Biased LexRank: Passage Retrieval using Random Walks with Question-Based Priors

Jahna Otterbacher [a], Gunes Erkan [b], Dragomir R. Radev [a,b,*]

[a] *School of Information, University of Michigan, Ann Arbor, MI 48109-1092*

[b] *Computer Science and Engineering, University of Michigan, Ann Arbor, MI 48109-1092*

Abstract

We present Biased LexRank, a method for semi-supervised passage retrieval in the context of question answering. We represent a text as a graph of passages linked based on their pairwise lexical similarity. We use traditional passage retrieval techniques to identify passages that are likely to be relevant to a user's natural language question. We then perform a random walk on the lexical similarity graph in order to recursively retrieve additional passages that are similar to other relevant passages. We present results on several benchmarks that show the applicability of our work to question answering and topic-focused text summarization.

Key words: question answering, lexrank biased lexrank, lexical centrality, passage retrieval, semi-supervised learning, biased random walks

1 Introduction

Text summarization is one of the hardest problems in information retrieval, mainly because it is not very well-defined. There are various definitions of text summarization resulting from different approaches to solving the problem. Furthermore, there is often no agreement as to what a good summary is even when we are dealing with a particular definition of the problem. In this paper, we focus on the *query-based* or *focused* summarization problem where we seek to generate a summary of a set of related documents given a specific aspect of their common topic formulated as a natural language query. This is in contrast

* Corresponding author.

Email addresses: jahna@umich.edu (Jahna Otterbacher), gerkan@umich.edu (Gunes Erkan), radev@umich.edu (Dragomir R. Radev).

to *generic* summarization, where a set of related documents is summarized without a query, with the aim of covering as much salient information in the original documents as possible.

The motivation behind focused summarization is that readers often prefer to see specific information about a topic in a summary rather than a generic summary (e.g. (Tombros and Sanderson, 1998)). An example summarization problem from the Document Understanding Conferences (DUC) 2006 [1] is as follows:

Example 1
- *Topic: International adoption*
- *Focus: What are the laws, problems, and issues surrounding international adoption by American families?*

Given a set of documents about a topic (e.g. "international adoption"), the systems are required to produce a summary that *focuses* on the given, specific aspect of that topic. In more general terms, this task is known as *passage retrieval* in information retrieval. Passage retrieval also arises in question answering as a preliminary step: given a question that typically requires a short answer of one or a few words, most question answering systems first try to retrieve passages (sentences) that are relevant to the question and thus potentially contain the answer. This is quite similar to summarization with the key difference being that the summarization queries typically look for longer answers that are several sentences long.

In the current work, we propose a unified method for passage retrieval with applications to multi-document text summarization and passage retrieval for question answering. Our method is a query-based extension of the LexRank summarization method introduced in (Erkan and Radev, 2004). LexRank is a random walk-based method that was proposed for generic summarization. Our contribution in this paper is to derive a graph-based sentence ranking method by incorporating the query information into the original LexRank algorithm, which is query independent. The result is a very robust method that can generate passages from a set of documents given a query of interest.

An important advantage of the method is that it has only a single parameter to tune that effectively determines how much the resultant passage should be generic (query-independent) or query-based. Therefore, in comparison to supervised learning approaches, it does not require a lot of training data. In addition, it does not make any assumptions about the structure of the language in which the documents are written and does not require the use of any particular linguistic resources (as in (Tiedemann, 2005), (Woods et al., 2000)) and therefore its potential applications are quite broad. Finally, in con-

[1] `http://duc.nist.gov`

trast to methods for sentence selection that primarily consider the similarity of the candidate sentences to the query (e.g. (Llopis et al., 2002), (Allan et al., 2003), (Turpin et al., 2007)), Biased LexRank exploits the information gleaned from intra-sentence similarities as well. We previously presented this method in (Otterbacher et al., 2005). Here, we extend our experiments to include the summarization problem, and show that our approach is very general with promising results for more than one information retrieval problem.

2 Our Approach: Topic-sensitive LexRank

We formulate the summarization problem as sentence extraction, that is, the output of our system is simply a set of sentences retrieved from the documents to be summarized. To determine the sentences that are most relevant to the user's query, we use a probabilistic model to rank them. After briefly describing the original version of the LexRank method, previously introduced in (Erkan and Radev, 2004) in Section 2.1, we then present in Section 2.2 an adapted, topic-sensitive (i.e. "biased") version, which will be evaluated in two sentence retrieval experiments. More specifically, in Section 4 we apply Biased LexRank to the problem of topic-focused summarization and in Section 5 we evaluate it in the context of passage retrieval for question answering.

2.1 The LexRank Method

In (Erkan and Radev, 2004), the concept of graph-based centrality was used to rank a set of sentences for producing *generic* multi-document summaries. To compute LexRank, the documents are first segmented into sentences, and then a *similarity graph* is constructed where each node in the graph represents a sentence. The edge relation between the nodes is induced by a similarity metric of choice, as will be explained in the details of our experiments. In a generalized form the LexRank equation can be written as:

$$\mathrm{LR}(u) = \frac{d}{N} + (1-d) \sum_{v \in adj[u]} \frac{w(v,u)}{\sum_{z \in adj[v]} w(v,z)} \mathrm{LR}(v) \qquad (1)$$

where $\mathrm{LR}(u)$ is the LexRank value of sentence u, N is the total number of sentences (nodes) in the graph, d is a damping factor to be determined empirically, $adj[u]$ is the set of the sentences that are neighbors of u in the graph, and $w(v,u)$ is the weight of the link from sentence v to sentence u. Therefore, the LexRank value of a node (sentence) is a constant term plus the (weighted) average of the LexRank values of its neighboring nodes.

An interesting interpretation of the LexRank value of a sentence can be understood in terms of the concept of a random walk. A random walk on a graph is the process of *visiting* the nodes of the graph according to a specified *transition probability* distribution. Suppose we have a sentence similarity graph as described above. We define a random walk on this graph in such a way that it starts at a random sentence and then at each step, with probability d it jumps to a random sentence with uniform probability, with probability $1 - d$ it visits a sentence that is adjacent to the current sentence with a probability in proportion to the outgoing edge (similarity) weights of the current sentence. The LexRank value of a sentence gives us the limiting probability that such a random walk will visit that sentence *in the long run*. Equivalently, the LexRank value is the *fraction* of the time such a random walk spends on the particular sentence. The LexRank Equation 1 as described above is defined in a recursive manner, and can be computed via an iterative routine called the *power method*[2] . An extractive summarization method that is almost equivalent to LexRank with cosine links was independently proposed in (Mihalcea and Tarau, 2004).

