Activities and findings:

Research and Education Activities:

A text collection such as a newswire archive or a Web crawl typically contains a great deal of repeated information. Different authors may each present versions of a story or event, the same event may get presented in different ways for different audiences, and the facts of an event may get recapitulated each time it is presented. Sometimes such presentations have little in common with each other; at other times one may be a copy of the other with minor edits. Given a topic of interest, then, a sufficiently extensive archive may contain much of the history of the topic. In particular, it can plausibly be used to identify when particular ideas or statements originated. It can also be used to check facts or the sources of statements made about the topic. Our interest has been in exploring whether we can identify alternative versions of the same information in order to reconstruct the information flow.

Over the course of the project, we have studied a number of different aspects of this problem, including comparing sentence similarity measures (Balasubramanian et al, 2007), detecting text reuse on a large scale using fingerprint techniques (Seo and Croft, 2008), finding text reuse using a web search engine (Bendersky and Croft, 2009; Chiu et al, 2010), and query transformation techniques for long queries (Bendersky and Croft, 2008; Bendersky et al, 2009; Dang and Croft, 2010; Huston and Croft, 2010). This research was focused on understanding reuse in real collections and developing core techniques to allow a user to specify a sentence-length query by indicating some statement in a web page, for example, transform that query into one or more queries that are effective at finding examples of text reuse, and refining that initial match through sentence-level comparison. We also studied some related topics such as blog search (Seo and Croft, 2008), due to the prevalence of reuse in blogs, and techniques for document representation (Seo and Croft, 2010).

Some of these activities were described in more detail in previous reports. In the rest of this section, we focus on describing work done in the past year.

This year we hired two new students. The focus of the work has been on developing larger-scale implementations of algorithms for text reuse detection, continuing to study the feasibility of text reuse detection on the web, developing algorithms for sentence extraction and comparison with a patent collection, and algorithms for extracting structure from longer queries. The following paragraphs describe the most important work:

Evaluating a Text Reuse Architecture for the Web:

Most of the research on text reuse detection was conducted on relatively small and homogenous datasets. The proposed methods are useful for information analysts who work on particular datasets which might be created by crawling specific web pages, but they are impractical for use

on the web directly due to the huge size of the web environment. Approaches to finding text reuse instances on the web have been recently suggested in work supported by this grant by Bendersky and Croft (2009). However, the practical applicability of a text reuse detection system for the web was not fully investigated.

In this work, we evaluated the accuracy and time efficiency of the text reuse architecture proposed by Bendersky and Croft (2009) to discover the advantages and drawbacks of such a system, and address the problems that arise in the web environment. As previously pointed out, computationally complex text reuse detection methods (e.g. sentence retrieval techniques), cannot be applied to the whole web. Thus, the architecture first creates an initial document set, and then applies the methods to this relatively small corpus. Choosing which documents to be retrieved initially, downloading these documents, and finally doing the sentence-level retrieval are labeled as Step 1, 2, and 3 respectively in Figure 1.

As Figure 1 illustrates, the input query is first passed to a "Query Formulator" module, which formulates various queries based on the input and acquires the resulting URLs via a search API. The purpose of this step is to build a small but rich initial document dataset. Then, the chosen URLs in the first step are downloaded from the web in Step 2. Finally, these documents are split into sentences and sentence retrieval methods are applied at the last step. The performance of the whole system initially depends on which documents are retrieved at Step 1, and finally the sentence retrieval method. Therefore, we investigated various approaches for both steps. Our findings are reported in the next section.

