Simple is Best: Experiments with Different Document Segmentation Strategies for Passage Retrieval

Jörg Tiedemann
Information Science
University of Groningen
j.tiedemann@rug.nl

Jori Mur
Information Science
University of Groningen
j.mur@rug.nl

Abstract

Passage retrieval is used in QA to filter large document collections in order to find text units relevant for answering given questions. In our QA system we apply standard IR techniques and index-time passaging in the retrieval component. In this paper we investigate several ways of dividing documents into passages. In particular we look at semantically motivated approaches (using coreference chains and discourse clues) compared with simple window-based techniques. We evaluate retrieval performance and the overall QA performance in order to study the impact of the different segmentation approaches. From our experiments we can conclude that the simple techniques using fixed-sized windows clearly outperform the semantically motivated approaches, which indicates that uniformity in size seems to be more important than semantic coherence in our setup.

1 Introduction

Passage retrieval in question answering is different from information retrieval in general. Extracting relevant passages from large document collections is only one step in answering a natural language question. There are two main differences: i) Passage retrieval queries are generated from complete sentences (questions) compared to bag-of-keyword queries usually used in IR. ii) Retrieved passages have to be processed further in order to extract concrete answers to the given question. Hence, the size of the passages retrieved is important and smaller units are preferred. Here, the division of documents into passages is crucial. The textual units have to be big enough to ensure IR works properly and they have to be small enough to enable efficient and accurate QA. In this study we investigate whether semantically motivated passages in the retrieval component lead to better QA performance compared to the use of document retrieval and window-based segmentation approaches.

1.1 Index-time versus Search-time Passaging

In this paper, we experiment with various possibilities of dividing documents into passages *before* indexing them. This is also called *index-time* passaging and refers to a one-step process of retrieving appropriate textual units for subsequent answer extraction modules (Roberts and Gaizauskas, 2004; Greenwood, 2004). This is in contrast to other strategies using a two-step procedure consisting of document retrieval and *search-time* passaging thereafter. Here, we can distinguish between approaches that only return one passage per relevant document (see, for example, (Robertson et al., 1992)) and the ones that allow multiple passages per document (see, for example (Moldovan et al., 2000)). In general, allowing multiple passages per document is preferable for QA as possible answers can be contained at various positions in a document (Roberts and Gaizauskas, 2004). For this, an index-time approach has the advantage that the retrieval of multiple passages per documents is straightforward because all of them compete which each other in the same index using the same metric for ranking.

A comparison between index-time and search-time passaging has been carried out in (Roberts and Gaizauskas, 2004). In their experiments, index-time passaging performs similarly to search-time passaging in terms of coverage and redundancy (measures which have been introduced in the same paper; see section 4.2 for more information). Significant differences between the various approaches can only be observed in redundancy on higher ranks (above 50). However, as we will see later in our experiments (section 4.2), redundancy is not as important as coverage for our QA system . Furthermore, retrieving more than about 40 passages does not produce significant improvements of the QA system anymore but slows down the processing time substantially.

Another argument for our focus on a one-step retrieval procedure can be taken from (Tellex et al., 2003). In this paper, the authors do not actually use any index-time passaging approach but compare various search-time passage retrieval algorithms. However, they obtain a huge performance difference when applying an oracle document retriever (only returning relevant documents in the first retrieval step) instead of a standard IR engine. Compared to this, the differences between the various passage retrieval approaches tested is very small. From this we can conclude that much improvement can be gained by improving the initial retrieval step, which seems to be the bottleneck in the entire process. Unfortunately, the authors do not compare their results with index-time approaches. However, looking at the potential gain in document retrieval and keeping in mind that the performance of index-time and search-time approaches is rather similar (as we have discussed earlier) we believe that the index-time approach is preferable.

1.2 Passages in IR

Certainly, IR performance is effected by changing the size of the units to be indexed. The task in document segmentation for our index-time passaging approach is to find the proper division of documents into text passages which optimize the retrieval in terms of overall QA performance.