The motivating assumption behind the LexRank method is that the information that is repeated many times in a cluster of sentences is the salient information that needs to be represented in a summary. This correlation between the repeated information and the salient information is the starting intuition that most summarization systems try to exploit. LexRank makes the observation that if a sentence is similar to a lot of other sentences in a cluster, then it contains common information with other sentences; therefore it is a good candidate to be included in an extractive summary. Note that such a sentence will be strongly connected to a lot of other sentences in the similarity graph. The random walk we described above is more likely to visit a sentence that is better connected to the rest of the graph with strong links since the direction of the random walk is determined by the similarity-weighted edges in the graph. Thus the LexRank value of such a sentence will be higher. Furthermore, LexRank takes into account not only the similarity values of a sentence to its neighbors but also the individual importance of the neighbors of that sentence. This is achieved by the recursive formulation of LexRank and the propagation of the importance from node to node by the random walk.

[2] The stationary distribution is unique and the power method is guaranteed to converge provided that the Markov chain is ergodic (Seneta, 1981). A non-ergodic Markov chain can be made ergodic by reserving a small probability for jumping to any other state from the current state (Page et al., 1998).

2.2 Biased LexRank

In deriving a topic-sensitive or biased version of LexRank, we begin with the generalized form of the LexRank equation as described in Section 2.1 (Equation 1). To induce the sentence graph, any symmetric or asymmetric metric may be used for $w(v, u)$. It can be noted that there is nothing in Equation 1 that favors certain sentences based on a topic focus: LexRank is completely *unsupervised* in the sense that it only depends on the overall structure of the graph. The first term, $\frac{d}{N}$, is introduced to make the matrix ergodic so that a solution to the equation exists. It does not have a big impact on the final ranking of the nodes since it favors all the nodes equally during the random walk. With probability d, the random walk jumps to any node with uniform probability. This suggests an alternative view of the random walk process. We can combine more than one random walk into a random walk process. Indeed, we could use a non-uniform distribution in combination with the random walk based on the weight function $w(\cdot, \cdot)$.

Suppose we have a prior belief about the ranking of the nodes in the graph. This belief might be derived from a baseline ranking method which we trust to a certain extent. For example, in the focused summarization task, we can rank the sentences by looking at their similarity to the topic description. Let $b(u)$ be the score of sentence u based on this baseline method. We can then *bias* the random walk based on $b(\cdot)$ while computing LexRank as follows:

$$\mathrm{LR}(u) = d \cdot \frac{b(u)}{\sum_{z \in C} b(z)} + (1 - d) \sum_{v \in adj[u]} \frac{w(v, u)}{\sum_{z \in adj[v]} w(v, z)} \mathrm{LR}(v) \qquad (2)$$

where C is the set of all nodes in the graph.[3] We call Equation 2 *biased* or *topic-sensitive* LexRank since it favors certain set of sentences during the random walk based on a prior distribution. When $d = 1$, $\mathrm{LR}(\cdot)$ ranks the nodes exactly the same as $b(\cdot)$. When $d < 1$, we have a mixture of the baseline scores and the LexRank scores derived from the *unbiased* structure of the graph. In other words, Biased LexRank ranks the sentences by looking at the baseline method and the inter-sentence similarities at the same time. Figure 1 shows an illustration.

[3] As a technical detail, note that the corresponding matrix is ergodic if $b(u) > 0$ for all u's, thus the solution to the above equation exists.

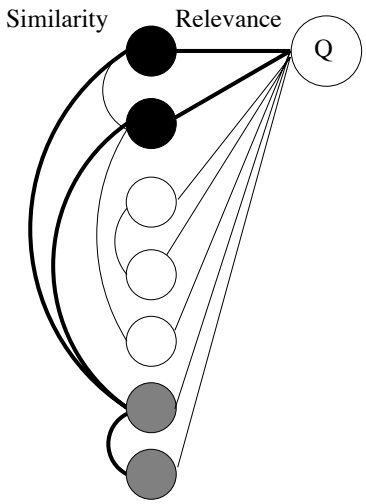

Fig. 1. An illustration of Biased LexRank. The nodes (sentences) are initially ranked by their baseline relevance to the query Q. However, the rankings of the bottom two shaded nodes are expected to be boosted up since they are similar to the top ranked nodes and each other.

3 A Question Answering Example

A problem closely related to focused summarization is question answering (QA). Essentially, the only difference between QA and focused summarization is that QA addresses questions that require very specific and short answers that are usually only a few words long whereas summarization involves questions that can be answered by composing a short document of a few sentences. A crucial first step for most QA systems is to retrieve the sentences that potentially contain the answer to the question (Gaizauskas et al., 2004). Consider the following example in which answers to the given question are sought from a set of topically related documents.

Example 2

Question: What was the **plane***'s* **destination***?*

(1) The **plane** *was* **destined** *for Italy's capital Rome.*

(2) The **plane** *was in route from Locarno in Switzerland, to its* **destination***, Rome, Italy.*

(3) The pilot was on a 20-minute flight from Locarno, Switzerland to Milan.

(4) The aircraft had taken off from Locarno, Switzerland, and was heading to Milan's Linate airport.

These four sentences were taken from various news articles about the same event - the crash of a small plane into a skyscraper in Milan, Italy. As can be seen, there is some contradictory information among them since they come from different sources published at different points in time. However, in a question answering scenario all of them need to be retrieved. In the absence of

any useful external information, a popular sentence scoring technique in QA is to look at the words in the question. If a sentence has some words in common with the question, it is considered to be a *relevant* sentence that may contain the answer to the question. Given a sentence u and a question q, an example sentence scoring formula, as used in (Allan et al., 2003), is as follows:

$$\text{rel}(u|q) = \sum_{w \in q} \log(tf_{w,u} + 1) \times \log(tf_{w,q} + 1) \times \text{idf}_w \quad (3)$$

where $tf_{w,u}$ and $tf_{w,q}$ are the number of times w appears in u and q, respectively, and idf_w is the inverse document frequency of the word w (Salton and Buckley, 1988). Using this technique, one is able to conclude that the first two sentences in Example 2 are somewhat related to the question since they include words from the question (shown in boldface).[4] However, the last two sentences are also clearly related to the question and actually contain the answer. An important observation is that sentences 3 and 4 have some words in common with sentences 1 and 2. Knowing that sentences 1 and 2 are relevant or important, it is easy to infer that sentences 3 and 4 should also be relevant only by looking at the inter-sentence similarities between them. Hence, a suitable choice for the Biased LexRank formula is:

$$\text{LR}(u|q) = d \cdot \frac{\text{rel}(u|q)}{\sum_{z \in C} \text{rel}(z|q)} + (1-d) \sum_{v \in adj[u]} \frac{w(v,u)}{\sum_{z \in adj[v]} w(v,z)} \text{LR}(v|q) \quad (4)$$

The random walk interpretation of this formula is as follows: With probability d, the random walk visits a sentence with a probability proportional to its relevance to the question, with probability $(1-d)$ the random walk chooses a sentence that is a neighbor of the current sentence with a probability proportional to the link weight in the graph. As can be seen, the random walk is biased towards the *neighborhoods* of the highly relevant sentences in the graph.

