Using Random Walks for Question-focused Sentence Retrieval

Jahna Otterbacher[1], **Güneş Erkan**[2], **Dragomir R. Radev**[1,2]
[1]School of Information, [2]Department of EECS
University of Michigan
{jahna,gerkan,radev}@umich.edu

Abstract

We consider the problem of question-focused sentence retrieval from complex news articles describing multi-event stories published over time. Annotators generated a list of questions central to understanding each story in our corpus. Because of the dynamic nature of the stories, many questions are time-sensitive (e.g. "How many victims have been found?") Judges found sentences providing an answer to each question. To address the sentence retrieval problem, we apply a stochastic, graph-based method for comparing the relative importance of the textual units, which was previously used successfully for generic summarization. Currently, we present a topic-sensitive version of our method and hypothesize that it can outperform a competitive baseline, which compares the similarity of each sentence to the input question via IDF-weighted word overlap. In our experiments, the method achieves a TRDR score that is significantly higher than that of the baseline.

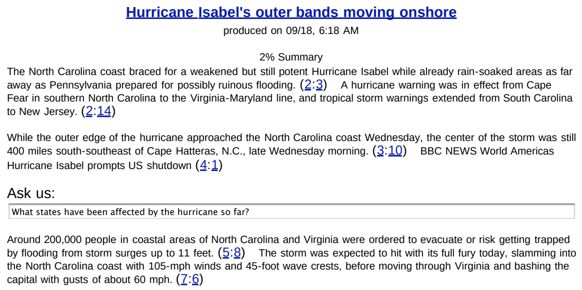
1 Introduction

Recent work has motivated the need for systems that support "Information Synthesis" tasks, in which a user seeks a global understanding of a topic or story (Amigo et al., 2004). In contrast to the classical question answering setting (e.g. TREC-style Q&A (Voorhees and Tice, 2000)), in which the user presents a single question and the system returns a corresponding answer (or a set of likely answers), in this case the user has a more complex information need.

Similarly, when reading about a complex news story, such as an emergency situation, users might seek answers to a set of questions in order to understand it better. For example, Figure 1 shows the interface to our Web-based news summarization system, which a user has queried for information about Hurricane Isabel. Understanding such stories is challenging for a number of reasons. In particular, complex stories contain many sub-events (e.g. the devastation of the hurricane, the relief effort, etc.) In addition, while some facts surrounding the situation do not change (such as "Which area did the hurricane first hit?"), others may change with time ("How many people have been left homeless?"). Therefore, we are working towards developing a system for question answering from clusters of complex stories published over time. As can be seen at the bottom of Figure 1, we plan to add a component to our current system that allows users to ask questions as they read a story. They may then choose to receive either a precise answer or a question-focused summary.

Currently, we address the question-focused sentence retrieval task. While passage retrieval (PR) is clearly not a new problem (e.g. (Robertson et al., 1992; Salton et al., 1993)), it remains important and yet often overlooked. As noted by (Gaizauskas et al., 2004), while PR is the crucial first step for question answering, Q&A research has typically not empha-

Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP), pages 915–922, Vancouver, October 2005. ©2005 Association for Computational Linguistics



Hurricane Isabel's outer bands moving onshore

produced on 09/18, 6:18 AM

2% Summary

The North Carolina coast braced for a weakened but still potent Hurricane Isabel while already rain-soaked areas as far away as Pennsylvania prepared for possibly ruinous flooding. (2:3) A hurricane warning was in effect from Cape Fear in southern North Carolina to the Virginia-Maryland line, and tropical storm warnings extended from South Carolina to New Jersey. (2:14)

While the outer edge of the hurricane approached the North Carolina coast Wednesday, the center of the storm was still 400 miles south-southeast of Cape Hatteras, N.C., late Wednesday morning. (3:10) BBC NEWS World Americas Hurricane Isabel prompts US shutdown (4:1)

Ask us:

What states have been affected by the hurricane so far?

Around 200,000 people in coastal areas of North Carolina and Virginia were ordered to evacuate or risk getting trapped by flooding from storm surges up to 11 feet. (5:8) The storm was expected to hit with its full fury today, slamming into the North Carolina coast with 105-mph winds and 45-foot wave crests, before moving through Virginia and bashing the capital with gusts of about 60 mph. (7:6)

Figure 1: Question tracking interface to a summarization system.

sized it. The specific problem we consider differs from the classic task of PR for a Q&A system in interesting ways, due to the time-sensitive nature of the stories in our corpus. For example, one challenge is that the answer to a user's question may be updated and reworded over time by journalists in order to keep a running story fresh, or because the facts themselves change. Therefore, there is often more than one correct answer to a question.

