from a document that contains all terms which occur in the query and the document. More formally:

Definition 1. (*Matching span*) Given a query q and a document d, where the function $\text{term\_at\_pos}_d(p)$ returns the term occurring at position p in d. A *matching span* (ms) is a set of positions that contains at least one position of each matching term, i.e. $\bigcup_{p \in \text{ms}} \text{term\_at\_pos}_d(p) = q \cap d$. ■

Definition 2. (*Minimal matching span*) Given a matching span ms, let b_d (the beginning of the excerpt) be the minimal value in ms, i.e., $b_d = \min(\text{ms})$, and e_d (the end of the excerpt) be the maximal value in ms, i.e., $e_d = \max(\text{ms})$. A matching span ms is a *minimal matching span* (mms) if there is no other matching span ms$'$ with $b'_d = \min(\text{ms}')$, $e'_d = \max(\text{ms}')$, such that $b_d \neq b'_d$ or $e_d \neq e'_d$, and $b_d \leq b'_d \leq e'_d \leq e_d$. ■

The next step is to use minimal matching spans to compute the similarity between a query and a document. Minimal span weighting depends on three factors.

1. *document similarity*: The document similarity is computed using the Lnu.ltc weighting scheme, see [1], for the whole document; i.e., positional information is not taken into account. Similarity scores are normalized with respect to the maximal similarity score for a query.
2. *span size ratio*: The span size ratio is the number of unique matching terms in the span over the total number of tokens in the span.
3. *matching term ratio*: The matching term ratio is the number of unique matching terms over the number of unique terms in the query, after stop word removal.

The msw score is the sum of two weighted components: The normalized original retrieval status value (RSV), which measures *global similarity* and the spanning factor which measures *local similarity*. Given a query q, the original retrieval status values are normalized with respect to the highest retrieval status value for that query:

$$\text{RSV}_n(q,d) = \frac{\text{RSV}(q,d)}{\max_d \text{RSV}(q,d)}$$

The spanning factor itself is the product of two components: The span size ratio, which is weighted by α, and the matching term ratio, which is weighted by β. Global and local similarity are weight by λ. The optimal values of the three variables λ, α, and β were determined empirically, leading to the following instantiations: $\lambda = 0.4$, $\alpha = 1/8$, and $\beta = 1$. Parameter estimation was done using the TREC-9 data collection only, but it turned out to be the best parameter setting for all collections.

The final retrieval status value (RSV') based on minimal span weighting is defined as follows, where $|\cdot|$ is the number of elements in a set:

Definition 3. (*Minimal span weighting*) If $|q \cap d| > 1$ (that is, if the document and the query have more than one term in common), then

$$\begin{aligned}\text{RSV'}(q,d) = \quad & \lambda \, \text{RSV}_n(q,d) \\ & + (1-\lambda) \left(\frac{|q \cap d|}{1 + \max(mms) - \min(mms)} \right)^{\alpha} \\ & \cdot \left(\frac{|q \cap d|}{|q|} \right)^{\beta}\end{aligned}$$

If $|q \cap d| = 1$ then $\text{RSV'}(q,d) = \text{RSV}_n(q,d)$. ■

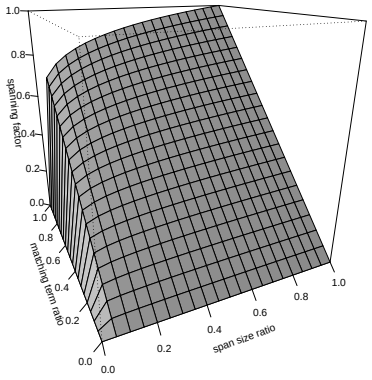

Figure 1: 3D plot of the spanning factor function, which is (span size ratio)$^{1/8}$· (matching term ratio).

Note that minimal span weighting only exploits minimal matching spans for documents containing more than one matching term, as proximity between terms is not defined for documents containing only one matching term. The retrieval status value for documents containing only one matching term is equal to the documents normalized global retrieval status value. Here, we consider only a single minimal matching span per document. It is possible that a document contains several identical minimal matching spans, but that does not affect the spanning factor as it is the same for all minimal spans in a document.