4 Application to Focused Summarization

The Document Understanding Conferences summarization evaluations in 2005 and 2006 included a focused summarization task. Given a topic and a set of 25 relevant documents, the participants were required "to synthesize a fluent, well-organized 250-word summary of the documents that answers the

[4] *was* and *the* are considered to be stop words, so they are not included in the word matching process.

question(s) in the topic statement." An example topic statement and related questions are shown in Example 1. In this section, we explain how we formulated the summarization tasks of DUC 2005 and 2006 based on the biased LexRank technique detailed in Section 2.2.

4.1 Using Generation Probabilities as Link Weights

In approaching the task of focused summarization, we use language model-based similarity measures between the sentences as proposed by (Kurland and Lee, 2005). Here, we first recall the language modeling approach to information retrieval and adapt it to the summarization domain. Given a sentence v, we can compute a (unigram) language model from it. A straightforward way of computing this language model is the maximum likelihood estimation (MLE) of the probabilities of the words to occur in v:

$$p_{\mathrm{ML}}(w|v) = \frac{\mathrm{tf}_{w,v}}{|v|} \tag{5}$$

where $\mathrm{tf}_{w,v}$ is the number of times the word w occurs in sentence v. The MLE is often not a good approximation for a language model since the words that do not occur in the text from which we compute the word frequencies get zero probability. This is an even bigger problem when we compute language models from a relatively shorter input text such as a sentence composed of only a few words. To account for the unseen words, we smooth the language model computed from a sentence using the language model computed from the entire cluster:

$$p_{\mathrm{JM}}(w|v) = (1-\lambda)p_{\mathrm{ML}}(w|v) + \lambda p_{\mathrm{ML}}(w|C) \tag{6}$$

where C is the entire document cluster. Equation 6 is a special instance of the more general Jelinek-Mercer smoothing method (Jelinek and Mercer, 1980), where the constant $\lambda \in [0,1]$ is a trade-off parameter between the MLE computed from the sentence and the MLE computed from the entire cluster. $p_{\mathrm{JM}}(w|v)$ is nonzero for all words that occur in the document cluster to be summarized provided that $\lambda > 0$.

We can also consider the *generation probability* of a sentence given the language model computed from another sentence. For example,

$$p_{\mathrm{gen}}(u|v) = \prod_{w \in u} p_{\mathrm{JM}}(w|v)^{\mathrm{tf}_{w,u}} \tag{7}$$

defines the generation probability of sentence u given the language model of sentence v. Since the generation probability of a given sentence is the product of the probabilities of all the words it contains, longer sentences tend to get smaller generation probabilities. Therefore, we normalize the generation probability of each sentence by the sentence length to make different pairwise similarities comparable to each other:

$$p_{\text{norm}}(u|v) = \left(\prod_{w \in u} p_{\text{JM}}(w|v)^{\text{tf}_{w,u}} \right)^{\frac{1}{|u|}} \tag{8}$$

We use $p_{\text{norm}}(u|v)$ as the weight of the link *from* u *to* v in the graph-based representation of the cluster. Note that $p_{\text{norm}}(u|v)$ is not necessarily equal to $p_{\text{norm}}(v|u)$. The probability of a 1-step random walk (i.e. a random walk of length 1) from u to v is proportional to the (normalized) generation probability of u given the language model computed from v. The reason we are not using the value for the opposite direction ($p_{\text{norm}}(v|u)$) is that each sentence should get credit for its capacity to generate the other sentences, not to be generated by them. If a sentence has strong incoming generation links in the graph, this is evidence that the language model of that sentence can generate other sentences more successfully. Revisiting the random walk model of LexRank, the LexRank value of a sentence is a measure of its accumulated *generation power*, that is, how likely it is to generate the rest of the cluster from the language model of the specific sentence in the long run. We advocate that a sentence with a high generation power is a good candidate for the summary of its cluster.

Extending the use of generation probabilities to Biased LexRank for the focused summarization task is straightforward. For the baseline ranking method, we use the generation probability of the topic description from the sentences. A sentence is ranked higher if its language model can generate the topic description with a larger probability. This is analogous to the language modeling approach in information retrieval (Ponte and Croft, 1998) where the documents are ranked with respect to the generation probability of the given query from each document's language model. In our summarization method, given a topic description t, the final score for a sentence u is computed by the following Biased LexRank equation:

$$\text{LR}(u|t) = d \cdot \frac{p_{\text{gen}}(t|u)}{\sum_{z \in C} p_{\text{gen}}(t|z)} + (1-d) \sum_{v \in adj[u]} \frac{p_{\text{norm}}(v|u)}{\sum_{z \in adj[v]} p_{\text{norm}}(v|z)} \text{LR}(v|t) \tag{9}$$

4.2 DUC 2005 and 2006 Experiments

In the Document Understanding Conferences (DUC) 2005 and 2006 summarization evaluations, the task was to summarize a set of documents based on a particular aspect of their common topic. This particular aspect was provided as a "topic description" (see Example 1). Other than this change, the setting was similar to DUC 2003 and 2004 evaluations: There were 50 topical clusters to be summarized for each year. Each cluster had 25 English news documents that concerned the same topic.

There are two parameters in our framework summarized by Equation 9 above. d is the biased jump probability in Biased LexRank, which is a trade-off between the similarity of a sentence to the topic description and to other sentences in the cluster. λ is the Jelinek-Mercer smoothing parameter (Equation 6) that is used when computing the language model of each sentence. For both parameters, we experimented with several values in the $[0.1, 0.9]$ interval. Here, we report the results for one of the best parameter settings we obtained for the DUC 2005 dataset.