The general advantages of passage retrieval over full-text document retrieval has been investigated in various studies, e.g., (Kaszkiel and Zobel, 2001; Callan, 1994; Hearst and Plaunt, 1993; Kaszkiel and Zobel, 1997). Besides the argument of decreasing the search space for subsequent answer extraction modules in QA, passage retrieval also improves standard IR techniques by "normalizing" textual units in terms of size which is especially important in cases where documents come from very diverse sources. IR is based on similarity measures between documents and queries and standard approaches have shortcomings when applying them to documents of various sizes and text types. Often there is a bias for certain types raising problems of discrimination between documents of different lengths and content densities. Passages on the other hand provide convenient units to be returned to the user avoiding such ranking difficulties (Kaszkiel and Zobel, 2001). For IR, passage-level evidence may be incorporated into document retrieval (Callan, 1994; Hearst and Plaunt, 1993) or passages may be used directly as retrieval unit (Kaszkiel and Zobel, 2001; Kaszkiel and Zobel, 1997). For QA only the latter is interesting and will be applied in our experiments.

Passages can be defined in various ways. The most obvious way is to use existing markup (explicit discourse information) to divide documents into smaller units. Unfortunately, such markup is not always available or ambiguous with other types of separators. For example, headers, list elements or table cells might be separated in the same way (for example using an empty line) as discourse related paragraphs. Also, the division into paragraphs may differ a lot depending on the source of the document. For example, Wikipedia entries are divided on various levels into rather small units whereas newspaper articles often include very long paragraphs.

There are several ways of automatically dividing documents into passages without relying on existing markup. One way is to search for linguistic clues that indicate a separation of consecutive text blocks. These clues may include lexical patterns and relations. We refer to such approaches as *semantically motivated* document segmentation. Another approach is to cut documents into arbitrary pieces ignoring any other type of information. For example, we can use fixed-sized windows to divide documents into passages of similar size. Such windows can be defined in terms of words and characters (Kaszkiel and Zobel, 2001; Monz, 2003) or sentences and paragraphs (Zobel et al., 1995; Llopis et al., 2002). It is also possible to allow varying window sizes and overlapping sections to be indexed (Kaszkiel and Zobel, 2001; Monz, 2003). In this case it is up to the IR engine to decide which of the competing window types is preferred and it may even return overlapping sections multiple times.

In the following sections we will discuss two techniques of semantically motivated document segmentation and compare them to simple window-based techniques in terms of passage retrieval and QA performance.

2 Passage Retrieval in our QA system

Our QA system is an open-domain question answering system for Dutch. It includes two strategies: (1) A table-lookup strategy using fact databases that have been created off-line, and, (2) an "on-line" answer extraction strategy with passage retrieval and subsequent answer identification and ranking modules. We will only look at the second strategy as we are interested in the passage retrieval component and its impact on QA performance.

The passage retrieval component is implemented as an interface to several open-source IR engines. The query is generated from the given natural language question after question analysis. Keywords are sent to the IR engine(s) and results (in form of sentence IDs) are returned to the QA system.

In the experiments described here, we apply Zettair (Lester et al., 2006), an open-source IR engine developed by the search engine group at the RMIT University in Melbourne, Australia. It implements a very efficient standard IR engine with high retrieval performance according to our experiments with various alternative systems. Zettair is optimized for speed and is very efficient in both indexing and retrieval. The outstanding speed in indexing is very fortunate for our experiments in which we had to create various indexes with different document segmentation strategies.

3 Document Segmentation

We now discuss the different methods for document segmentation, starting with the semantically motivated ones and then looking at the window-based techniques.