We also ran a number of experiments using alternative weighting schemes to the presented in Defintion 3, but none of them outperformed the one above.

At this point it might be helpful to further illustrate the definition by considering the following example question:

(1) Who is Tom Cruise married to? (topic id: 1395)

After stop word removal and applying morphological normalization, the query q={*cruise*, *marri*, *tom*}. Assume that there is a document d with terms matching at the following positions: $\text{pos}_d(cruise) = \{20, 35, 70\}$, $\text{pos}_d(marri) = \{38, 80\}$, and $\text{pos}_d(tom) = \emptyset$. Then, the minimal matching span (mms) = {35, 38}, the span size ratio is $2/(1+38-35) = 0.5$, and the matching term ratio is 2/3. Taking the latter two and the proper instantiations of α and β, the spanning factor is $0.5^{1/8} \cdot 2/3 = 0.611$. If the global (normalized) similarity between q and d is n ($0 < n \leq 1$), for instance $n = 0.8$, and $\lambda = 0.4$, the final msw-score for q and d (RSV'(q, d)) is $0.4 \cdot 0.8 + 0.6 \cdot 0.611 = 0.6866$.

To illustrate the behavior of the spanning factor, Figure 1 plots the values of the spanning factor for all possible combinations of span size ratio and matching term ratio. One can see that, initially, the spanning factor decreases slowly as the span size ratio decreases, but then it drops sharply as the span size ratio falls below a certain threshold, approx. 0.05. Along the other dimension, the spanning factor decreases linearly with the matching term ratio.

4. EVALUATING RETRIEVAL

4.1 Expermental Set-Up

We used the TREC-9, TREC-10, and TREC-11 data sets consisting of 500 questions each with 978,952 documents for TREC-9 and TREC-10 from the TIPSTER/TREC distribution and 1,033,461 documents for TREC-11 from the AQUAINT distribution. At TREC-9 and TREC-10, participants were required to return up to five answer-document-id pairs for each question, where the answer can be any text string containing maximally 50 characters, and the document-id refers to the document from which the answer was extracted. At TREC-11, participants were required to return one answer-document-id pair for each question, where the answer had to be the exact answer.

In addition, we used the judgment files which were provided by NIST as a result of their evaluation.[1] A judgment file, which is comparable to a qrel file in ad-hoc retrieval, indicates for each submitted answer-document-id pair, whether the answer is correct and whether the document supports, i.e., justifies, the answer. The justifying documents form the set of relevant documents against which we evaluate the different document retrieval approaches for pre-fetching. If none of the participants returned a supported answer, that topic was discarded from our evaluation. This also included questions that did not have an answer in the collection, which can be the case since TREC-10.

The final evaluation sets consist of 480, 433, and 455 topics for TREC-9, TREC-10, and TREC-11, respectively. The original question set for TREC-9 actually contained 693 questions where 193 questions were syntactic variants of 54 of the remaining 500 questions. Here, we did not use the variants, but if a relevant document for a variant was included in the judgment file, it was added to the set of relevant documents of the original question. Variants were removed to avoid repetition of topics, which could bias the overall evaluation. We also included 10 topics of the TREC-11 question set, where NIST assessors 'coincidentally' recognized a document containing an answer.

In the remainder, we use the following two evaluation measures, where R_q is the set of documents that contain an answer to question q.

p@n: $|\{d \in R_q \mid \text{rank}(d) \leq n\}|/n$. The number of found relevant documents up to rank n divided by n.

p@n measures the precision of a given retrieval system at rank n. Note that the internal order of the ranks up to rank n does not affect p@n. Often, it is convenient to neglect the exact precision and simply measure whether a system returns a relevant document:

a@n: 1 if $|\{d \in R_q \mid \text{rank}(d) \leq n\}| \geq 1$, and 0 otherwise.

