Compare&Contrast: Using the Web to Discover Comparable Cases for News Stories

Jiahui Liu, Earl Wagner and Larry Birnbaum
Northwestern University
Intelligent Information Laboratory
2133 Sheridan Road
Evanston, Illinois 60208 USA
{j-liu2, ewagner}@northwestern.edu, birnbaum@cs.northwestern.edu

ABSTRACT
Comparing and contrasting is an important strategy people employ to understand new situations and create solutions for new problems. Similar events can provide hints for problem solving, as well as larger contexts for understanding the specific circumstances of an event. Lessons can leaned from past experience, insights can be gained about the new situation from familiar examples, and trends can be discovered among similar events. As the largest knowledge base for human beings, the Web provides both an opportunity and a challenge to discover comparable cases in order to facilitate situation analysis and problem solving. In this paper, we present Compare&Contrast, a system that uses the Web to discover comparable cases for news stories, documents about similar situations but involving distinct entities. The system analyzes a news story given by the user and builds a model of the story. With the story model, the system dynamically discovers entities comparable to the main entity in the original story and uses these comparable entities as seeds to retrieve web pages about comparable cases. The system is domain independent, does not require any domain-specific knowledge engineering efforts, and deals with the complexity of unstructured text and noise on the web in a robust way. We evaluated the system with an experiment on a collection of news articles and a user study.

Categories and Subject Descriptors
H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval – *retrieval models, query formulation, information filtering.*

General Terms
Human Factors, Algorithms, Design.

Keywords
Intelligent information retrieval, comparable case, query formulation, knowledge discovery.

1. INTRODUCTION
In writing a news story, a reporter often compares the new event with other similar events to make it more familiar to readers, as well as to analyze any trends involving the new event. When considering specific business problems, it is common for executives to consult examples of other companies that solved similar problems to learn the strategies that were successful in analogous situations. Intelligence analysts also relate previous experience to current circumstances to provide support in building their arguments. These people all use *comparing and contrasting* as an important strategy for analyzing new situations or solving new problems. Information about similar events can provide hints for problem solving, as well as larger contexts for understanding specific circumstances. Lessons can be learned from past experience, insights can be gained about the new situation from familiar examples, and trends can be discovered among similar events.

Use of similar cases for analysis and problem solving has been studied by cognitive scientists [9, 18]. Based on the research, computer scientists in the area of Case-Based Reasoning (CBR) have developed various systems to provide support for specific tasks of human users based on structured "case-bases" [10, 15]. However, building a case-base requires considerable knowledge engineering efforts. Moreover, the application of the system is confined to the domains covered by the case-base. The end result is that classic CBR systems cannot handle new problems if the comparable cases are not contained in the case-base.

In contrast to the limitations of case-bases in CBR systems, the Web is abundant with information that is potentially useful for comparing and contrasting. A large amount of important and even sensitive information is now published on the Web, including government and business documents. Descriptions and discussions of events occurring around the world are available in different forms, such as news and blogs. The Web being the largest knowledge base for human beings provides an opportunity to discover comparable cases to facilitate situation analysis and problem solving. However, finding comparable cases on the Web is not easy for users. Currently, the primary mechanism for finding information on the Web is query-based. A user looking for comparable cases to a situation she is interested in must first search her memory for potential candidates and then translate the potential candidates into specific queries. A search engine (e.g. Google and Yahoo!) will find the web pages according to the specified key words. An important shortcoming of this approach is that it cannot find unexpected information, except accidentally, because the queries are derived from the users' knowledge space [24]. For example, a manager who is developing a business plan may search for other companies that she knows are faced with

WWW 2007, May 8–12, 2007, Banff, Alberta, Canada.
ACM 978-1-59593-654-7/07/0005.

Oracle tried to buy open-source MySQL

SAN FRANCISCO--Oracle tried to acquire open-source database maker MySQL, an indication of the profound changes the software giant is willing to make as it adapts to the increasingly significant collaborative programming philosophy.

MySQL Chief Executive Marten Mickos confirmed the acquisition attempt in an interview at the Open Source Business Conference here but wouldn't provide details such as when the approach was made or how much money Oracle offered.

…

O'Grady said Oracle could benefit from MySQL in the way that IBM has from its acquisition of Gluecode, a company that commercializes the open-source Geronimo Java application server software and competed with IBM's own proprietary WebSphere product. IBM now offers Gluecode's software as a free product called WebSphere community edition.

