Combining Sources of Evidence for Recognition of Relevant Passages in Texts*

Alexander Gelbukh[1,2], NamO Kang[1], and SangYong Han[1,**]

[1] Chung-Ang University, Korea
kang@archi.cse.cau.ac.kr, hansy@cau.ac.kr
[2] National Polytechnic Institute, Mexico
www.Gelbukh.com

Abstract. Automatically recognizing in large electronic texts short selfcontained passages relevant for a user query is necessary for fast and accurate information access to large text archives. Surprisingly, most search engines practically do not provide any help to the user in this tedious task, just presenting a list of whole documents supposedly containing the requested information. We show how different sources of evidence can be combined in order to assess the quality of different passages in a document and present the highest ranked ones to the user. Specifically, we take into account the relevance of a passage to the user query, structural integrity of the passage with respect to paragraphs and sections of the document, and topic integrity with respect to topic changes and topic threads in the text. Our experiments show that the results are promising.

1 Introduction

The huge amount of textual information available nowadays makes it impossible for a person to read all documents in order to find the information of intrest. Hence the necessity of development of automatic means for recognition of spots in the text carrying information corresponding to a given criterion.

Currently the most frequent approach to finding relevant information in large text collections is *document retrieval* [1, 17]. In response to a user query, a document retrieval system returns a list of documents (ranked by relevance) supposedly containing relevant information – somewhere in the text. This approach works perfectly when the documents are relatively small. However, when the documents contain more than several paragraphs, the user usually has to perform a second task: to search for the relevant information inside the text of the document. Surprisingly, current systems offer very little help in this task, which is in fact the bottleneck of the whole process.

Alternatives to document retrieval have been suggested for certain types of queries. One of such approaches is question answering [4, 6]. When the user is interested in a simple factoid question, e.g., *Who won the Nobel Peace Prize in 1992?*, returning a bunch of long documents is particularly inappropriate, being a much more meaningful reaction of the system just one phrase: *Rigoberta Menchú Tum* [6].

* Work done under partial support of the ITRI of Chung-Ang University, Korea, and for the first author, Korean Government (KIPA) and Mexican Government (SNI, CONACyT, The first author is currently on Sabbatical leave at Chung-Ang University.

** Corresponding author

F.F. Ramos et al. (Eds.): ISSADS 2005, LNCS 3563, pp. 283–290, 2005.

However, for more complicated types of queries such an approach is not possible. For example, when the user needs to know the development of some sequence of events, e.g., *What was the history of the wars between England and France?*, or a detailed explanation on some topic, e.g., *What is the structure of an information retrieval system?*. Such information is often contained in very long documents: one can imagine, for example, a tractate on British history or a textbook on natural language processing. In this case the user would expect some help from the system to find the relevant piece of information in the long text.

An intermediate approach between full document retrieval and question answering is *passage retrieval* known also as *passage extraction*. Given a user query, a passage retrieval system returns a (ranked) list of text segments extracted from a long document (or a document collection) that contain the requested information.

In this paper we present a general architecture of a passage retrieval system and discuss the factors that contribute to assessment of passage quality and thus the ranking of retrieved passages.

The paper is organized as follows. In Section 2 we discuss our motivation and related work. In Section 3 we outline our approach and present the details on each of the sources of evidence for ranking the retrieved passages: scoring by relevance to the user query (Section 3.1) and by alignment of their boundaries to structural units of the text and to topic change points (Section 3.2). In Section 4 we present our experimental results, and in Section 5 draw conclusions and discuss possible future work.

2 Motivation and Related Work

Applications of Passage Retrieval

Most literature on passage extraction is devoted to three main applications: text comparison, summarization, and question answering.