It should be noted that we did not carry out an extensive parameter tuning. Rather, our goal was to show that the Biased LexRank method is effective even when little or no parameter turning is possible. To further support this claim, we did not perform any parameter tuning for the DUC 2006 dataset at all, directly using the same parameter values from the DUC 2005 experiments. Overall, $d \approx 0.7$ and $\lambda \approx 0.6$ performed well. Note that 0.7 is a relatively large value for d considering that we set it to 0.15 in the generic summarization experiments in (Erkan and Radev, 2004). However, using large values for d makes perfect sense for focused summarization since we would certainly want to give more weight to a sentence's similarity to the topic description than its similarity to other sentences. For small values of d, the summaries would be more generic rather than based on the topic description.

In the similarity graphs, we connected each node (sentence) to k other nodes that are most similar to it rather than connecting to all the other nodes in the graph. We observed that this improves not only the running time of the algorithm but also the quality of the resultant summaries. One reason for this may be that small similarity values actually indicate "dissimilarity" among sentences. Accumulating scores from dissimilar neighbors of a node is not intuitive no matter how small the similarity values are. In our experiments, we considered only the top 20 most similar neighbors for each sentence, that is, each node in the similarity graphs has exactly 20 outgoing links. Note that the number of incoming links for each node may vary depending on how similar a node is to the rest of the nodes in the graph. Indeed, a good summary sentence would typically have more incoming links than outgoing links, which

is an indication that its language model can better generate the rest of the sentences in the graph.

In selecting the sentences for inclusion in the focused summaries, we did not use any features other than the Biased LexRank values in our experiments. To construct the final summaries, we ranked the sentences based on their Biased LexRank scores. The ranked sentences were added to a summary one by one until the summary exceeded 250 words which was the limit in the DUC evaluations. When considering the ranked sentences for inclusion into the summary, we ignored any sentence which has a *cosine* similarity of larger than 0.5 to any of the sentences that were ranked above it. This simple reranking scheme ensures that the resulting summaries cover as much information as possible within the length limit.

For the evaluation of our summaries, we used the official ROUGE metrics of DUC 2005 and 2006, i.e. ROUGE-2 and ROUGE-SU4 (Lin and Hovy, 2003). ROUGE-2 compares the bigram overlap between the system summary and the manual summaries created by humans. ROUGE-SU4 does the same except that it introduces the relaxation of allowing as many as four words between the two words of a bigram.

Tables 1 and 2 show the results for DUC 2005 and 2006, respectively. In both datasets, it can be seen that the performance of Biased LexRank is comparable to that of the human summarizers. In some cases, its performance is not significantly different from certain human summarizers considering the 95% confidence intervals. For comparison, we also include the top ranked system's score each year. It can be seen that LexRank achieves almost the same scores or better.

5 Application to Passage Retrieval for Question Answering

In this section, we show how Biased LexRank can be applied effectively to the problem of passage retrieval for question answering. As noted by (Gaizauskas et al., 2004), while passage retrieval is the crucial first step for question answering, QA research has typically not emphasized it. As explained in Section 3, we formulate passage retrieval at the sentence level, that is, we aim to extract sentences from a set of documents in response to a question. We demonstrate that Biased LexRank significantly improves question-focused sentence selection over a baseline that only looks at the overlap between the sentences and the query.

Table 1
ROUGE-2 and ROUGE-SU4 score comparison against the human summarizers in DUC 2005 along with their 95% confidence intervals. A-J: human summarizers; LR: Biased LexRank; Best: best system in the original evaluations.

	ROUGE-2		ROUGE-SU4	
A	0.11711	[0.10578,0.12865]	0.17560	[0.16595,0.18552]
B	0.10020	[0.08643,0.11533]	0.16102	[0.14997,0.17283]
C	0.11785	[0.10469,0.13062]	0.17798	[0.16587,0.18931]
D	0.10035	[0.08880,0.11336]	0.16155	[0.15120,0.17342]
E	0.10508	[0.09160,0.11959]	0.15922	[0.14712,0.17163]
F	0.09977	[0.08877,0.11138]	0.15822	[0.14781,0.16919]
G	0.09683	[0.08546,0.10833]	0.15954	[0.15010,0.16905]
H	0.08811	[0.07682,0.10055]	0.14798	[0.13795,0.16023]
I	0.09888	[0.08467,0.11642]	0.16116	[0.14652,0.17897]
J	0.09968	[0.08960,0.11133]	0.16058	[0.15126,0.17180]
LR	0.07531	[0.07203,0.07858]	0.13630	[0.13274,0.13969]
Best	0.07440	[0.07169,0.07736]	0.13458	[0.13173,0.13749]

5.1 Description of the Problem

Our goal is to build a question-focused sentence retrieval mechanism using the Biased LexRank method. In contrast to previous passage retrieval systems such as Okapi (Robertson et al., 1992), which ranks documents for relevancy and then proceeds to find paragraphs related to a question, we address the finer-grained problem of finding sentences containing answers. In addition, the input is a set of documents relevant to the topic of the query that the user has already identified (e.g. via a search engine). Our method does not rank the input documents, nor is it restricted in terms of the number of sentences that may be selected from the same document.

The output produced by Biased LexRank, a ranked list of sentences relevant to the user's question, can be subsequently used as input to an answer selection system in order to find specific answers from the extracted sentences. Alternatively, the sentences can be returned to the user as a question-focused summary. This is similar to "snippet retrieval" (Wu et al., 2004). However, in our approach, answers are extracted from a set of multiple documents rather than on a document-by-document basis.

Table 2
ROUGE-2 and ROUGE-SU4 score comparison against the human summarizers in DUC 2006 along with their 95% confidence intervals. A-J: human summarizers; LR: Biased LexRank; Best: best system in the original evaluations.

	ROUGE-2		ROUGE-SU4	
A	0.10361	[0.09260,0.11617]	0.16829	[0.16042,0.17730]
B	0.11788	[0.10501,0.13351]	0.17665	[0.16356,0.19080]
C	0.13260	[0.11596,0.15197]	0.18385	[0.17012,0.19878]
D	0.12380	[0.10751,0.14003]	0.17814	[0.16527,0.19094]
E	0.10365	[0.08935,0.11926]	0.16298	[0.15012,0.17606]
F	0.10893	[0.09310,0.12780]	0.16043	[0.14518,0.17771]
G	0.11324	[0.10195,0.12366]	0.17121	[0.16301,0.17952]
H	0.10777	[0.09833,0.11746]	0.16665	[0.15627,0.17668]
I	0.10634	[0.09632,0.11628]	0.16843	[0.15828,0.17851]
J	0.10717	[0.09293,0.12460]	0.16934	[0.15716,0.18319]
LR	0.09232	[0.08806,0.09684]	0.15010	[0.14598,0.15417]
Best	0.09505	[0.09093,0.09914]	0.15464	[0.15061,0.15837]

5.2 Relevance to the Question

We first stem all of the sentences in a set of articles and compute word IDFs by the following formula:

$$\text{idf}_w = \log\left(\frac{N+1}{0.5+sf_w}\right) \tag{10}$$

where N is the total number of sentences in the cluster, and sf_w is the number of sentences that the word w appears in.