3.1 Using Coreference Chains

Coreference is the relation which holds between two NPs both of which are interpreted as referring to the same unique referent in the context in which they occur ((Van Deemter and Kibble, 2000)). Since the coreference relation is an equivalence relation and consequently a transitive relation chains of coreferring entities can be detected in arbitrary documents. We can use these coreference chains to demarcate passages in the text. The assumption in this approach is that coreference chains mark semantically coherent passages, which are good candidates for splitting up documents.

Figure 1 illustrates chains detected by a resolution system in five successive sentences.

1. [Jim McClements en Susan Sandvig-Shobe]$_i$ hebben een onrechtmatig argument gebruikt.
2. [De Nederlandse scheidsrechter]$_j$ [Jacques de Koning]$_j$ bevestigt dit.
3. [Kuipers]$_k$ versloeg zondag in een rechtstreeks duel [Shani Davis]$_m$.
4. Toch werd [hij]$_k$ in de rangschikking achter [de Amerikaan]$_m$ geklasseerd.
5. [De twee hoofdarbiters]$_i$ verklaarden dat [Kuipers']$_k$ voorste schaats niet op de grond stond.

Cluster i (1,5): [Jim McClements en Susan Sandvig-Shobe] [De twee hoofdarbiters]

Cluster j (2): [De Nederlandse scheidsrechter] [Jacques de Koning]

Cluster k (3-5): [Kuipers] [hij] [Kuipers']

Cluster m (3,4): [Shani Davis] [de Amerikaan]

Figure 1: Example of coreference chains used for document segmentation

The coreferential units can then be used to form passages consisting of all sentences the coreference chain spans over, i.e. the boundaries of passages are sentences containing the first occurrence of the referent and the last occurrence of a referent. Thus, in the example in figure 1 we obtain four passages: 1) sentence one to sentence five, 2) sentence two, 3) sentence three to five, and, 4) sentence three and four. Note that such passages can be included in others and may overlap with yet others. Furthermore, there might be sentences which are not included in any chain which have to be handled by some other techniques.

For our purposes we used our own coreference resolution system which is based on information derived from Alpino, a wide-coverage dependency parser for Dutch (van Noord, 2006). We approached the task of coreference resolution as a clustering-based ranking task. Some NP pairs are more likely to be coreferent than others. The system ranks possible antecedents for each anaphor considering syntactic features, semantic features and surface structure features from the anaphor and the candidate itself, as well as features from the cluster to which the candidate belongs. It picks the most likely candidate as the coreferring antecedent.

References relations are detected between pronouns, common nouns and named entities. The resolution system yields a precision of 67.9% and a recall of 45.6% (F-score = 54.5%) using MUC scores (Vilain et al., 1993) on the annotated test corpus developed by (Hoste, 2005) which consist of articles taken from KNACK, a Flemish weekly news magazine.

3.2 TextTiling

TextTiling is a well-known algorithm for segmenting texts into subtopic passages (Hearst, 1997). It is based on the assumption that a significant portion of a set of lexical items in use during the course of a given subtopic discussion changes when that subtopic in the text changes.

Topic shifts are found by searching for lexical co-occurrence patterns and comparing adjacent blocks. First the text is subdivided into pseudo-sentences of a predefined size rather than using syntactically-determined sentences. These pseudo-sentences are called token-sequences by Hearst.

The algorithm identifies discourse boundaries by calculating a score for each token-sequence gap. This score is based on two methods, block comparison and vocabulary introduction. The block comparison method compares adjacent blocks of text to see how similar they are according to how many words the adjacent blocks have in common. The vocabulary introduction method is based on how many new words were seen in the interval in which the token-sequence gap is the midpoint.

The boundaries are assumed to occur at the largest valleys in the graph that results from plotting the token-sequences against their scores. In this way the algorithm produces a flat subtopic structure from a given document.