A number of authors have noted that similarity in passages better reflects document similarity than traditional bag-of-word comparison methods. Indeed, if the same number of the same keywords is scattered across the whole text of one of the documents (and thus these keywords are unrelated to each other) but concentrated in specific places of the second one (and thus these words describe a common idea), the two documents are scarcely similar. Passage-level comparison has been suggested to solve this problem [5, 9]. The key idea is that short passages are extracted from each of the two documents, and the sets of these excerpts are compared. With this, only similar groupings of the words lead to high document similarity. Document similarity measure improved with passage extraction has been applied to document retrieval [5, 11], as well as to document clustering and classification tasks [10].

Other applications of passage extraction are related to spotting specific, or removing irrelevant, information in long texts. In text summarization, the passages extracted from the document can be combined in a kind of summary [13]. A problem in such task is that the passages must not overlap and in particular a passage cannot be a part of a longer passage. Other applications do not impose such constraints [6].

A task similar to passage extraction but with smaller pieces of text to be found is information extraction. In this task, simple patterns carrying specific information are to be detected in a large document collection or flow, usually to fill in a relational

database. Full passage extraction techniques have been applied to detect the passages likely to contain the desired information, for their further processing [3].

A task where even smaller piece of information is to be found for a specific request is question answering, as discussed in Section 1. Similarly to information extraction, full passage extraction was applied to spot the relevant information in large texts [4, 6, 8].

However, as discussed in Section 1, our motivation is a direct application of passage extraction to the interactive information retrieval – a kind of question answering with questions implying some narrative (and not just a name or a number) as answer. Thus the extracted passages are to be directly presented to the user. This requires special attention to the quality of the passages, leading to the desiderata from Section 3.

Previous Work

Early passage extraction techniques concentrated on finding whole paragraphs or sections of a document most relevant to the user query [11]. They do not adapt themselves to the situation when the section or paragraph is too short or too long for a good passage.

Later, sliding windows of fixed length were used as candidates for passages [3, 8]. The research was concentrated on the selection of an optimal window size. Variable-size windows have been applied, too. However, no specific attention has been paid to check self-containedness of the passages, as described in Section 3.

The main issue in selecting the candidate windows is assessment of their quality (weighting). Traditionally, vector space model [1] is used to detect the passages relevant to a given query. Salton *et al*. [12] introduced the balance between global and local term weighting: the terms of the query that are used in many other documents (or passages) are less important, while the terms used many times in the current document (passage) are more important. We develop on this idea in Section 3.1.

Other cues on the importance of a text unit have been exploited. For example, a passage that is similar (in some measure of similarity, such as cosine measure of the vector space model) to many other paragraphs in the document, is more important [13]. This idea is quite similar to Google's PageRank algorithm [14]. Comparison using semantic units has been employed to improve the selection of important passages [9, 10]. While determining important paragraphs in a text is necessary for text summarization without customizing the summary for the given user query, it does not help much in finding passages relevant to the specific query, which might not coincide with the main topic of the document intended by its author – which is our goal in this paper.

3 The Algorithm

We consider the task of selecting passages that are likely answers to a user query, for subsequent presenting them to the user. For a piece of text to be a passage answering the question, it should:

- be relevant: contains relevant information, and
- be "a good passage" rather than a detached extract out of context.

To be as self-contained as possible, the passage should not imply information communicated in the previous part of the document; in particular, it should not refer to entities or develop on the ideas just introduces. We suggest that the passages more likely to be self-contained are those that correspond to

– structural units of the text, especially the beginning of structural units, and/or
– thematic threads in the text.

The general architecture of our approach is presented in Fig. 1. After some preprocessing of the text of the document, currently consisting mainly of stemming [16], candidate windows are generated. For our experiments we used all possible windows in a range of sizes (from 5 to 1000 words) as candidates; stricter criteria for candidate generation can be applied for sake of performance.