We aim to develop a method for sentence retrieval that goes beyond finding sentences that are similar to a single query. To this end, we propose to use a stochastic, graph-based method. Recently, graph-based methods have proved useful for a number of NLP and IR tasks such as document re-ranking in ad hoc IR (Kurland and Lee, 2005) and analyzing sentiments in text (Pang and Lee, 2004). In (Erkan and Radev, 2004), we introduced the LexRank method and successfully applied it to generic, multi-document summarization. Presently, we introduce topic-sensitive LexRank in creating a sentence retrieval system. We evaluate its performance against a competitive baseline, which considers the similarity between each sentence and the question (using IDF-weighed word overlap). We demonstrate that LexRank significantly improves question-focused sentence selection over the baseline.

2 Formal description of the problem

Our goal is to build a question-focused sentence retrieval mechanism using a topic-sensitive version of the LexRank method. In contrast to previous PR 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 to our system 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 system 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 of our system, 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 system answers are extracted from a set of multiple documents rather than on a document-by-document basis.

3 Our approach: topic-sensitive LexRank

3.1 The LexRank method

In (Erkan and Radev, 2004), the concept of graph-based centrality was used to rank a set of sentences, in producing generic multi-document summaries. To apply LexRank, a similarity graph is produced for the sentences in an input document set. In the graph, each node represents a sentence. There are edges between nodes for which the cosine similarity between the respective pair of sentences exceeds a given threshold. The degree of a given node is an indication of how much information the respective sentence has in common with other sentences. Therefore, sentences that contain the most salient information in the document set should be very central within the graph.

Figure 2 shows an example of a similarity graph for a set of five input sentences, using a cosine similarity threshold of 0.15. Once the similarity graph is constructed, the sentences are then ranked according to their eigenvector centrality. As previously mentioned, the original LexRank method performed well in the context of generic summarization. Below, we describe a topic-sensitive version of LexRank, which is more appropriate for the question-focused sentence retrieval problem. In the new approach, the score of a sentence is determined by a mixture model of the relevance of the sentence to the query and the similarity of the sentence to other high-scoring sentences.

3.2 Relevance to the question

In topic-sensitive LexRank, we first stem all of the sentences in a set of articles and compute word IDFs by the following formula:

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

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:

$$\text{rel}(s|q) = \sum_{w \in q} \log(tf_{w,s} + 1) \times \log(tf_{w,q} + 1) \times \text{idf}_w \tag{2}$$

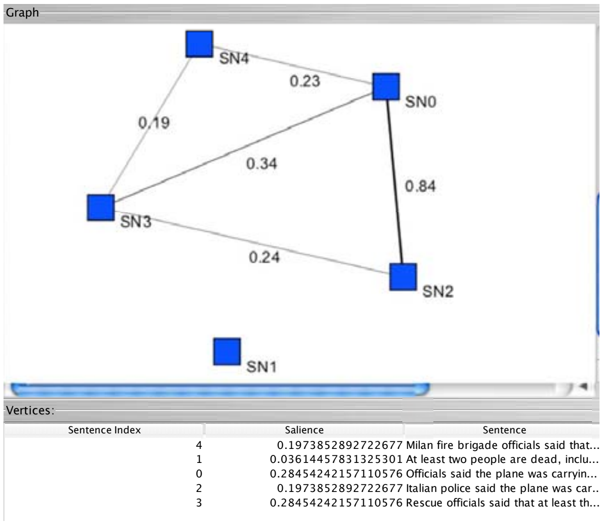
where $tf_{w,s}$ and $tf_{w,q}$ are the number of times w appears in s and q, respectively. This model has proven to be successful in query-based sentence retrieval (Allan et al., 2003), and is used as our competitive baseline in this study (e.g. Tables 4, 5 and 7).

3.3 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:

$$p(s|q) = d\,\frac{\text{rel}(s|q)}{\sum_{z \in C} \text{rel}(z|q)} + (1-d) \sum_{v \in C} \frac{\text{sim}(s,v)}{\sum_{z \in C} sim(z,v)} p(v|q) \tag{3}$$



Vertices:

Sentence Index	Salience	Sentence
4	0.1973852892722677	Milan fire brigade officials said that...
1	0.03614457831325301	At least two people are dead, inclu...
0	0.28454242157110576	Officials said the plane was carryin...
2	0.1973852892722677	Italian police said the plane was car..
3	0.28454242157110576	Rescue officials said that at least th...

Figure 2: LexRank example: sentence similarity graph with a cosine threshold of 0.15.

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:

$$\text{sim}(x,y) = \frac{\sum_{w \in x,y} \text{tf}_{w,x}\text{tf}_{w,y}(\text{idf}_w)^2}{\sqrt{\sum_{x_i \in x}(\text{tf}_{x_i,x}\text{idf}_{x_i})^2} \times \sqrt{\sum_{y_i \in y}(\text{tf}_{y_i,y}\text{idf}_{y_i})^2}} \tag{4}$$

Equation 3 can be written in matrix notation as follows:

$$\mathbf{p} = [d\mathbf{A} + (1-d)\mathbf{B}]^{\mathrm{T}}\mathbf{p} \tag{5}$$

$\mathbf{A}$ 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 5 is the stationary distribution of the Markov chain. An intuitive interpretation of the stationary distribution can be understood by the concept of a random walk on the graph representation of the Markov chain.