3.3 Window-based

The simplest way of dividing documents into passages is to use a fixed-sized window. Here we do not take any discourse information nor semantic clue into account but split documents at arbitrary positions. Windows can be defined in various ways, in terms of characters, words or sentences. In our case it is important to keep sentences together because of the answer extraction component in our QA system that works on that level and expects complete sentences. Window-based segmentation techniques may be applied with various amounts of overlaps. The simplest method is to split documents into passages in a greedy way, starting a new passage immediately after the previous one (and starting the entire process at the beginning of each document)[1]. Another method is to allow some overlap between consecutive passages, i.e. starting a new passage at some position within the previous one. If we use the maximum possible overlap such an approach is usually called a “sliding window” in which the difference between two consecutive passages is only two basic units (sentences) - the first and the last one.

4 Experiments

4.1 Setup

For our experiments we applied the Dutch newspaper corpus used at the QA track at CLEF, the cross-language evaluation forum. It contains about 190,000 documents consisting of about 4,000,000 sentences (roughly 80 million words). As mentioned earlier, we applied the open-source IR engine, Zettair, in our experiments and used a language modeling metric with Dirichlet smoothing, which is implemented in the system.

The evaluation is based on 778 Dutch CLEF questions from the QA tracks in the years 2003 – 2005 which are annotated with their answers. We use simple matching of possible answer strings to determine if a passage is relevant for finding an accepted answer or not. Similarly, answer string matching is applied to evaluate the output of the entire QA system; i.e. an answer by the system is counted as correct if it is identical to one of the accepted answer strings without looking at the supporting sentence/passage. For evaluation we used the standard measure of MRR which is defined as follows:

[1]Note that in our approach we still keep the document boundaries intact, i.e. the segmentation ends at the end of each document and starts from scratch at the beginning of the next document. In this way, the last passage in a document may be smaller than the pre-defined fixed size.

$$MRR_{QA} = \frac{1}{N}\sum_{1}^{N}\frac{1}{rank(\text{first\_correct\_answer})}$$

Using the string matching strategy for evaluation this corresponds to the *lenient* MRR measures frequently used in the literature. *Strict* MRR scores (requiring a match with supporting documents) is less appropriate for our data coming from the CLEF QA tracks. In CLEF there are usually only a few participants and, therefore, only a small fraction of relevant documents are known for the given questions.

4.2 Evaluation of Passage Retrieval

There are various metrics that can be employed for evaluating passage retrieval. Commonly it is argued that passage retrieval for QA is merely a filtering task and ranking (precision) is less important than recall. Therefore, the measure of *redundancy* has been introduced which is defined as the average number of relevant passages retrieved per question (independent of any ranking). Passage retrieval is, of course, a bottleneck in QA systems that make use of such a component. The system has no chance to find an answer if the retrieval engine fails to return relevant passages. Therefore, another measure, *coverage* is often used in combination with redundancy. It is defined as the proportion of questions for which at least one relevant passage is found. In order to validate the use of these measures in our setup we experimented with retrieving various amounts of paragraphs. Figure 2 illustrates the relation of coverage and redundancy scores compared to the overall QA performance measured in terms of MRR scores.

From the figure we can conclude that coverage is more important than redundancy in our system. In other words, our QA system is quite good in finding appropriate answers if there is at least one relevant passage in the set of retrieved ones. Redundancy on the other hand does not seem to provide valuable insides for the end-to-end performance of our QA system.

However, our system also uses the passage retrieval score (and, hence, the ranking) as a clue for answer extraction. Therefore, other standard IR measures might be interesting for our investigations as well. The following three metrics are common in the IR literature.