We also stem the question and remove the stop words from it. Then the relevance of a sentence s to the question q is computed by:

$$\text{rel}(s|q) = \sum_{w \in q} \log(tf_{w,s}+1) \times \log(tf_{w,q}+1) \times \text{idf}_w \tag{11}$$

where $tf_{w,s}$ and $tf_{w,q}$ are the number of times w appears in s and q, respectively. This model has proven to be successful in query-based sentence retrieval (Allan et al., 2003), and is used as our competitive baseline in this study (e.g. Tables 6, 7 and 9).

5.3 The Mixture Model

The baseline system explained above does not make use of any inter-sentence information in a cluster. We hypothesize that a sentence that is similar to the high scoring sentences in the cluster should also have a high score. For instance, if a sentence that gets a high score in our baseline model is likely to contain an answer to the question, then a related sentence, which may not be similar to the question itself, is also likely to contain an answer.

This idea is captured by the following mixture model, where $p(s|q)$, the score of a sentence s given a question q, is determined as the sum of its relevance to the question (using the same measure as the baseline described above) and the similarity to the other sentences in the document cluster:

$$p(s|q) = d\,\frac{\text{rel}(s|q)}{\sum_{z\in C}\text{rel}(z|q)} + (1-d)\sum_{v\in C}\frac{\text{sim}(s,v)}{\sum_{z\in C} sim(z,v)}p(v|q) \qquad (12)$$

where C is the set of all sentences in the cluster. The value of d, which we will also refer to as the "question bias," is a trade-off between two terms in the equation and is determined empirically. For higher values of d, we give more importance to the relevance to the question compared to the similarity to the other sentences in the cluster. The denominators in both terms are for normalization, which are described below. We use the cosine measure weighted by word IDFs as the similarity between two sentences in a cluster:

$$\text{sim}(x,y) = \frac{\sum_{w\in x,y}\text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i\in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2}\times\sqrt{\sum_{y_i\in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}} \qquad (13)$$

Equation 12 can be written in matrix notation as follows:

$$\mathbf{p} = [d\mathbf{A} + (1-d)\mathbf{B}]^{\text{T}}\mathbf{p} \qquad (14)$$

$\mathbf{A}$ is the square matrix such that for a given index i, all the elements in the i^{th} column are proportional to $\text{rel}(i|q)$. $\mathbf{B}$ is also a square matrix such that each entry $\mathbf{B}(i,j)$ is proportional to $sim(i,j)$. Both matrices are normalized so that row sums add up to 1. Note that as a result of this normalization, all rows of the resulting square matrix $\mathbf{Q} = [d\mathbf{A} + (1-d)\mathbf{B}]$ also add up to 1. Such a matrix is called *stochastic* and defines a Markov chain. If we view each sentence as a state in a Markov chain, then $\mathbf{Q}(i,j)$ specifies the transition probability from state i to state j in the corresponding Markov chain. The vector $\mathbf{p}$ we are looking for in Equation 14 is the stationary distribution of the Markov chain.

An illustration of this representation is shown in Figure 2. The five input sentences are represented in the graph as nodes with the cosine similarity

between each pair of sentences shown on the respective edge. In the example, node 1 is isolated since sentence 1 did not have a cosine similarity of greater than the threshold of 0.15 with any other sentence.

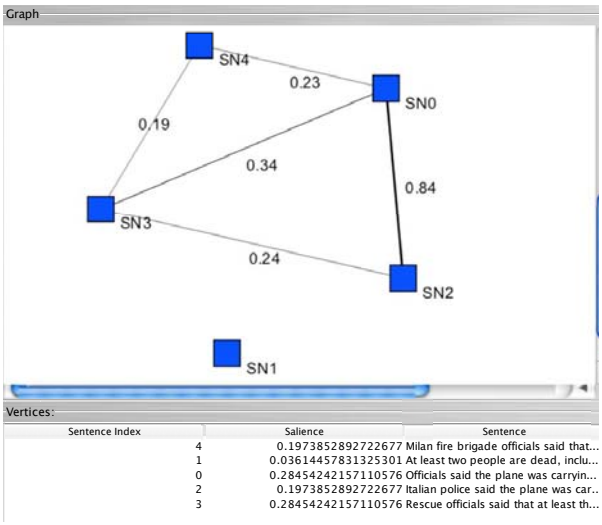

Fig. 2. Sentence similarity graph with a cosine threshold of 0.15.

With probability d, a transition is made from the current node (sentence) to the nodes that are similar to the query. With probability (1-d), a transition is made to the nodes that are lexically similar to the current node. Every transition is weighted according to the similarity distributions. Each element of the vector $\mathbf{p}$ gives the asymptotic probability of ending up at the corresponding state in the long run regardless of the starting state.

A simpler version of Equation 14, where $\mathbf{A}$ is a uniform matrix and $\mathbf{B}$ is a normalized binary matrix, is known as PageRank (Brin and Page, 1998; Page et al., 1998) and used to rank the web pages by the Google search engine. It was also the model used to rank sentences for generic summarization in (Erkan and Radev, 2004).

We experimented with different values of d on our training data. We also considered several threshold values for inter-sentence cosine similarities, where we ignored the similarities between the sentences that are below the threshold. In the training phase of the experiment, we evaluated all combinations of LexRank with d in the range of $[0, 1]$ (in increments of 0.10) and with a similarity threshold ranging from $[0, 0.9]$ (in increments of 0.05). We then found all configurations that outperformed the baseline. These configurations were then applied to our development/test set. Finally, our best sentence retrieval system was applied to our test data set and evaluated against the baseline. The remainder of this section of the paper will explain this process and the results in detail.

5.4 Corpus

A key challenge for passage retrieval for QA is that, when attempting to retrieve answers to questions from a set of documents published by multiple sources over time (e.g. in a Web-based environment), answers are typically lexically diverse and may change over time or even contradict one another. Therefore, in order to evaluate Biased LexRank on such challenging questions, we built a corpus of 20 multi-document, multi-source clusters of complex news stories. The topics covered included plane crashes, political controversies and natural disasters. The data clusters and their characteristics are shown in Table 3. The news articles were collected from the Web sites of various news agencies or from an automated news summarization system. In particular, "Newstracker" clusters were collected automatically by a Web-based news article collection and summarization system. The number of clusters randomly assigned to the training, development/test and test data sets were 11, 3 and 6, respectively.