To determine whether the observed differences between two retrieval approaches are statistically significant and not just caused by chance, we used the bootstrap method, a powerful non-parametric inference test [7]. The method has previously been applied to retrieval evaluation by, e.g., Savoy

[1]The judgment files are available from the TREC web site: `http://trec.nist.gov`.

[15] and Wilbur [23]. The basic idea of the bootstrap is a simulation of the underlying distribution by randomly drawing (with replacement) a large number of samples of size N from the original sample of N observations. These new samples are called *bootstrap samples*; we set the number of bootstrap samples to 2,000 as using the standard size of 1,000 did not always result in a normal distribution of the bootstrap sample. The mean and the standard error of the bootstrap samples allow computation of a confidence interval for different levels of confidence (typically 0.95 and higher). We compare two retrieval methods a and b by one-tailed significance testing. If the left limit of the confidence interval is greater than zero, we reject the null hypothesis, stating that method b is not better than a, and conclude that the improvement of b over a is statistically significant, for a given confidence level. Analogously, if the right limit of the confidence interval is less than zero, one concludes that method b performs significantly worse than a.

In the remainder, we indicate improvements at a confidence level of 95% with "$^{\triangle}$" and at a confidence level of 99% with "$^{\blacktriangle}$". Analogously, decreases in performance at a confidence level of 95% are marked with "$^{\triangledown}$" and at a confidence level of 99% with "$^{\blacktriangledown}$". No markup is used if neither an increase nor a decrease in performance is significant at either of the 95% or 99% confidence levels.

4.2 Experimental Results

Table 1 provides more details on the differences in performance between minimal span weighting and the Lnu.ltc baseline. The msw approach significantly improves retrieval for all three collections compared to the baseline. Improvements are especially high at lower cut-offs. Taking a closer look at the precision at a given cut-off level n (p@n) reveals even higher improvements, see Table 2. The drop in absolute precision at n for the TREC-11 data set (as compared to the TREC-9 and TREC-10 data sets), at all cut-off levels, is probably due to the fact that the questions were more difficult than questions of the TREC-9 and TREC-10 data sets, and, which is more likely, to the smaller average number of relevant documents. All improvements of using minimal span weighting instead of Lnu.ltc weighting are significant at a confidence level of 99%.

4.3 Individual Query Performance

Despite the significant improvements of the minimal span weighting scheme over Lnu.ltc weighting, it does not improve for all queries. Figure 2 shows the histograms for the respective TREC collections, measuring the absolute difference in average precision between the Lnu.ltc baseline and minimal span weighting for each query.

All three data sets exhibit a similar distribution of increases and decreases in effectiveness of minimal span weighting for individual queries. In most cases, the retrieval performances of the individual queries are affected positively, but for some queries msw performs slightly worse, and for a few queries performance drops dramatically. In order to see whether the impact of minimal span weighting depends on some characteristic of the query, we looked at the individual queries. If one could find such a characteristic, the λ factor in the msw scheme (see definition 3) could be easily instantiated in such a way that the effect of span matching is controlled appropriately. Unfortunately, it is hard to find such a characteristic, and it might be possible that such a trait simply does not exist. Research on predicting the hardness of an information need [22], which is loosely related to the current problem, has shown the difficulties in finding features in the topic that predict the behavior of a retrieval system.

Here, we only looked at one factor that could affect the performance of minimal span weighting, viz. query length, assuming that the longer the query is, the harder it is to find a short span. To compute the correlation between query length and average precision, we used Kendall's τ measure, which resulted in a correlation of -0.056, strongly suggesting that query length and average precision are not related.

Just looking at the questions and their respective average precisions unfortunately did not suggest any prevalent characteristics of the question that might be indicative for predicting the retrieval system's performance. On the other hand, there are many more aspects of a question than its length that might play a role for the effectiveness of minimal span weighting, but a thorough investigation of these aspects is a very involved enterprise and remains an issue for future research.