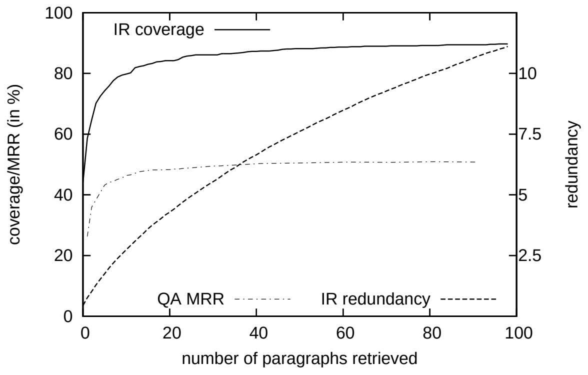

Figure 2: The correlation between coverage and redundancy and MRR_{QA} with varying numbers of paragraphs retrieved. Note that redundancy and coverage use different scales on the y-axis which makes them not directly comparable. The intention of this plot is to illustrate the tendency of both measures in comparison with QA performance.

Mean average precision (MAP): Average of precision scores for top k documents; MAP is the mean of these averages over all the N queries.

$$MAP = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{K}\sum_{k=1}^{K}P_n(1..k)$$

($P_n(1..k)$ is the precision of the top k documents retrieved for query q_n)

Uninterpolated average precision (UAP): Average of precision scores at each *relevant* document retrieved; UAP is the mean of these averages over the N queries.

$$UAP = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{|D_r^n|}\sum_{k:d_k\in D_r^n}P_n(1..k)$$

(D_r^n is the set of relevant documents among the ones retrieved for question n)

Mean reciprocal ranks: The mean of the reciprocal rank of the first relevant passage retrieved.

$$MRR_{IR} = \frac{1}{N}\sum_{1}^{N}\frac{1}{rank(\text{first\_relevant\_passage})}$$

In figure 3 the correlation of these measures with the overall QA performance is illustrated.

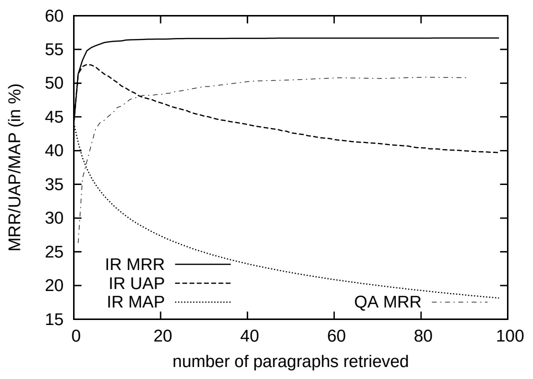

Figure 3: The correlation between IR evaluation measures (MAP, UAP and MRR_{IR}) and QA evaluation scores (MRR_{QA}) with varying numbers of paragraphs retrieved.

From the picture we can clearly see that the MRR_{IR} scores correlate the most with the QA evaluation scores when retrieving different numbers of paragraphs. This, again, confirms the importance of coverage as the MRR_{IR} score only takes the first relevant passage into account and ignores the fact that there might be more answers to be found in lower ranked passages. Hence, MRR_{IR} seems to be a good measure that combines coverage with an evaluation of the ranking and, therefore, we will use it as our main IR evaluation metric instead of coverage, redundancy, MAP & UAP.

4.3 Baselines

The CLEF newspaper corpus comes with paragraph markup which can easily be used as the segmentation granularity for passage retrieval. Table 1 shows the scores obtained by different baseline retrieval approaches using either sentences, paragraphs or documents as base units.

We can see from the results that document retrieval (used for QA) is clearly outperformed by both sentence and paragraph retrieval. Surprisingly, sentence retrieval works even better than paragraph retrieval when looking at the QA performance even though all IR evaluation measures (cov, red, MRR_{IR}) suggest a lower score. Note that MRR_{IR} is almost as good as MRR_{QA} for sentence retrieval whereas the difference between them is quite large for the other settings. This indicates the importance of narrowing down the search space for the answer extraction modules. The

				MRR		
	#sent	cov	red	IR	QA	CLEF
sent	16,737	0.784	2.95	0.490	**0.487**	**0.430**
par	80,046	0.842	4.17	0.565	0.483	0.416
doc	618,865	0.877	6.13	0.666	0.457	0.387