…

from CNET News.com

Figure 1. News Article about Oracle

similar problems. However, there may be many other cases that the manager is not aware of. These unknown cases may be more valuable for the manager, because they expand her knowledge space and may inspire new ideas for handling the situation.

In this paper we propose a novel system for using the web to discover comparable cases for the news story a user is reading. The system accepts as input the URL of a news web page. It then analyzes the news story and builds a model of the reported situation, which serves as the basis for formulating queries and selecting comparable cases. To do this, the system identifies the *generic situation keywords*, terms and phrases describing the situation of the story, and the *main entity*, the person, place, or organization that the story is about. The system then dynamically discovers *comparable entities*, entities involved in similar situation as the main entity, based on word contexts similarity. Each comparable entity forms the nucleus of a comparable case. The system formulates queries to general web search engines by combining the comparable entities and the generic situation keywords to retrieve web pages about comparable cases involving the comparable entities. Unlike traditional CBR systems, our system is domain independent, does not require any domain-specific knowledge engineering, and employs robust mechanisms to deal with the complexity of unstructured text and the noise on the Web. The web pages the system finds can help human users to analyze the events of their interests or solve specific problems. They may also be used as feeds to be further processed for automatically or semi-automatically building case-bases for machine reasoning.

In the remainder of the paper, we analyze the problem of discovering comparable cases and summarize our proposed approach for tackling the problem in Section 2. In Section 3, we review related research in the areas of information management

IBM Expands Paid Open Source Strategy

UPDATED: IBM (Quote) is making a bid on professional open source with the acquisition of privately held Gluecode, officials announced Tuesday.

Officials did not discuss financial and operational details of the merger, the first acquisition made by Big Blue of an open source company.

Gluecode's operations will be assimilated into IBM's software group and expand the company's WebSphere application integration middleware product line.

Officials plan to offer customers and business partners Gluecode's application server software and sell software and support services on top of the offering, as well as let customers upgrade to IBM WebSphere products.

…

from internetnews.com

Figure 2. News Article about IBM

assistants, textual case-based reasoning and topic detection and tracking. We describe details of the implementation of the system in Section 4. Two experiments aimed at evaluating the performance of the system are presented in Section 5. Finally, we conclude with future work.

2. THE PROBLEM AND OVERVIEW OF THE PROPOSED SOLUTION

Figure 1 and Figure 2 illustrate excerpts from two online news stories. One story describes Oracle's intention to acquire open-source database maker MySQL. The other story describes the acquisition made by IBM of open-source firm Gluecode. The IBM acquisition is referred to in the Oracle story as a comparable case for the news event. Close analysis of the two stories reveal some similarities and differences of the two cases on an abstract level. As a comparable case, the *generic situation* described in the IBM article is similar to that of the Oracle article. For example, the basic events are of the same type, i.e. acquisition. The strategies employed by these two companies are similar, i.e. adapting to the open-source paradigm. The differences between these two articles are in the specific details, including the actors in the events, the products of the companies, and the people mentioned in the news. These details correspond to the various entities in the events, such as people, organizations, locations, etc. Applying this observation, we define the task of finding a comparable case as finding a case with a similar generic situation but involving different entities.

The question then arises of how to identify and represent the generic situation and the entities involved. In describing an event, a reporter seeks to answer the five W questions: *who, what, where, when and why*. We note that the *who, where and when* of a news account are named entities and can be easily be recognized using conventional approaches Actions and relationships among the actors, on the other hand, appear as non-named entity terms and give information about *what* and *why*, which constitute the generic situation

Based on this insight, we propose an approach for finding comparable cases by using the named entities and the non-named entity terms differently in modeling the story and retrieving information. The situation described in the original news story is analyzed and represented using an extended bag of words model, with named entities and non-named entity terms separated. For a given news story, the system creates two vector representations: one vector consists of all of the named entities, and the other consists of all of the non-named entity terms. These two vectors represent the case specific details and the generic situation of the original story, respectively. In terms of our theory of comparable cases, documents about comparable cases should contain similar non-named entity terms as the original story, but have different named entities.