Next, we assigned each cluster of articles to an annotator, who was asked to read all articles in the cluster. He or she then generated a list of factual questions key to understanding the story. Once we collected the questions for each cluster, two judges independently annotated nine of the training clusters. For each sentence and question pair in a given cluster, the judges were asked to indicate whether or not the sentence contained a complete answer to the question. Once an acceptable rate of interjudge agreement was verified on the first nine clusters (a Kappa (Carletta, 1996) of 0.68), the remaining 11 clusters were annotated by one judge each.

In some cases, the judges did not find any sentences containing the answer for a given question. Such questions were removed from the corpus. The final number of questions annotated for answers over the entire corpus was 341, and the distributions of questions per cluster can be found in Table 3.

5.5 Evaluation Metrics and Methods

To evaluate our sentence retrieval mechanism, we produced extract files, which contain a list of sentences deemed to be relevant to the question, for the system and from human judgment. To compare different configurations of our system to the baseline system, we produced extracts at a fixed length of 20 sentences. While evaluations of question answering systems are often based on a shorter list of ranked sentences, we chose to generate longer lists for several reasons. One is that we are developing a passage retrieval system, of which the output can then serve as the input to an answer extraction system for further

Table 3. Corpus of complex news stories.

Cluster	Sources	Articles	Questions	Data set	Sample question
Algerian terror threat	AFP, UPI	2	12	train	What is the condition under which GIA will take its action?
Milan plane crash	MSNBC, CNN, ABC, Fox, USAToday	9	15	train	How many people were in the building at the time of the crash?
Turkish plane crash	BBC, ABC, FoxNews, Yahoo	10	12	train	To where was the plane headed?
Moscow terror attack	UPI, AFP, AP	7	7	train	How many people were killed in the most recent explosion?
Rhode Island club fire	MSNBC, CNN, ABC, Lycos, Fox, BBC, Ananova	10	8	train	Who was to blame for the fire?
FBI most wanted	AFP, UPI	3	14	train	How much is the State Department offering for information leading to bin Laden's arrest?
Russia bombing	AP, AFP	2	11	train	What was the cause of the blast?
Bali terror attack	CNN, FoxNews, ABC, BBC, Ananova	10	30	train	What were the motivations of the attackers?
Washington DC sniper	FoxNews, Ha'aretz, BBC, BBC, Washington Times, CBS	8	28	train	What kinds of equipment or weapons were used in the killings?
GSPC terror group	Newstracker	8	29	train	What are the charges against the GSPC suspects?
China earthquake	Novelty 43	25	18	train	What was the magnitude of the earthquake in Zhangjiakou?
Gulfair	ABC, BBC, CNN, USAToday, FoxNews, Washington Post	11	29	dev/test	How many people were on board?
David Beckham trade	AFP	20	28	dev/test	How long had Beckham been playing for MU before he moved to RM?
Miami airport evacuation	Newstracker	12	15	dev/test	How many concourses does the airport have?
US hurricane	DUC d04a	14	14	test	In which places had the hurricane landed?
EgyptAir crash	Novelty 4	25	29	test	How many people were killed?
Kursk submarine	Novelty 33	25	30	test	When did the Kursk sink?
Hebrew University bombing	Newstracker	11	27	test	How many people were injured?
Finland mall bombing	Newstracker	9	15	test	How many people were in the mall at the time of the bombing?
Putin visits England	Newstracker	12	20	test	What issue concerned British human rights groups?

processing. In such a setting, we would most likely want to generate a relatively longer list of candidate sentences. As previously mentioned, in our corpus the questions often have more than one relevant answer, so ideally, our passage retrieval system would find many of the relevant sentences, sending them on to the answer component to decide which answer(s) should be returned to the user. Each system's extract file lists the document and sentence numbers of the top 20 sentences. The "gold standard" extracts list the sentences judged as containing answers to a given question by the annotators (and therefore have variable sizes) in no particular order. [5]

We evaluated the performance of the systems using two metrics - Mean Reciprocal Rank (MRR) (Voorhees and Tice, 2000) and Total Reciprocal Document Rank (TRDR) (Radev et al., 2005). MRR, used in the TREC Q&A evaluations, is the reciprocal rank of the first correct answer (or sentence, in our case) to a given question. This measure gives us an idea of how far down we must look in the ranked list in order to find a correct answer. To contrast, TRDR is the total of the reciprocal ranks of all answers found by the system. In the context of answering questions from complex stories, where there is often more than one correct answer to a question, and where answers are typically time-dependent, we should focus on maximizing TRDR, which gives us a measure of how many of the relevant sentences were identified by the system. However, we report both the average MRR and TRDR over all questions in a given data set.

5.6 LexRank Versus the Baseline Approach

In the training phase, we searched the parameter space for the values of d (the question bias) and the similarity threshold in order to optimize the resulting TRDR scores. For our problem, we expected that a relatively low similarity threshold pair with a high question bias would achieve the best results. Table 4 shows the effect of varying the similarity threshold. [6] The notation $LR[a, d]$ is used, where a is the similarity threshold and d is the question bias. As seen in Table 4, the optimal range for the parameter a was between 0.14 and 0.20. This is intuitive because if the threshold is too high, such that only the most lexically similar sentences are represented in the graph, the method does not find sentences that are more lexically diverse (e.g. paraphrases). Table 5 shows the effect of varying the question bias at two different similarity thresholds (0.02 and 0.20). It is clear that a high question bias is needed. However, a small probability for jumping to a node that is lexically similar to the given sentence

[5] For clusters annotated by two judges, all sentences chosen by at least one judge were included.

[6] A threshold of -1 means that no threshold was used such that all sentences in all documents were included in the graph.

Table 4
Training phase: effect of similarity threshold (a) on Ave. MRR and TRDR.

System	**Ave. MRR**	**Ave. TRDR**
LR[-1.0,0.65]	0.5270	0.8117
LR[0.02,0.65]	0.5261	0.7950
LR[0.16,0.65]	0.5131	0.8134
LR[0.18,0.65]	0.5062	0.8020
LR[0.20,0.65]	0.5091	0.7944
LR[-1.0,0.80]	0.5288	0.8152
LR[0.02,0.80]	0.5324	0.8043
LR[0.16,0.80]	0.5184	0.8160
LR[0.18,0.80]	0.5199	0.8154
LR[0.20,0.80]	0.5282	0.8152

Table 5
Training phase: effect of question bias (d) on Ave. MRR and TRDR.