Table 1: Baselines with sentence (sent), paragraph (par) and document (doc) retrieval (20 units). MRR_{QA} is measured on the top 5 answers retrieved. *CLEF* is the accuracy of the QA system measured on the top answer provided by the system. *cov* refers to coverage and *red* refers to redundancy. *#sent* gives the total number of sentences included in the retrieved text units to give an impression about the amount of text to be processed by subsequent answer extraction modules.

amount of data to be processed is much smaller for sentence retrieval than for the other two while coverage is still reasonably high. The CLEF scores (accuracy measured on the top answer provided by the system) follow the same pattern. Here, the difference between sentence retrieval and document retrieval is even more apparent.

Certainly, the success of the retrieval component depends on the metric used for ranking documents as implemented in the IR engine. In order to verify the importance of document segmentation in a QA setting we also ran experiments with another standard metric implemented in Zettair, the Okapi BM-25 metric (Robertson et al., 1992). Similar to the previous setting using the LM metric, QA with paragraph retrieval (now yielding $MRR_{QA} = 0.460$) outperforms QA with document retrieval ($MRR_{QA} = 0.449$). However, sentence retrieval does not perform as well ($MRR_{QA} = 0.420$) which suggests that the Okapi metric is not suited for very small retrieval units. Still, the success of paragraph retrieval supports the advantage of passage retrieval compared to document retrieval and suggests potential QA performance gains with improved document segmentation strategies. In the remaining we only report results using the LM metric for retrieval due to its superior performance.

4.4 Semantically Motivated Passages

As described earlier, coreference chains can be used to extract semantically coherent passages from textual documents. In our experiments we used several settings for the integration of such passages in the retrieval engine. First of all, coreference chains have been used as the only way of forming passages. Sentences which are not included in any passage are included as single-sentence passages. This settings is referred to as *sent/coref*.

In the second setting we restrict the passages in length. Coreference chains can be arbitrary long and, as we can see in the results in table 2, the IR engine tends to prefer long passages which is not desirable in the QA setting. Hence, we define the constraint that passages have to be longer than 200 characters and shorter than 1000. This setup is referred to as *sent/coref (200-1000)*.

In the third setting we combine paragraphs (using existing markup) and coreference chain passages including the length restriction. This is mainly to get rid of the single-sentence passages included in the previous settings. Note that all paragraphs are used even if all sentences within them are included in coreferential passages. Note also that in all settings passages may refer to overlapping text units as coreference chains may stretch over various overlapping passages of a document.

We did not perform an exhaustive optimization of the length restriction. However, we experimented with various settings and 200-1000 was the best performing one in our experiments. For illustration we include one additional experiment using a slightly different length constraint (200-400) in table 2.

For the document segmentation strategy using TextTiling we used a freely available implementation of that algorithm (the Perl Module `Lingua::EN::Segmenter::TextTiling` available at CPAN). Note that we do not include other passages in this approach (paragraphs using existing markup nor single-sentence passages).

Table 2 summarizes the scores obtained by the various settings when applied for passage retrieval and when embedded into the QA system.

It is worth noting that including coreferential chains without length restriction forced the retrieval engine to return a lot of very long passages which resulted in a degraded QA performance (also in terms of processing time which is not shown here). The combination of paragraphs and coreferential passages with length restrictions produced MRR_{QA} scores above the baseline. However, these improvements are not statistically significant according to the Wilcoxon matched-pair signed-ranks test and looking at the corresponding CLEF scores we can even see a slight drop in performance. Applying TextTiling yielded improved scores in both passage retrieval and QA performance (MRR_{QA} and CLEF). The MRR_{QA} improvement is statistically significant according to the same test.