However, it is not necessary to require all entities in comparable cases to be different from the original news story. We note that typically only a few entities make up the focus of a news story. The study by Zhang et al. [25] provides support for this intuition. Their investigation into "focused entities", which they define as the entities most relevant to the main topic of a news article, showed that there is a high level of agreement on "focused entities" among human readers. In our system, we developed a mechanism to extract the most important entity of the news story, namely the *main entity*, and associate the story with the main entity. For example, in the Oracle story, the main entity is "Oracle". Accordingly, each comparable case should have a *comparable entity* (i.e. "IBM") to the *main entity* which will be used as seed for retrieving information about the case.

In order to find comparable cases from the Web, we use the web search engine, Google [8]. The system selects the top non-named entity terms and phrases as the *generic situation keywords* (i.e. "open source", "software" and "acquisition") to query for relevant documents. If we know a priori the comparable entities for the comparable cases, we could simply formulate a query by combining names of the comparable entities and the generic situation keywords. One possible source of the comparable entities may be a static hierarchy of named entities, such as a directory of companies organized by industries, within which the companies in the same business are treated as comparable entities. However, whether two entities are comparable is dependent upon the context of the situation, not just by their static similarities and distinctions. For instance, given a news story about job cuts at an insurance company through compulsory redundancies, stories about telegram companies with job cuts through compulsory redundancies and outsourcing is informative for the user, even if the companies are not in the same industry.

Instead of using a static hierarchy of entities, we developed a mechanism for dynamically discovering comparable entities from the Web. The basic idea for the method is that a comparable

Figure 3. A Screen Shot of Compare&Contrast

entity should share a similar word context as the main entity in the original news story since they are involved in similar situations. The details of comparable entity discovery will be discussed in Section 4. After the system finds the comparable entities, the system formulates a query for each comparable entity and retrieves the web pages about the comparable cases involving the entity. The retrieved web pages are organized under the comparable entities and presented to the user.

The prototype of Compare&Contrast is currently implemented as a server-based system. The user provides the URL of the news page she is interested in, and the system will find the comparable cases and present them to the user. Figure 3 shows a screen shot of the system. The left frame presents the original news; the right frame lists the relevant web pages with their titles and summaries from Google. Presenting the original news and the comparable cases side by side helps the users compare and contrast the stories. Furthermore, organizing the documents under the comparable entities helps the user browse the search results. Instead of browsing through the retrieved documents, the user just needs to judge the relevance of the comparable cases by the comparable entities and the listed summaries, and investigate the web pages if she is interested.

3. RELATED WORK

Some researchers in Artificial Intelligence (AI) and Information Retrieval (IR) have studied the use of modeling users' task related documents to automatically retrieve useful information for the user [3, 4, 6, 17]. Watson [3, 4], for example, analyzes opened text documents, such as web pages that the user is browsing and word documents that the user is authoring, formulates queries to traditional IR systems (i.e. web search engines and databases), and provides just-in-time information to the user. This kind of system reduces the effort of querying and filtering by bringing the information to users as they need it. However, these systems identify relevant web pages mainly based on document similarity. As pointed out by Budzik et al [4] and Rhodes and Maes [17], for some tasks, similar documents may not be useful documents. Given a news page, most of the documents Watson finds would be information about the same story. But for many tasks, the user would be interested in related but different stories. Compare&Contrast is a system that finds those stories and brings them to the user even when the user is not aware of them.

With regard to retrieving comparable cases, Textual Case-Based Reasoning (TCBR), is a subfield of CBR that aims to use the textual knowledge sources in an automated or semi-automated way for supporting problem solving through case comparison [21]. Adapted from Information Retrieval and other text-oriented techniques, TCBR researchers developed various intelligent approaches to indexing and retrieving textual comparable cases [2, 12, 16].

One prominent work in TCBR is SMart Index Learner (SMIL) in the domain of law [1, 2]. In SMIL, Brüninghaus and Ashley noted that case specific names are unhelpful or even detrimental for case retrieval. Their system specifically addresses the issue through information extraction (IE) techniques. SMIL utilizes extraction rules and domain specific heuristics to replace the names of the parties and product-related information by their role in lawsuits. Coupled with other IE techniques for extracting actions and ascertaining negations, SMIL is able to generate abstract representations of textual legal cases, which are then used for classifying and indexing the cases. SMIL demonstrates the use of an IE tool in textual case-based reasoning. However, the system is domain-dependent, requires a certain level of knowledge engineering effort and is designed for situations in which all documents have similar content.