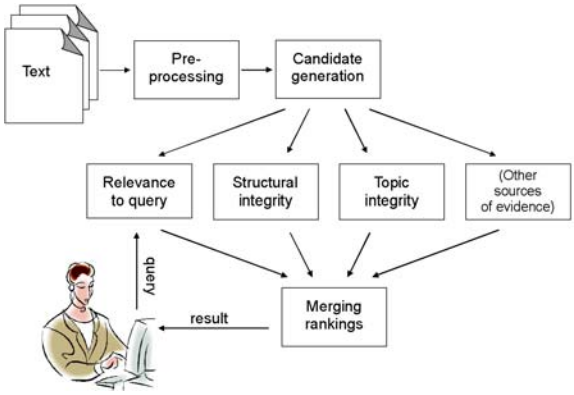

Fig. 1. The general architecture of the system.

Then each candidate window is assessed independently by its relevance to the query as well as by the criteria of self-contaidness: structural and topic integrity; more assessment criteria might be added in the future. The scores assigned to a candidate window by each module are combined, and the highest scored passages are presented to the user, with the proper ranking. To combine the rankings we use multiplication; with this, the passages scored as irrelevant receive zero total score even if they are otherwise good (self-contained) passages.

Thus, we can prefer a good self-contained passage even over a more relevant but less understandable one. This does not mean that we hide relevant information from the user. Since our variable-size windows are overlapping, for a relevant but too short to be self-contained passage there usually exists a longer one containing it but properly aligned to the formal and thematic structure of the document. Its relevance score will be lower since the share of the relevant keywords in it is lower (among a greater amount of other words). However, it still presents to the user the same information, but with the necessary context added. Thus the main goal of our combined scoring is not to reject relevant passages but to extend them to be self-contained, i.e., to prefer whenever possible a slightly larger but self-contained passage.

In the following subsections we discuss each of the scoring blocks individually.

3.1 Assessing Relevance

To assess relevance, we use the classical techniques known in information retrieval. However, we take advantage of the general context of the document to performs certain disambiguation that helps matching the passage with the query.

Boolean and Vector Space Estimation

For the passages to more likely contain the full requested information, and also for sake of efficiency, we first apply Boolean selection of candidates: we only consider the windows that contain all the query terms except stopwords. This can be turned off if the returned set is too small. Then we order the retrieved set by the similarity with the query.

To compare a candidate window with a query, we use the traditional vector space similarity measure [1]. We represent both the passage and the query as vectors, being coordinates the frequencies of individual words. Then the similarity between the query q and a passage p is expressed as an angle between the two vectors in the Euclidean affine space:

$$sim(p,q) = \frac{\sum_i F_{p,i} F_{q,i}}{\sqrt{\sum_i F_{p,i}^2} \sqrt{\sum_i F_{q,i}^2}},$$

where $F_{p,i}$ and $F_{q,i}$ are the weighted frequencies of the term i in the passage and query, respectively: $F_{\cdot,i} = w_i f_{\cdot,i}$; the summation is by all terms occurring in the document. Here $f_{\cdot,i}$ is the frequency of the term in the passage or query, and the meaning of the coefficients w_i is described in the following subsection.

Term Weighting in Context

We determine the importance weights w_i of individual terms combining the global context of the document collection (or language in general) with the local context of the given document. In document retrieval, the wellknown IDF weighting is used [1]: $w_i = \log (N/n_i)$, where N is the total number of documents in the collection and n_i is the number of documents containing the term i; when the large document is not a part of a collection, the figures from a general language corpus are used.

However, unlike in full document retrieval, a passage is in the linearly ordered immediate context of surrounding paragraphs. Firstly, they provide additional information on the passage that is not contained in it directly. This information is useful, for example, for word sense disambiguation and anaphora resolution. Secondly, some of the surrounding paragraphs are more closely related with the passage in question (being located nearer in the text) than others. Thus, we construct an IDF-like expression using the surrounding paragraphs. What is more, to reflect the closeness of the paragraph to the given passage, we scale the expression by a smooth function decreasing with the linear distance. Thus, we use the following expression:

$$w_i = \log\left(\frac{N}{n_i} + \sum_k e^{-a\left(\frac{d_i}{D}\right)^2} \frac{1}{n_{k,i}}\right),$$

where d_k is the distance in paragraphs from the given paragraph k to the passage in question, D is the maximal such distance in the document, $n_{k,i}$ is the number of occurrences of the term i in the paragraph k, and a is a coefficient which we determine empirically; we experimented with $a = 1$. The idea underlining decreasing of the weight of the words frequent in the document is that they are likely to express the general topic of the document already known to the user.