System	**Ave. MRR**	**Ave. TRDR**
LR[0.02,0.65]	0.5261	0.7950
LR[0.02,0.70]	0.5290	0.7997
LR[0.02,0.75]	0.5299	0.8013
LR[0.02,0.80]	0.5324	0.8043
LR[0.02,0.85]	0.5322	0.8038
LR[0.02,0.90]	0.5323	0.8077
LR[0.20,0.65]	0.5091	0.7944
LR[0.20,0.70]	0.5244	0.8105
LR[0.20,0.75]	0.5285	0.8137
LR[0.20,0.80]	0.5282	0.8152
LR[0.20,0.85]	0.5317	0.8203
LR[0.20,0.90]	0.5368	0.8265

(rather than the question itself) is needed. Table 6 shows the configurations of LexRank that performed better than the baseline system on the training data, based on mean TRDR scores over the 184 training questions. We applied all four of these configurations to our unseen development/test data, in order to see if we could further differentiate their performances.

Table 6
Training phase: systems outperforming the baseline in terms of TRDR score.

System	**Ave. MRR**	**Ave. TRDR**
Baseline	0.5518	0.8297
LR[0.14,0.95]	0.5267	0.8305
LR[0.18,0.90]	0.5376	0.8382
LR[0.18,0.95]	0.5421	0.8382
LR[0.20,0.95]	0.5404	0.8311

Table 7
Development testing evaluation.

System	**Ave. MRR**	**Ave. TRDR**
Baseline	0.5709	1.0002
LR[0.14,0.95]	0.5882	1.0469
LR[0.18,0.90]	0.5820	1.0288
LR[0.18,0.95]	0.5956	1.0411
LR[0.20,0.95]	0.6068	1.0601

5.6.1 Development/testing Phase

Having established the optimal ranges of the question bias and similarity threshold parameters on the training data, Biased LexRank was then evaluated on the unseen development/test data. As shown in Table 7, all four LexRank systems outperformed the baseline, both in terms of average MRR and TRDR. A more detailed, cluster-by-cluster analysis of the performance of the best Biased LexRank configuration, LR[0.20,0.95], over the 72 questions for the three development/test data clusters is shown in Table 8. While LexRank outperforms the baseline system on the first two clusters both in terms of MRR and TRDR, their performances are not substantially different on the third cluster. Therefore, we examined properties of the questions within each cluster in order to see what effect they might have on system performance.

We hypothesized that the baseline system, which compares the similarity of each sentence to the question using IDF-weighted word overlap, should perform well on questions that provide many content words. To contrast, LexRank might perform better when the question provides fewer content words, since it considers both similarity to the query and inter-sentence similarity. Out of the 72 questions in the development/test set, the baseline system outperformed LexRank on 22 of the questions. In fact, the average number of content words among these 22 questions was slightly, but not significantly, higher than the

Table 8
Average scores by cluster: baseline versus LR[0.20,0.95].

Cluster	**B-MRR**	**LR-MRR**	**B-TRDR**	**LR-TRDR**
Gulfair	0.5446	0.5461	0.9116	0.9797
David Beckham trade	0.5074	0.5919	0.7088	0.7991
Miami airport evacuation	0.7401	0.7517	1.7157	1.7028

Table 9
Testing phase: baseline vs. LR[0.20,0.95].

	Ave. MRR	**Ave. TRDR**
Baseline	0.5780	0.8673
LR[0.20,0.95]	0.6189	0.9906
p-value	NA	0.0619

average on the remaining questions (3.63 words per question versus 3.46). Given this observation, we experimented with two mixed strategies, in which the number of content words in a question determined whether LexRank or the baseline system was used for sentence retrieval. We tried threshold values of 4 and 6 content words, however, this did not improve the performance over the pure strategy of system LR[0.20,0.95]. Therefore, we applied this system versus the baseline to our unseen test set of 134 questions.

5.6.2 Testing Phase

As shown in Table 9, LR[0.20,0.95] outperformed the baseline system on the test data both in terms of average MRR and TRDR scores. The improvement in average TRDR score was statistically significant with a p-value of 0.0619. Since we are interested in a passage retrieval mechanism that finds sentences relevant to a given question, providing input to the question answering component of our system, the improvement in average TRDR score is very promising. While we saw in Section 5.6.1 that LR[0.20,0.95] may perform better on some question or cluster types than others, we conclude that it beats the competitive baseline when one is looking to optimize mean TRDR scores over a large set of questions. However, in future work, we will continue to improve the performance, perhaps by developing mixed strategies using different configurations of LexRank.

5.7 Discussion

The idea behind using LexRank for sentence retrieval is that a system that considers only the similarity between candidate sentences and the input query, and not the similarity between the candidate sentences themselves, is likely to miss some important sentences. When using any metric to compare sentences and a query, there is always likely to be a tie between multiple sentences (or, similarly, there may be cases where fewer than the number of desired sentences have similarity scores above zero). LexRank effectively provides a means to break such ties. An example of such a scenario is illustrated in Tables 10 and 11, which show the top ranked sentences according to the baseline and LexRank respectively, for the question "What caused the Kursk to sink?" from the Kursk submarine cluster. It can be seen that all top five sentences chosen by the baseline system have the same sentence score (similarity to the query), yet the top ranking two sentences are not actually relevant according to the human judges. To contrast, LexRank achieved a better ranking of the sentences since it is better able to differentiate between them. As can be seen, LexRank has ordered the three relevant sentences first, followed by the two sentences that are not relevant. It should be noted that both for the LexRank and baseline systems, chronological ordering of the documents and sentences is preserved, such that in cases where two sentences have the same score, the one published earlier is ranked higher.

6 Conclusion

We have presented a generic method for passage retrieval that is based on random walks on graphs. Unlike most ranking methods on graphs, LexRank can be tuned to be biased, such that the ranking of the nodes (sentences) in the graph is dependent on a given query. The method, Biased LexRank, has only one parameter to be trained, namely, the topic or query bias.

In the current paper, we have also demonstrated the effectiveness of our method as applied to two classical IR problems, extractive text summarization and passage retrieval for question answering. In the context of the Document Understanding Conference 2005 and 2006 summarization tasks, we have shown that LexRank performed well in producing topic-focused summaries. Despite performing very limited parameter tuning for the evaluation, the LexRank method produced summaries comparable to those generated by the human summarizers. Biased LexRank was also shown to be effective in retrieving relevant passages from a set of related news articles, given a user's unaltered natural language question. More specifically, by using inter-sentence similarity in addition to the similarity between the candidate sentences and the input

Table 10
Top ranked sentences using the baseline system on the question "What caused the Kursk to sink?".