SOPHIA CBR developed by Patterson et al. [16] is a domain-independent system that automatically discovers cases and similar knowledge based on a contextual document clustering approach. It intelligently discovers important groups of words that appear in similar documents, and clusters semantically similar cases accordingly. This method is designed for retrieving cases and discovering knowledge in a given text collection, which may not be feasible for processing the documents on the Web. However, their algorithm for discovering the important terms representing topics can be adopted to enhance the query formulation techniques for discovering comparable cases on the Web.

Unlike the TCBR systems that retrieve cases for machine reasoning, Compare&Contrast is aimed at finding comparable cases for human readers. Therefore, less emphasis is put on knowledge generation from texts. Moreover, the system finds comparable cases from the entire Web, which is much broader and noisier than typical text collections. Different methods must be developed to address the challenges presented by the Web.

A research area related to characterizing the similarities and differences between news events is the New Event Detection (NED) task within Topic Detection and Tracking (TDT). NED addresses the issue of detecting the first story about a new topic within a stream of news. To improve the performance of NED systems, researchers have explored using named entities to modify the document representation of news articles [11, 23]. Kumaran and Allan [11] introduced multiple document models for news stories, with one vector representation consisting of all the terms in the document, one consisting of all the named entities and one consisting of all the non-named entity terms. They observed that some categories of news stories were better tackled using only named entities, while using only non-named entity terms helped for others. Our document representation is similar to that of Kumaran and Allan, but the goal of our system is to find comparable stories, rather than new stories. Furthermore, rather than passively detecting a piece of useful text from a stream of news, Compare&Contrast actively finds useful information from the Web, which involves different processes like querying and filtering.

4. IMPLEMENTATION OF COMPARE&CONTRAST

The goal of Compare&Contrast is to find related but different comparable cases for news stories. The architecture of the system is displayed in Figure 4. When a news web page is given to the system, the News Modeler processes the web page and creates a model of the news story with two vector representations. The main entity and generic situation keywords are determined from these two vectors respectively. In order to discover comparable entities, the Potential Page Collector formulates queries to the Web Search Engine, retrieves the *potentially relevant pages*, and filters out some irrelevant pages. The potentially relevant pages are fed into the Comparable Entity Identifier, which identifies and selects the comparable entities according to the similarity of the word contexts. The Potential Page Collector and Comparable Entity Identifier make up the Comparable Entity Discovery

component. Finally, the comparable entities are combined with the generic situation keywords to retrieve web pages about comparable cases. In the following sections, we describe our approach to modeling news stories and discovering comparable entities. We also describe the approach developed for pre-filtering out some irrelevant pages to improve the performance of the system on the Web

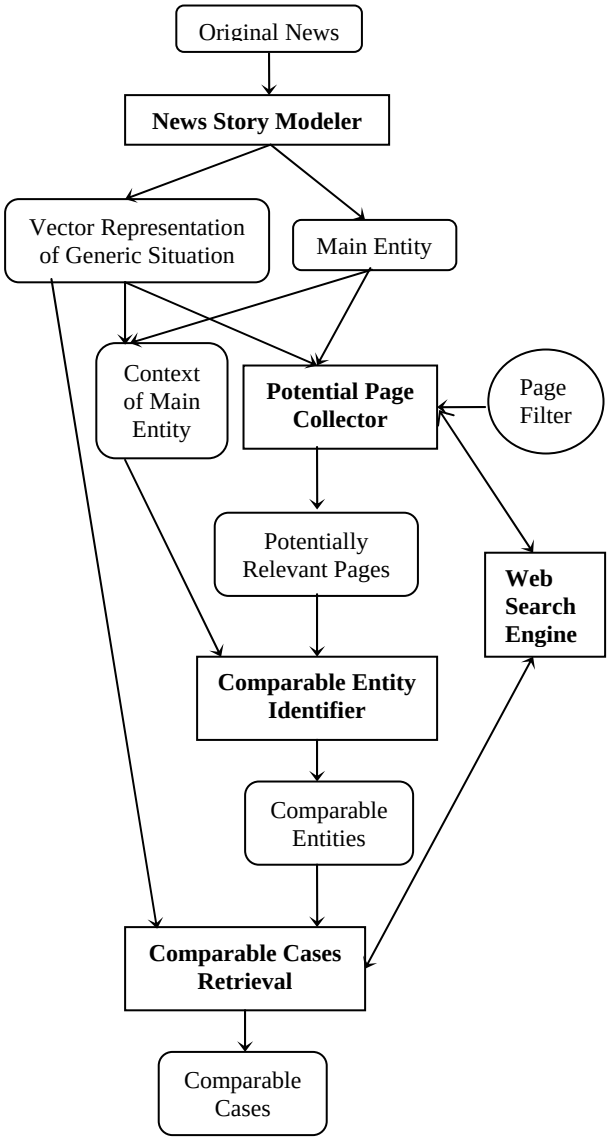

Figure 4. Architecture of Compare&Contrast

4.1 News Story Modeling

The central idea for our approach is that a comparable case should be about a similar situation involving different entities. To provide support for formulating queries and discovering comparable entities, the News Story Modeler separately models the generic situation and the case specific details with the non-named entity terms and named entities, respectively.