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. The stationary distribution of a Markov chain can be computed by a simple iterative algorithm, called power method.[1]

A simpler version of Equation 5, 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 in (Erkan and Radev, 2004).

3.4 Experiments with topic-sensitive LexRank

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 the paper will explain this process and the results in detail.

4 Experimental setup

4.1 Corpus

We built a corpus of 20 multi-document clusters of complex news stories, such as plane crashes, political controversies and natural disasters. The data clusters and their characteristics are shown in Table 1. The news articles were collected from various sources. "Newstracker" clusters were collected automatically by our Web-based news 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 (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 1.

4.2 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 PR system, of which the output can then be input to an answer extraction system for further 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 PR 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

[1] 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).

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?

Table 1: Corpus of complex news stories.

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.[2]

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 LexRank versus the baseline system

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 2 shows the effect of varying the similarity threshold.[3] The notation $LR[a, d]$ is used, where a is the similarity threshold and d is the question bias. 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 related but are more lex-

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

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

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 2: Training phase: effect of similarity threshold (a) 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

Table 3: Training phase: effect of question bias (d) on Ave. MRR and TRDR.

ically diverse (e.g. paraphrases). Table 3 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 (rather than the question itself) is needed. Table 4 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.

5.1 Development/testing phase

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

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

Table 5: Development testing evaluation.

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

The scores for the four LexRank systems and the baseline on the development/test data are shown in Table 5. This time, all four LexRank systems outperformed the baseline, both in terms of average MRR and TRDR scores. An analysis of the average scores over the 72 questions within each of the three clusters for the best system, LR[0.20,0.95], is shown in Table 6. 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 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

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

Table 7: Testing phase: baseline vs. LR[0.20,0.95].

system versus the baseline to our unseen test set of 134 questions.

5.2 Testing phase

As shown in Table 7, 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 5.1 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.

6 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 8 and 9, which show the top ranked sentences by 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 judges. To contrast, LexRank achieved a better ranking of the sentences since it is better able to differentiate between them. 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.

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

7 Conclusion

We presented topic-sensitive LexRank and applied it to the problem of sentence retrieval. In a Web-based news summarization setting, users of our system could choose to see the retrieved sentences (as in Table 9) as a question-focused summary. As indicated in Table 9, each of the top three sentences were judged by our annotators as providing a complete answer to the respective question. While the first two sentences provide the same answer (a collision caused the Kursk to sink), the third sentence provides a different answer (an explosion caused the disaster). While the last two sentences do not provide answers according to our judges, they do provide context information about the situation. Alternatively, the user might prefer to see the extracted

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

Table 9: Top ranked sentences using the LR[0.20,0.95] system on the question "What caused the Kursk to sink?"

answers from the retrieved sentences. In this case, the sentences selected by our system would be sent to an answer identification component for further processing. As discussed in Section 2, our goal was to develop a topic-sensitive version of LexRank and to use it to improve a baseline system, which had previously been used successfully for query-based sentence retrieval (Allan et al., 2003). In terms of this task, we have shown that over a large set of unaltered questions written by our annotators, LexRank can, on average, outperform the baseline system, particularly in terms of TRDR scores.

8 Acknowledgments

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

References

James Allan, Courtney Wade, and Alvaro Bolivar. 2003. Retrieval and novelty detection at the sentence level. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 314–321. ACM Press.

Enrique Amigo, Julio Gonzalo, Victor Peinado, Anselmo Peñas, and Felisa Verdejo. 2004. An Empirical Study of Information Synthesis Task. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*, pages 207–214, Barcelona, Spain, July.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

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

Gunes Erkan and Dragomir Radev. 2004. LexRank: Graph-based Lexical Centrality as Salience in Text. *JAIR*, 22:457–479.

Robert Gaizauskas, Mark Hepple, and Mark Greenwood. 2004. Information Retrieval for Question Answering: a SIGIR 2004 Workshop. In *SIGIR 2004 Workshop on Information Retrieval for Question Answering*.

Oren Kurland and Lillian Lee. 2005. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *SIGIR 2005*, Salvador, Brazil, August.

L. Page, S. Brin, R. Motwani, and T. Winograd. 1998. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford University, Stanford, CA*.

Bo Pang and Lillian Lee. 2004. A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. In *Association for Computational Linguistics*.

Dragomir Radev, Weiguo Fan, Hong Qi, Harris Wu, and Amardeep Grewal. 2005. Probabilistic Question Answering on the Web. *Journal of the American Society for Information Science and Technology*, 56(3), March.

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

G. Salton, J. Allan, and C. Buckley. 1993. Approaches to Passage REtrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–58.

E. Seneta. 1981. *Non-negative matrices and markov chains*. Springer-Verlag, New York.

Ellen Voorhees and Dawn Tice. 2000. The TREC-8 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-8*, Gaithersburg, MD.

Harris Wu, Dragomir R. Radev, and Weiguo Fan. 2004. Towards Answer-focused Summarization Using Search Engines. *New Directions in Question Answering*.