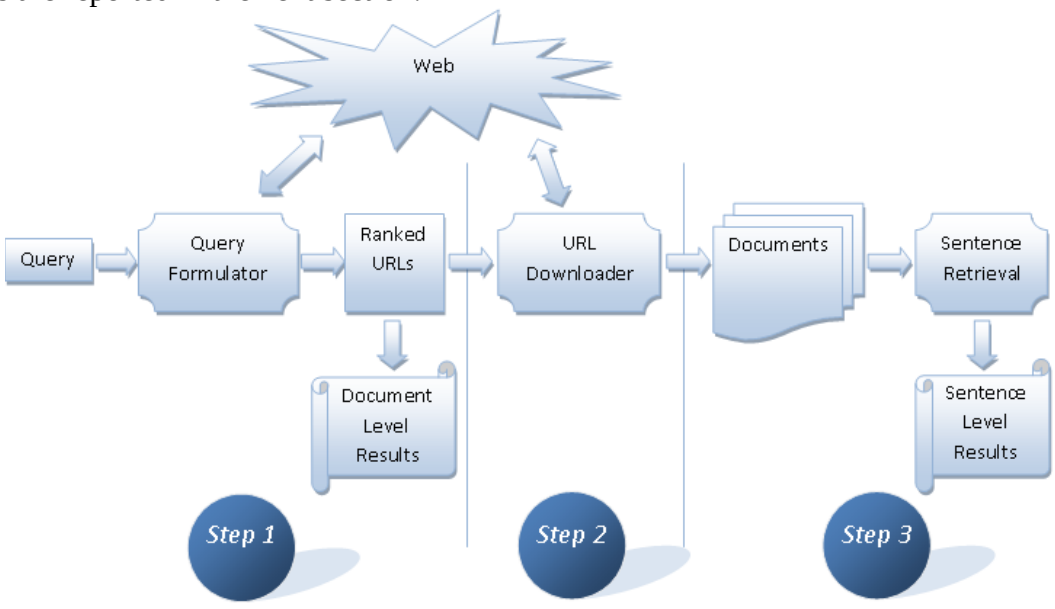

Figure 1: A Text Reuse Architecture for the Web

Evaluating Verbose Query Processing Techniques:

To further evaluate the text reuse techniques proposed by Bendersky and Croft, we focused on the methods used to reformulate long queries. These long queries would come from the user selecting a fact or statement in the text and looking for similar statements in the web.

The query transformation techniques we consider include stopword removal, phrase detection, key concept identification, and stop structure removal. Each of these techniques was applied individually and in combination to the verbose queries in our test sets. We measured the performance of the original queries and of the queries produced by these techniques and compare the results to evaluate the techniques' effectiveness. The most effective single technique found in this first round of experiments was stop structure removal.

A stop phrase has been defined as a phrase that does not provide any information on the user's information need. We define a stop structure as a stop phrase that begins at the first word in a query. Though it may be possible to effectively remove many stop structures from search engine queries using static lists, similar to those used for stopwords, doing so may inadvertently remove relevant words from the query. To address this problem, we used a sequential classifier that enabled us to automatically identify the stop structure in each query.

Two-stage Query Segmentation:

One of the important techniques for inferring the structure of long queries is query segmentation, which essentially means deciding how to break up the query into concepts. This is related to the key concept extraction work done earlier in the project, but is more general.

For natural language queries, a segmentation can be obtained by a shallow syntactic parser. For non-grammatical long queries, often encountered in web search (e.g., *new york times square*), supervised segmentation that leverages information from external sources such as query logs, web corpora and Wikipedia has been shown to be more effective. In fact, the segmentation techniques need not be mutually exclusive. We show that a supervised segmenter trained on an external data source can be applied as a second stage for a better segmentation of long noun phrases.

Query Reformulation using Anchor Text:

Text reuse involves finding statements that may have been rewritten to some extent over time. To address the issue of finding text that is similar in the sense of being a possible rewrite, we have been looking at query reformulation or rewriting techniques such as substitution, stemming, or expansion. Much of the previous work in this area has focused on using a large query log as a source of training data about rewrites, but we have shown that anchor text from a large collection can serve a similar purpose.

The techniques that we focus on generate new queries by substituting or adding words or phrases to the original query. For example, one technique works on the phrase level by looking at successive pairs of queries in user sessions. Two queries that are different from each other by

only one phrase are selected and the corresponding pairs of phrases are recorded as substitution candidates, which are used to generate substitutions for new queries. Another example of recent work on log-based query reformulation works on the word level. The method first extracts term associations based on their context distribution. For a new query, the method will decide whether to substitute a term with one of its "similar" words based on whether this new word matches the context of the query better than the original term.