Rank	Sentence	Score	Relevant?
1	The Russian governmental commission on the accident of the submarine Kursk sinking in the Barents Sea on August 12 has rejected 11 original explanations for the disaster, but still cannot conclude what caused the tragedy indeed, Russian Deputy Premier Ilya Klebanov said here Friday.	4.2282	N
2	There has been no final word on what caused the submarine to sink while participating in a major naval exercise, but Defense Minister Igor Sergeyev said the theory that Kursk may have collided with another object is receiving increasingly concrete confirmation.	4.2282	N
3	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	4.2282	Y
4	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	4.2282	Y
5	President Clinton's national security adviser, Samuel Berger, has provided his Russian counterpart with a written summary of what U.S. naval and intelligence officials believe caused the nuclear-powered submarine Kursk to sink last month in the Barents Sea, officials said Wednesday.	4.2282	N

Table 11
Top ranked sentences using the LR[0.20,0.95] system on the question "What caused the Kursk to sink?"

Rank	**Sentence**	**Score**	**Relevant?**
1	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	0.0133	Y
2	Russian Deputy Prime Minister Ilya Klebanov said Thursday that collision with a big object caused the Kursk nuclear submarine to sink to the bottom of the Barents Sea.	0.0133	Y
3	The Russian navy refused to confirm this, but officers have said an explosion in the torpedo compartment at the front of the submarine apparently caused the Kursk to sink.	0.0125	Y
4	President Clinton's national security adviser, Samuel Berger, has provided his Russian counterpart with a written summary of what U.S. naval and intelligence officials believe caused the nuclear-powered submarine Kursk to sink last month in the Barents Sea, officials said Wednesday.	0.0124	N
5	There has been no final word on what caused the submarine to sink while participating in a major naval exercise, but Defense Minister Igor Sergeyev said the theory that Kursk may have collided with another object is receiving increasingly concrete confirmation.	0.0123	N

question, Biased LexRank finds significantly more passages containing the desired answer and it also ranks them more accurately.

Acknowledgments

This paper is based upon work supported by the National Science Foundation under Grant N. 0534323, "BlogoCenter: Infrastructure for Collecting, Mining and Accessing Blogs" and Grant No. 0329043, "Probabilistic and Link-based Methods for Exploiting Very Large Textual Repositories".

Any opinions, findings, and conclusions or recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the National Science Foundation.

We would also like to thank the current and former members of the CLAIR group at Michigan and in particular Siwei Shen and Yang Ye for their assistance with this project.

References

Allan, J., Wade, C., and Bolivar, A. (2003). Retrieval and Novelty Detection at the Sentence Level. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '03)*, pages 314–321. ACM Press.

Brin, S. and Page, L. (1998). The Anatomy of a Large-scale Hypertextual Web Search Engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Carletta, J. (1996). Assessing Agreement on Classification Tasks: The Kappa Statistic. *CL*, 22(2):249–254.

Erkan, G. and Radev, D. (2004). LexRank: Graph-based Lexical Centrality as Salience in Text. *JAIR*, 22:457–479.

Gaizauskas, R., Hepple, M., and Greenwood, M. (2004). Information Retrieval for Question Answering: a SIGIR 2004 Workshop. In *SIGIR 2004 Workshop on Information Retrieval for Question Answering*.

Jelinek, F. and Mercer, R. (1980). *Pattern Recognition in Practice, Eds. Gelsema, E. S. and Kanal, L. N.*, chapter Interpolated Estimation of Markov Source Parameters from Sparse Data. North Holland.

Kurland, O. and Lee, L. (2005). PageRank without Hyperlinks: Structural Re-ranking Using Links Induced by Language Models. In *Proceedings of the 28th Annual International ACM SIGIR Conference of Research and Development in Information Retrieval*.

Lin, C.-Y. and Hovy, E. (2003). Automatic Evaluation of Summaries Using

N-gram Co-Occurrence Statistics. In *Proceedings of the 2003 Human Language Technology Conference (NAACL-HLT '03)*, Edmonton, Canada.

Llopis, F., Vicedo, L. V., and Ferrendez, A. (2002). Passage Selection to Improve Question Answering. In *Proceedings of the COLING Workshop on Multilingual Summarization and Question Answering*.

Mihalcea, R. and Tarau, P. (2004). TextRank: Bringing Order into Texts. In Lin, D. and Wu, D., editors, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '04)*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Otterbacher, J., Erkan, G., and Radev, D. (2005). Using Random Walks for Question-focused Sentence Retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 915–922, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998). The PageRank Citation Ranking: Bringing Order to the Web. *Technical report, Stanford University, Stanford, CA*.

Ponte, J. M. and Croft, W. B. (1998). A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Confere nce of Research and Development in Information Retrieval*, pages 275–281. ACM.

Radev, D., Fan, W., Qi, H., Wu, H., and Grewal, A. (2005). Probabilistic Question Answering on the Web. *Journal of the American Society for Information Science and Technology*, 56(3).

Robertson, S. E., Walker, S., Hancock-Beaulieu, M., Gull, A., and Lau, M. (1992). Okapi at TREC. In *Text REtrieval Conference*, pages 21–30.

Salton, G. and Buckley, C. (1988). Term-weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5):513–523.

Seneta, E. (1981). *Non-negative Matrices and Markov Chains*. Springer-Verlag, New York.

Tiedemann, J. (2005). Integrating Linguistic Knowledge in Passage Retrieval for Question Answering. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, Vancouver, Canada.

Tombros, A. and Sanderson, M. (1998). Advantages of Query Biased Summaries in Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference of Research and Development in Information Retrieval*, Melbourne, Australia.

Turpin, A., Tsegay, Y., Hawking, D., and Williams, H. E. (2007). Fast Generation of Result Snippits in Web Search. In *Proceedings of the 30th Annual International ACM SIGIR Conference of Research and Development in Information Retrieval*, Amsterdam, The Netherlands.

Voorhees, E. and Tice, D. (2000). The TREC-8 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-8*, Gaithersburg, MD.

Woods, W. A., Bookman, L. A., Houston, A., Kuhns, R. J., Martin, P., and

Green, S. (2000). Linguistic Knowledge Can Improve Information Retrieval. In *Proceedings of the 6th Applied Natural Language Processing Conference (ANLP '00)*, Seattle.

Wu, H., Radev, D. R., and Fan, W. (2004). Towards Answer-focused Summarization Using Search Engines. *New Directions in Question Answering*.