Additionally, we can modulate the weight of the query terms, thus controlling the desired size of the passages: the higher the weight of the query terms the longer the passages retrieved, since the noise words contribute less in the length of the vectors.

3.2 Assessing Structural and Topic Integrity

Structural Integrity

As we have discussed, "good" passages selected for presentation to the user, to be understandable should be, desirably, self-contained. We suggest that at the boundaries of the structural units of the document, such as paragraphs, sections, and chapters, the passages are less likely to develop on the ideas introduced previously or to require further explanations in the following text. Thus, we boost (score higher) those passages that begin at the structural boundaries of the document, and (in a smaller degree) those that end at such boundaries. In our experiments, the structural integrity module in Fig. 1 assigns the score 1.0 to every candidate window, and then increments it by the following values found empirically (further research is necessary to tune these values):

Boundary	Beginning of passage	End of passage
Paragraph	0.2	0.1
Section	1.0	0.3
Chapter	2.0	0.5

Topic Integrity

Structural boundaries do not always correspond to the topic changes in the text: for example, a group of paragraphs can comprise a topic thread, while some long paragraphs contain more than one topic thread [2]. For each candidate window, we estimate the strength of the topic change at its boundaries in the way much like area boundaries are detected in images: by the degree of difference between the points (in our case, words) to the left and to the right of the boundary and the degree of similarity of the points at the same side of the suspected boundary. We rely on the notion of *word relatedness* (similar to color similarity in image processing): some words are known to be related more than others, e.g., *galaxy* and *astronomer* are more related than *galaxy* and *baker*. Several word relatedness measures have been suggested [7, 15]. We use the following formula:

$$S = S(b) + 0.3S(e), \quad S(x) = \frac{\sum_{i<j<x \vee x<i<j} e^{-a(x-i)^2} e^{-a(x-j)^2} R(i,j)}{1 + \sum_{i<x<j} e^{-a(x-i)^2} e^{-a(x-j)^2} R(i,j)},$$

where S is the topical integrity score of the window, b is the position of the beginning of the window, e of the end, $R(i,j)$ is the relatedness of the words i and j, and a is a coefficient; we experimented with $a = 0.05$. Note that proper alignment of the beginning of the passage is more important than that of the end.

4 Experimental Results

We have implemented our method in a system that given a long text and a user query, presents a list of passages that are likely to contain the requested information. For preprocessing, we only applied Porter stemmer [16]. We used low weight for the query terms to get short passages, to be able to present them in the paper. Only passages consisting of complete sentences were considered. The top three passages returned by the system for the query "wars between England and France" on the text of *A Child's History of England* by Charles Dickens, 164,772 words, are as follows:

Rank	Score	Passage
1	0.49	*The Queen's husband who was now mostly abroad in his own dominions and generally made a coarse jest of her to his more familiar courtiers was at war with France and came over to seek the assistance of England. England was very unwilling to engage in a French war for his sake but it happened that the King of France at this very time aided a descent upon the English coast.*
2	0.48	*As his one merry head might have been far from safe if these things had been known they were kept very quiet and war was declared by France and England against the Dutch.*
3	0.46	Same as 1 plus the continuation: *Hence war was declared greatly to Philip's satisfaction and the Queen raised a sum of money with which to carry it on by every unjustifiable means in her power.*

As one can see, the lack of semantic processing (ignoring the word *between* in the query) results in some passages in fact unrelated to the query, like the second passage in the table. Elements of meaning understanding can be a topic of future work.