5. SPANS AND ANSWERHOOD

Passage-based retrieval is widely used as a pre-fetch for question answering for two reasons. First, the answer to a question is normally expressed very locally, and using passages instead of whole documents takes the aspect of locality better into account. Whether passage-based retrieval is indeed more effective than document-based retrieval remains questionable as earlier experimental results did not show any improvements. Second, returning passages instead of documents, allows later components of the QA system, such as answer extraction, to work on smaller and more focused text excerpts, thus reducing computational costs.

Similar to passage-based retrieval, minimal span weighting computes a text excerpt (a minimal matching span) which is used to re-weight the document it was extracted from. In addition, it is also possible to return the minimal matching span instead of the document and have later components process the minimal span. The question is how useful is the minimal matching span for answer extraction, or to put it differently, how often does it contain a correct answer to a question?

Definition 2 of a minimal matching span, simply uses the positions of terms in a document, neglecting any kind of textual structure, such as sentence or paragraph boundaries. When using minimal span weighting for document retrieval this is indeed irrelevant, but when using the minimal matching spans for further processing one would like to have them obey at least sentence boundaries, which increases readability and enables them to be analyzed by a full parser. Additionally, it may happen that the answer is just to the left or right boundary of the minimal matching span, and the returned span would not include the answer, although the answer is in the same sentence as one of the span boundaries. In order to accomplish this, we extend each minimal matching span such that the left boundary is moved to the first word of the sentence in which it occurred, and the right

	TREC-9			TREC-10			TREC-11		
a@n	Lnu.ltc	msw		Lnu.ltc	msw		Lnu.ltc	msw	
a@5	0.700	0.789	(+12.8%)▲	0.649	0.736	(+13.5%)▲	0.523	0.630	(+20.5%)▲
a@10	0.785	0.860	(+9.5%)▲	0.734	0.829	(+12.9%)▲	0.626	0.729	(+16.4%)▲
a@20	0.845	0.918	(+8.6%)▲	0.801	0.873	(+8.9%)▲	0.705	0.800	(+13.4%)▲
a@50	0.914	0.939	(+2.7%)▲	0.875	0.903	(+3.1%)▲	0.795	0.868	(+9.1%)▲

Table 1: Comparison of the a@n scores of msw retrieval runs to baseline runs.

	TREC-9			TREC-10			TREC-11		
p@n	Lnu.ltc	msw		Lnu.ltc	msw		Lnu.ltc	msw	
p@5	0.310	0.377	(+21.5%)▲	0.270	0.322	(+19.1%)▲	0.167	0.226	(+34.8%)▲
p@10	0.238	0.293	(+22.9%)▲	0.212	0.255	(+20.0%)▲	0.123	0.167	(+35.2%)▲
p@20	0.171	0.214	(+25.1%)▲	0.154	0.186	(+20.6%)▲	0.084	0.114	(+35.1%)▲
p@50	0.102	0.124	(+21.9%)▲	0.088	0.105	(+19.1%)▲	0.047	0.060	(+26.3%)▲

Table 2: Comparison of p@n scores of msw retrieval runs to baseline runs.

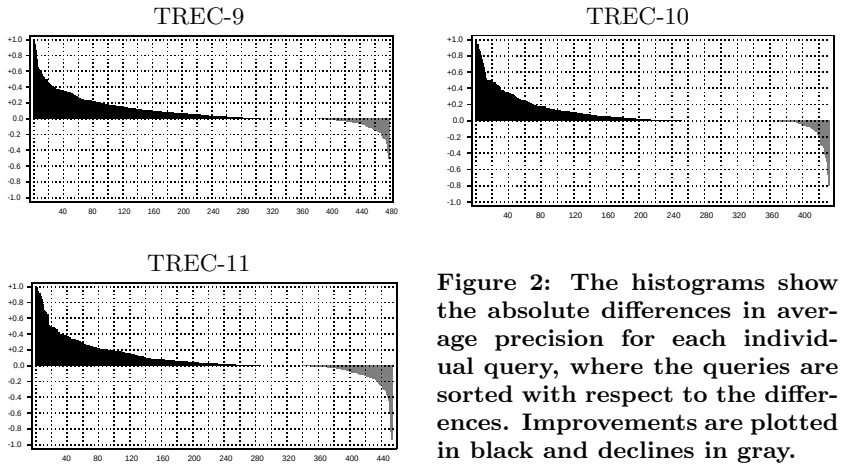

Figure 2: The histograms show the absolute differences in average precision for each individual query, where the queries are sorted with respect to the differences. Improvements are plotted in black and declines in gray.