In our work, we constructed a simulated query log or anchor log from the anchor text in a web test collection. We then evaluated the log-based query reformulation techniques using both the anchor log and a real query log and showed that the anchor log produces results that are at least as effective.

Ongoing work:

In the final period of funding for this project, we have also started two research efforts that we expect to finish and report in papers after this grant has finished. These are:

1. Text reuse detection algorithms based on MapReduce. MapReduce is becoming a dominant distributed processing paradigm for web-scale text mining so we believe it is very important to have text reuse detection algorithms specifically designed and evaluated in that environment.
2. Evaluating sentence extraction and comparison algorithms in a patent collection. We believe that patent search is potentially an important application for text reuse detection.

Findings:

In our initial plans for this grant, we proposed to design, implement, and evaluate algorithms for detecting text reuse on a web scale. We have made significant progress towards these goals, as described in the papers associated with the grant and the previous reports. In this final year, we have reached the stage of starting to evaluate a tool designed to find text reuse on the web, in addition to improving the algorithms for detecting reuse and comparing sentence –length texts.
In this section, we focus on reporting our most recent results related to the text reuse tool for the web. Results for the other work done this year (and previous years) can be found in the articles listed in the publications section.

The main results from the experiments using the web-based text reuse tool were as follows:

Table 1. Execution times (averaged over 25 queries) and NDCG@10 scores for the document-level retrieval methods. (*) indicates statistical significant difference (two-tailed t-test, $p<0.05$) between the marked method and Yahoo! (Unquoted) method which is used as the baseline.

Retrieval Method	Execution time (sec)	NDCG@10
Yahoo! (Unquoted)	<0.5	0.584
Yahoo! (Quoted)	<0.5	0.623
Iterative Chunking (IC)	7.62	0.466
Query 2-grams (Q2)	3.31	0.649
Query 4-grams (Q4)	3.26	**0.729***
Query 6-grams (Q6)	2.48	0.728*
Query 8-grams (Q8)	1.84	0.705*

Table 1 reports results for methods that break a text reuse query into "chunks" that are then submitted to a search engine to find related text. It shows that a method that breaks the query into 4-grams performs much better in terms of effectiveness (NDCG) than the iterative chunking method proposed by Bendersky and Croft and simply using quotes around the query and submitting it to the search engine (Yahoo in this case). There is a penalty in terms of time (multiple queries) but this should not be important for this application.

Table 2. Running times and NDCG@10 values for the different configurations of text reuse architecture. The configuration format is as follows: Doc-level Ret. Method + (# of docs downloaded at Step 2) + Sent-level Ret. Method. (*) is used as in Table 1.

Configuration	Step1 (sec)	Step2 (sec)	Step3 (sec)	Total (sec)	NDCG @10
IC+(~216)+WO	7.62	42.38	28.79	**54.22**	0.696*
IC+(~216)+MX	7.62	42.38	9.65	**59.65**	0.693*
Q4+(50)+MX	3.26	17.08	2.67	**23.01**	0.717*
Q4+(100)+MX	3.26	24.23	3.05	**30.54**	0.725*
Q4+(500)+MX	3.26	64.69	14.24	**82.19**	0.742*
Q4+(~842)+MX	3.26	111.04	18.27	**132.57**	0.754*

Table 2 reports results on overall effectiveness when sentence matching is included. It includes execution times and effectiveness fissures for the whole text reuse process including document retrieval and sentence matching (using a word overlap measure WO or the mixture model method MX proposed by Bendersky and Croft). These results show that there is a tradeoff between accuracy and speed, although the method that uses 4-grams to query, retrieves 500 documents, and uses the mixture model method to find sentences achieves good performance at a relatively low cost in terms of efficiency.

Overall, these experiments indicated that building a text reuse detection system for the web is feasible and effective enough to be useful.

In summary then, our research supported by this grant has resulted in advances in a number of areas related to text reuse detection and search involving sentence-length matching. We have

produced a number of papers in major conferences that are being cited by other researchers. Text reuse detection and retrieval has been established as feasible but the ultimate applications of these techniques are still unclear. We hope to do more work in the future to further advance our understanding of this area.