5 Conclusions and Future Work

We have presented a method of retrieving passages suitable for presenting them to a human user. We were mostly interested in self-contaidness of the extracted passages, which should improve their understandability. This aspect is specific for passage retrieval intended for human users, as compared with full document retrieval or passage extraction for automatic use, as described in Section 2. Our first results are promising.

In the future we plan to investigate the possibility of combining the retrieved passages in a summary of the document customized for the given query. One of the additional scoring sources in Fig. 1 will penalize the candidates with pronouns in the initial part of the window, since such windows are likely not self-contained. We also plan to further exploit the context of the passage within the document for word sense disambiguation and anaphora resolution, which is not very important – and not easy – in full document retrieval but is necessary in passage retrieval. Finally, we will consider merging the passages extracted from different documents; here the term weights (1) should be adjusted to be comparable between documents.

References

1. R. Baeza-Yates, B. Ribeiro-Neto. *Modern Information Retrieval*. Addison Wesley, 1999.
2. I. A. Bolshakov, A. Gelbukh. Text segmentation into paragraphs based on local text cohesion. *Text, Speech and Dialogue* (TSD-2001), Lecture Notes in Artificial Intelligence N 2166, Springer, 2001, pp. 158–166.
3. C. Cardie. Empirical Methods in Information Extraction. *AI Magazine*, 18:4, 65_79 1997.
4. C. L. A. Clarke, G. V. Cormack, T. R. Lynam, E. L. Terra. Question Answering by Passage Selection. In *Advances in Open Domain Question Answering*, Kluwer, 2004.
5. G. V. Cormack, C. L. A. Clarke, C. R. Palmer, S. S. L. To. Passage-Based Query Refinement. *Information Processing and Management*, 36(1):133-153, 2000.
6. A. Del-Castillo-Escobedo, M. Montes-y-Gómez, L. Villaseñor-Pineda. QA on the Web: A Preliminary Study for Spanish Language. *Proc. of ENC-2004*, IEEE, 2004.
7. G. Hirst, D. St-Onge. Lexical chains as representations of context for the detection and correction of malapropisms. In: C. Fellbaum (Ed.), *WordNet: An electronic lexical database*, Cambridge, MA: The MIT Press, 1998.
8. F. LLopis, J. L. Vicedo, A. Ferrández. Passage Selection to Improve Question Answering. In: *Multilingual Summarization and Question Answering* (COLING-2002), 2002.
9. H. Mochizuki, M. Iwayama, M. Okumura, Passage-Level Document Retrieval Using Lexical Chains, RIAO 2000, pp. 491-506.
10. Y. Nakao. A Method for Related-passage Extraction based on Thematic Hierarchy. *IPSJ Transactions on Databases* 42 (SIG 10 (TOD 11)), 2001, pp. 39–53.
11. G. Salton, J. Allan, C. Buckley. Approaches to passage retrieval in full text information systems. In: 16th annual international ACM SIGIR conf. on Research and development in information retrieval, US, 1993, 49–58.
12. G. Salton, and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
13. G. Salton, A. Singhal, M. Mitra, C. Buckley. Automatic text structuring and summarization. Mani, I., Maybury, M. (Eds), Advances in automatic text summarization. MIT, 1999.
14. L. Page and S. Brin. The anatomy of a large-scale hypertextual web search engine. Proc. 7th Intl. WWW Conf., 107-117, 1998.
15. S. Patwardhan, S. Banerjee, T. Pedersen. Using Measures of Semantic Relatedness for Word Sense Disambiguation. A. Gelbukh (Ed.), Computational Linguistics and Intelligent Text Processing (CICLing-2003), Lecture Notes in Computer Science N 2588, Springer, 2003, p. 241–257.
16. M.F.Porter. An algorithm for suffix stripping. *Program*, **14** no. 3, pp 130-137, July 1980.
17. T. Strzalkowski (Ed.). Natural Language Information Retrieval. Kluwer, 1999.