boundary is moved to the last word of the sentence in which it occurred. Such an extended span is called a minimal sentential span, and it is formally defined as follows:

Definition 4. (*Minimal matching sentential span*) Let F_d be the set of positions of a first words of a sentence in document d, L_d be the set of positions of a last words of a sentence in document d and $mms_{q,d}$ is the minimal matching span in d for a query q, with the left boundary $b = \min(mms_{q,d})$, and right boundary $e = \max(mms_{q,d})$. The *minimal matching sentential span* is $(mms_{q,d} - \{b, e\}) \cup \{b', e'\}$, where $b' \in F_d$ and there is no $b'' \in F_d$ such that $b'' \leq b$ and $b'' > b'$, and where $e' \in L_d$ and there is no $e'' \in L_d$ such that $e'' \geq e$ and $e'' < e'$. ■

In practice, the extraction of a minimal matching sentential span also depends on the accuracy of the identification of sentence boundaries. Here, we use our own sentence splitter, which uses the TreeTagger [16] part-of-speech tagger to annotate the document. TreeTagger's tag set includes a sentence boundary tag, but in some cases sentence boundary tagging is incorrect and a number of heuristics have been applied to correct this.

Returning to the use of minimal matching spans in the context of question answering, we reconsider the experiments discussed above, where minimal matching spans were used to rank documents, see section 4. For each of the top documents we know what the minimal matching span is and given that information, we computed the respective minimal matching sentential span. Before turning to the issue to what extent the minimal matching sentential spans contain answers to questions, their average lengths should be considered, because if the spans tend to be very long, the argument that they allow for a more focused analysis would be severely weakened. Table 3 shows the average and median number of words and bytes (characters) of the minimal matching sentential spans for different cut-off levels.

The first thing that jumps out is the large difference between average and median lengths; the former being roughly twice as large as the latter. This is due to a number of outliers with extremely long spans. Nevertheless, both average and median lengths are rather small and hence do allow for a focused analysis. Note that the numbers in Table 3 roughly correspond to an average span length of 2–3 sentences and a median span length of 1–2 sentences.

	TREC-9				TREC-10				TREC-11			
	words		bytes		words		bytes		words		bytes	
	avg	med	avg	med	avg	med	avg	med	avg	med	avg	med
@5	65	36	396	225	56	34	345	215	77	39	467	236
@10	71	37	427	230	59	35	362	220	79	39	480	240
@20	72	38	435	233	62	36	378	221	84	40	506	247
@50	77	39	464	238	69	36	420	223	93	41	561	254

Table 3: The average (avg) and median (med) minimal matching sentential span lengths for the different TREC collections at cut-off levels 5, 10, 20, and 50, counted in words and bytes (characters).

The next question is to check how often the minimal matching sentential span does contain a correct answer. In order to evaluate this, one has to look at each span and decide whether this is the case. Obviously, this is a very laborious process and practically almost impossible, if done manually. One way to automatize this is to collect the known correct answers and simply apply pattern matching to see whether the minimal matching sentential span does match one of the correct answers. NIST provided a set of regular expressions that characterize the correct answers for the TREC-9 data set and Ken Litkowski did the same for the TREC-10 and TREC-11 data sets.[2]