		MRR		
	#sent	IR	QA	CLEF
sent/coref	490,968	**0.604**	0.469	0.405
sent/coref (200-1000)	76,865	0.535	0.462	0.395
par+coref (200-1000)	82,378	0.560	**0.493**	0.426
par+coref (200-400)	67,580	0.555	**0.489**	0.422
TextTiling	107,879	**0.586**	△ **0.503**	**0.434**

Table 2: Passage retrieval with document segmentation using coreference chains and TextTiling (retrieving a maximum of 20 passages; △ means significant with $p < 0.05$ and Wilcoxon Matched-pair Signed-Ranks Test compared to paragraph baseline – only tested for MRR_{QA})

4.5 Window-based Passages

In comparison to the semantically motivated passages discussed above we also looked at simple window-based passages as described earlier. Here we do not consider any linguistic clues for dividing the documents besides the sentence and document boundaries. Table 3 summarizes the results obtained for various fixed-sized windows used for document segmentation.

		MRR		
	#sent	IR	QA	CLEF
2 sentences	33468	0.545	△ **0.506**	**0.443**
3 sentences	50190	0.554	**0.504**	**0.436**
4 sentences	66800	**0.581**	△ **0.512**	**0.447**
5 sentences	83575	**0.588**	**0.493**	0.422
6 sentences	100110	**0.583**	**0.489**	0.423
7 sentences	116872	**0.572**	**0.491**	0.422
8 sentences	133504	**0.577**	0.481	0.409
9 sentences	150156	**0.578**	0.475	0.405
10 sentences	166810	**0.596**	0.470	0.396

Table 3: Passage retrieval with window-based document segmentation (△ means significant with $p < 0.05$ and Wilcoxon Matched-pair Signed-Ranks Test)

Surprisingly, we can see that window-based segmentation approaches with small sizes between 2 and 7 sentences yield improved scores compared to the baseline. Two of the improvements (using 2-sentence passages and 4-sentence passages) are statistically significant. Three settings also outperform the best semantically motivated segmentation approach. This result was unexpected especially considering the naive way of splitting documents into parts disregarding any discourse structure (besides document boundaries) and other semantic clues.

We did another experiment using window-based segmentation and a sliding window approach. Here, fixed-sized passages are included starting at every point in the document and, hence, various overlapping passages are included in the index. In this way we split documents at various points and leave it to the IR engine to select the most appropriate ones for a given query. The results are shown in table 4.

		MRR		
	#sent	IR	QA	CLEF
2 sent (sliding)	29095	0.548	△ **0.516**	**0.456**
3 sent (sliding)	36415	0.549	**0.484**	0.411
4 sent (sliding)	41565	0.546	0.476	0.409
5 sent (sliding)	45737	0.534	0.465	0.403
6 sent (sliding)	49091	0.528	0.454	0.390
7 sent (sliding)	51823	0.529	0.439	0.372
8 sent (sliding)	54600	0.535	0.428	0.360
9 sent (sliding)	57071	0.531	0.420	0.351
10 sent (sliding)	59352	0.542	0.420	0.354

Table 4: Passage retrieval with window-based document segmentation and a sliding window

Again, we see a significant improvement with 2-sentence passages (the overall best score so far) but the performance degrades when increasing the window size. Note that the number of sentences retrieved is growing very slowly for larger windows. This is because more and more of the overlapping regions are retrieved and, hence, the total number of unique sentences does not grow with the size of the window as we have seen in the non-sliding approach.

5 Discussion & Conclusions

Our experiments show that passage retrieval is indeed different to general document retrieval. Improved retrieval scores do not necessarily lead to better QA performance. Important for QA is to reduce the search space for subsequent answer extraction modules and, hence, passage retrieval has to balance retrieval accuracy and retrieval size. In our setup it seems to be preferable to return very small units with a reasonable coverage. For this, index-time passaging is very effective.