In order to evaluate minimal matching sentential span extraction, two aspects have to be considered: First, does the span originate from a relevant document, and second, does the span contain a correct answer? The set of relevant documents for a question is defined as in section 3, and the set of spans containing a correct answer is identified by pattern matching. Given a cut-off level of n ($n \in \{5, 10, 20, 50\}$), R^+ (R^-) refers to the total number of relevant (non-relevant) documents for all questions, and S^+ (S^-) refers to the total number of spans containing (not containing) a correct answer. R^+S^+/R^+ is the number of relevant documents where the extracted span contains a correct answers divided by the total number of relevant documents. R^+S^+/R^+ indicates the ability of the span extraction to identify a text excerpt containing a correct answer, given a document that is known to contain a correct answer. On the other hand, R^-S^+/R^- is the ratio of spans from non-relevant documents that contain a correct answer. Table 4 shows the R^+S^+/R^+ and R^-S^+/R^- numbers for the different TREC collection at different cut-off levels. All in all, the minimal matching sentential span is a relatively good starting point for answer extraction, because it contains the correct answer in 64.1–71.8% of the cases, but of course we hasten to add that this is still far from perfect. One can also see that in 5.9–9.3% of the cases, a span from a document which was not judged relevant does match a correct answer, but this number is hard to interpret: It could be that a document does contain a correct answer, but was simply not judged during the TREC evaluations, but it could also be the case that a document contains a string matching an answer without allowing one to conclude that it is indeed an answer to the question.

In the discussion above, we evaluated to what extent the minimal matching sentential spans contain a correct answer with respect to all relevant documents. The next issue is to see for how many of the questions the spans allow an answer selection procedure to find at least one correct answer. Assuming that answer selection is perfect, i.e., if a minimal matching sentential span contains a correct answer, then the selection procedure will find it, it allows one to determine an upper bound for the usefulness of the spans for question answering. Table 5 gives the percentages of questions where at least one minimal matching sentential span, which was extracted from a relevant document, contains a correct answer. In addition to the percentages also the mean reciprocal rank (MRR) is given. The reciprocal rank of a question is 1 divided by the highest rank at which a span from a relevant document contained a correct answer, and the MRR is the average of the questions' individual reciprocal ranks, cf. [18]. Here, we impose another constraint on the spans, namely that they are not longer than 250 or 500 bytes (characters). We restrict the span lengths, because the role of a minimal matching sentential span is to function as a 'hotspot' for answer selection which requires more expansive analysis, including parsing, named entity extraction, etc. If the span size is large, the property of being a 'hotspot' is lost. Although the numbers in Table 5 show that a question answering system that would be based purely on minimal matching sentential spans is far from perfect, these results are roughly in the same ballpark as most of the better performing current QA systems, see [19, 21].

5.1 Minimal Span Weighting within Tequesta

As we have seen in section 4, minimal span weighting greatly outperforms retrieval based on the Lnu.ltc weighting scheme. Now, the question is to what extent a QA system benefits from the improved retrieval component. Here, we use our own Tequesta QA system [12]. In order to focus on the impact of the similarity weighting scheme itself, and not on the text units that are returned by the retrieval component, we had both approaches return the same unit, viz. the minimal matching span, see definition 2. Although minimal matching spans were also computed for the Lnu.ltc weighting scheme, they were not used for computing document similarity. The Lnu.ltc weighting scheme is just like the minimal span weighting scheme, see definition 3, where only the global similarity is used to compute the retrieval status value, i.e., λ is set to 1.

Table 6 shows the percentages of questions that were correctly answered at the respective top-5 ranks, and the MRR score, for the three TREC data sets. As one can see, using minimal span weighting instead of Lnu.ltc weighting also has a substantial positive effect on the overall performance of the Tequesta system. For all three data sets, the im-

[2]The sets of answer patterns are available from the TREC web site: `http://trec.nist.gov`.

	TREC-9		TREC-10		TREC-11	
	R^+S^+/R^+	R^-S^+/R^-	R^+S^+/R^+	R^-S^+/R^-	R^+S^+/R^+	R^-S^+/R^-
@5	70.0%	5.9%	65.3%	6.4%	64.1%	9.3%
@10	68.5%	6.5%	65.6%	8.1%	67.4%	9.1%
@20	70.2%	6.8%	68.0%	8.3%	67.1%	8.9%
@50	71.8%	7.5%	68.9%	8.7%	68.8%	7.9%

Table 4: Percentage of minimal matching sentential spans from relevant documents (R^+S^+/R^+) and non-relevant documents (R^-S^+/R^-) containing a correct answer for different TREC data sets measured at cut-off levels 5, 10, 20, and 50.

	TREC-9		TREC-10		TREC-11	
	250 bytes	500 bytes	250 bytes	500 bytes	250 bytes	500 bytes
@5	52.1%	60.0%	46.9%	53.1%	32.0%	38.1%
@10	60.9%	67.6%	56.6%	63.5%	40.1%	48.2%
@20	65.3%	74.1%	61.7%	68.6%	45.7%	53.6%
@50	68.2%	77.5%	64.0%	72.3%	46.9%	57.7%
MRR	0.39	0.44	0.34	0.37	0.23	0.27

Table 5: Percentage of questions, where at least one minimal matching sentential spans (not longer than 250/500 bytes) stems from a relevant document and contains a correct answer measured at cut-off levels 5, 10, 20, and 50.

	TREC-9			TREC-10			TREC-11		
rank	Lnu.ltc	msw		Lnu.ltc	msw		Lnu.ltc	msw	
1	16.7%	21.0%	(+25.8%)	17.1%	21.5%	(+25.7%)	16.2%	18.2%	(+12.4%)
2	20.6%	26.9%	(+30.6%)	21.3%	26.3%	(+23.5%)	20.3%	23.2%	(+14.3%)
3	22.9%	29.0%	(+26.6%)	23.1%	28.6%	(+23.8%)	23.2%	26.1%	(+12.5%)
4	25.1%	30.0%	(+19.5%)	24.5%	30.0%	(+22.5%)	25.9%	28.4%	(+9.7%)
5	26.5%	31.4%	(+18.5%)	25.4%	31.0%	(+22.1%)	28.2%	30.6%	(+8.5%)
MRR	0.203	0.252	(+24.1%)$^{\blacktriangle}$	0.203	0.252	(+24.1%)$^{\blacktriangle}$	0.204	0.227	(+11.3%)$^{\triangle}$

Table 6: Lenient evaluation of Tequesta using Lnu.ltc vs. msw retrieval

provements are statistically significant, with a confidence of 99% for TREC-9 and TREC-10, and a confidence of 95% for TREC-11.

6. CONCLUSIONS

Considering proximity between query terms when retrieving documents requires the indexation of positional information, which increases the size of the inverted index and results in a slight overhead in efficiency. On the other hand, the results in section 4 indicate that proximity-based retrieval exhibits significant improvements in effectiveness compared to regular Lnu.ltc retrieval.

Our minimal span weighting approach also allows the retrieval system to identify small text segments that can function as starting points for further processing steps, such as answer selection. We have shown that the minimal matching spans contain a correct answer in 64.1–71.8% of the cases, where the document is known to contain a correct answer.

We have also seen that the overall performance of the Tequesta question answering system benefits significantly from using minimal span weighting instead of Lnu.ltc weighting. Hence the effectiveness of a retrieval system does have a strong impact on the performance of the whole process of question answering.

Summing up, proximity-based retrieval does significantly improve document retrieval as a pre-fetch to a question answering system, and it is useful for finding short text segments in a document that are likely to contain a correct answer.

In this article, we did not address the issue to what extent minimal span weighting has an impact on the retrieval performance in tasks other than document retrieval as a pre-fetch to a question answering system. One might suspect that it should have a positive impact on any retrieval task where the information need is rather specific, or where early high precision is required. During our experimentation we have also applied minimal span weighting to the TREC-11 named page finding task, where a retrieval system is supposed to find a unique web page, given a topic which describes it by name, cf. [6]. Using minimal span weighting for this task resulted in an MRR score of 0.513, whereas the Lnu.ltc baseline MRR score was 0.359, which is an improvement of 43%, and is significant at a 99% confidence level. Applying minimal span weighting to other tasks and evaluating its effectiveness remains to be done.

Acknowledgments

This research was supported in part by the Netherlands Organization for Scientific Research (NWO) under project numbers 612-13-001 and 220-80-001.