In this study we were especially interested in semantically motivated approaches to document segmentation. In particular, two techniques have been investigated, one using the well-known TextTiling algorithm and one using coreference chains for passage boundary detection. We compared them to simple window-based techniques using various sizes. From our experiments we can conclude that simple document segmentation techniques using small fixed-sized windows work best among the ones tested here. Semantically motivated passages in the retrieval component helped to slightly improve QA performance but do not justify the effort spent in producing them. One of the main reasons for the failure of using coreference chains for segmentation might be the fact that this approach produces many overlapping passages which does not seem to be favorable for passage retrieval. This can also be seen in the sliding window approach which did not perform as well as the one without overlapping units (except for two-sentence passages). In conclusion, *uniformity* in terms of length and *uniqueness* (in terms of non-overlapping contents) seem to be more important than semantic coherence for one-step passage retrieval in QA. A future direction could be to test an approach that balances both a uniform document segmentation and semantic coherence.

References

Callan, James P. 1994. Passage-level evidence in document retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 302–310, New York, NY, USA. Springer-Verlag New York, Inc.

Greenwood, Mark A. 2004. Using pertainyms to improve passage retrieval for questions requesting information about a location. In *Proceedings of the Workshop on Information Retrieval for Question Answering (SIGIR 2004)*, Sheffield, UK.

Hearst, Marti A. and Christian Plaunt. 1993. Subtopic structuring for full-length document access. In *Research and Development in Information Retrieval*, pages 59–68.

Hearst, Marti A. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64.

Hoste, V. 2005. *Optimization Issues in Machine Learning of Coreference Resolution*. Ph.D. thesis, University of Antwerp.

Kaszkiel, Marcin and Justin Zobel. 1997. Passage retrieval revisited. In *SIGIR '97: Proceedings of the 20th annual international ACM SIGIR conference on*

Research and development in information retrieval, pages 178–185, New York, NY, USA. ACM Press.

Kaszkiel, Marcin and Justin Zobel. 2001. Effective ranking with arbitrary passages. *Journal of the American Society of Information Science*, 52(4):344–364.

Lester, Nicholas, Hugh Williams, Justin Zobel, Falk Scholer, Dirk Bahle, John Yiannis, Bodo von Billerbeck, Steven Garcia, and William Webber. 2006. The Zettair search engine. http://www.seg.rmit.edu.au/zettair/.

Llopis, F., J. Vicedo, and A. Ferrández. 2002. Passage selection to improve question answering. In *Proceedings of the COLING 2002 Workshop on Multilingual Summarization and Question Answering*.

Moldovan, D., S. Harabagiu, M. Pasca, R. Mihalcea, R. Girju, R. Goodrum, and V. Rus. 2000. The structure and performance of an open-domain question answering system.

Monz, Christof. 2003. *From Document Retrieval to Question Answering*. Ph.D. thesis, University of Amsterdam.

Roberts, Ian and Robert Gaizauskas. 2004. Evaluating passage retrieval approaches for question answering. In *Proceedings of 26th European Conference on Information Retrieval*.

Robertson, Stephen E., Steve Walker, Micheline Hancock-Beaulieu, Aarron Gull, and Marianna Lau. 1992. Okapi at TREC-3. In *Text REtrieval Conference*, pages 21–30.

Tellex, S., B. Katz, J. Lin, A. Fernandes, and G. Marton. 2003. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM Press.

Van Deemter, K. and R. Kibble. 2000. On coreferring: Coreference in muc and related annotation schemes. *Computational Linguistics*, 26(4):629–637.

van Noord, Gertjan. 2006. **A**t **L**ast **P**arsing **I**s **N**ow **O**perational. In *TALN 2006 Verbum Ex Machina, Actes De La 13e Conference sur Le Traitement Automatique des Langues naturelles*, pages 20–42, Leuven.

Vilain, M., J. Burger, J. Aberdeen, D. Connolly, and L. Hirschman. 1993. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th conference on Message understanding (MUC 6)*, pages 45–52.

Zobel, Justin, Alistair Moffat, Ross Wilkinson, and Ron Sacks-Davis. 1995. Efficient retrieval of partial documents. *Information Processing and Management*, 31(3):361–377.