7. REFERENCES

[1] C. Buckley, A. Singhal, and M. Mitra. New retrieval approaches using SMART: TREC 4. In D. Harman, editor, *Proceedings of the Fourth Text REtrieval Conference (TREC-4)*, pages 25–48. NIST Special Publication 500-236, 1995.

[2] C. Clarke, G. Cormack, G. Kemkes, M. Laszlo, T. Lynam, E. Terra, and P. Tilke. Statistical selection of exact answers (MultiText experiments for TREC 2002). In *Notebook of the 11th Text REtrieval Conference (TREC 2002)*, pages 162–170. NIST Publication, 2002.

[3] C. Clarke, G. Cormack, D. Kisman, and T. Lynam. Question answering by passage selection (MultiText experiments for TREC-9). In E. Voorhees and D. Harman, editors, *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 673–683. NIST Special Publication 500-249, 2000.

[4] C. Clarke and E. Terra. 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*, pages 427–428, 2003.

[5] G. Cormack, C. Clarke, C. Palmer, and D. Kisman. Fast automatic passage ranking. In E. Voorhees and D. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 735–742. NIST Special Publication 500-246, 1999.

[6] N. Craswell and D. Hawking. Overview of the TREC 2002 web track. In *Notebook of the 11th Text REtrieval Conference (TREC 2002)*, pages 248–257. NIST Publication, 2002.

[7] B. Efron. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7(1):1–26, 1979.

[8] K. Kwok, L. Grunfeld, N. Dinsl, and M. Chan. TREC-9 cross language, web and question answering track experiments using PIRCS. In E. Voorhees and D. Harman, editors, *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 419–429. NIST Special Publication 500-249, 2000.

[9] H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165, 1958.

[10] C. Monz. Document retrieval in the context of question answering. In F. Sebastiani, editor, *Proceedings of the 25th European Conference on Information Retrieval Research (ECIR-03)*, Lecture Notes in Computer Science 2633, pages 571–579. Springer, 2003.

[11] C. Monz. *From Document Retrieval to Question Answering*. PhD thesis, University of Amsterdam, 2003.

[12] C. Monz and M. de Rijke. Tequesta: The University of Amsterdam's textual question answering system. In *Proceedings of Tenth Text Retrieval Conference (TREC-10)*, pages 513–522, 2001.

[13] I. Roberts and R. Gaizauskas. Evaluating passage retrieval approaches for question answering. In *Proceedings 26th European Conference on IR Research (ECIR 2004)*, pages 72–84. Springer, 2004.

[14] S. Robertson and S. Walker. Okapi/Keenbow at TREC-8. In E. Voorhees and D. Harman, editors, *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pages 151–162. NIST Special Publication 500-246, 1999.

[15] J. Savoy. Statistical inference in retrieval effectiveness evaluation. *Information Processing and Management*, 33(4):495–512, 1997.

[16] H. Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, 1994.

[17] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. 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*, pages 41–47, 2003.

[18] E. Voorhees. Overview of the TREC-9 question answering track. In E. Voorhees and D. Harman, editors, *Proceedings of the Ninth Text REtrieval Conference (TREC-9)*, pages 71–80. NIST Special Publication 500-249, 2000.

[19] E. Voorhees. Overview of the TREC 2001 question answering track. In E. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, pages 42–51. NIST Special Publication 500-250, 2001.

[20] E. Voorhees. The TREC question answering track. *Natural Language Engineering*, 7(4):361–378, 2001.

[21] E. Voorhees. Overview of the TREC 2002 question answering track. In *Notebook of the 11th Text REtrieval Conference (TREC 2002)*, pages 115–123. NIST, 2002.

[22] E. Voorhees and D. Harman. Overview of the sixth text retrieval conference (trec-6). In E. Voorhees and D. Harman, editors, *Proceedings of the Sixth Text REtrieval Conference (TREC 6)*, pages 1–24. NIST Special Publication 500-240, 1997.

[23] J. Wilbur. Non-parametric significance tests of retrieval performance comparisons. *Journal of Information Science*, 20(4):270–284, 1994.