When the URL of a news web page is sent to Compare&Contrast, the system retrieves the web page, extracts the news content from the web page, splits the sentences and tags the named entities. We implement the algorithm similar to that of Ma et al. [14] to extract the most important block of a web page, which is usually the body of the news in the case of a news page. The title of the web page is used as the title of the news story. For named entity recognition, the system uses the web service provided by ClearForest Semantic Web Services (SWS) [5], adopting its tags of *person*, *organization*, *company*, *product* and *geographical location*. In addition, the system tags *nationality* with a gazetteer list and the rest of capitalized word sequences as *unknown*. After the entities are tagged, named entities and non-named entity terms are processed separately.

For the non-named entity terms, stop words are removed and the rest of the terms are stemmed with a Porter Stemmer [20]. To create a vector representation of the non-named entity terms, we used a modified TF-IDF model which incrementally decreases the importance of terms appearing later in the news article. It is well known that in news text the most important information is given in the top; some news articles are even written to enable truncation [13, 19]. To implement this idea, we assign scores to sentences according to their position. The score of $sentence_i$ is calculated using Equation 1, where *numSentence* is the total number of sentences in the news article.

$$score(sentence_i) = 2 - i / numSentence \quad (1)$$

Accordingly, when computing the term frequency (TF) for the non-named entity terms, each occurrence of a term is given the score of the sentence it appears in, rather than being counted evenly. Moreover, the TF of terms in the title or the lead sentence is doubled (the first sentence in the news article is treated as the lead sentence). The IDF of terms is computed using an archive of 343,187 news stories that we collected from April 2004 to June 2006.

As we are working on the story modeling, we found that it would be beneficial to capture the important word groups in event descriptions, such as “open source” or “nuclear test”. Therefore, the system treats the stemmed bigrams which appear more than three times in the article as phrases. The TF of a phrase is computed in the same way as a unigram. The IDF of a phrase is the maximum of the IDFs of the two terms of the phrase. Although this is a simple technique, it improves the representation of the news story, because for many phrases, the meaning of the word group is much different from the individual words. During query formulation, the words in a phrase are grouped together, resulting in more meaningful queries and produces better query results.

Unlike the non-named entity terms, the vector representation of named entities only uses TF. The TF for named entities is computed the same way as non-named entity terms, assigning greater weights to named entities appearing in the top of the news. The named entity with the highest score is chosen as the main entity.

However, a tricky issue in counting named entities is that different references to the same entity should be grouped together. In writing news stories, journalists usually give the full name of the entity at the first mention, but use some shortened form later. SWS [5] provides some support for coreference resolution, which we supplement with the following procedure. For two entities of the same type, if the tokens of one named entity are totally contained within the other, or the name of one entity is the abbreviation of the other, the two entities are treated as

coreference. Some of the unknown entities are also unified in the same way with named entities with specific types. In addition, nationality is treated as support for the country it belongs to. Though not perfect, this algorithm is effective because the primary goal here is to identify the main entity, and it works satisfactorily for this purpose.

4.2 Comparable Entity Discovery

As discussed in Section 2, Compare&Contrast dynamically discovers entities comparable to the main entity from the Web. Based on the story representation produced by the News Story Modeler, generic situation keywords are selected from the vector of non-named entity terms. According to our theory of comparable cases, the relevant web pages about comparable cases should contain the generic situation keywords, but not the main entity. Therefore, the system tries to retrieve a set of *potentially relevant pages* using the query "-'main entity' {generic situation keywords}" (i.e. "-'Oracle' 'open source' software acquisition"). The comparable entities are identified and selected from these pages.

According to our definition of comparable entities, the comparable entities should be involved in the similar generic situation as the main entity of the original news. Therefore, within a document of comparable cases, the text describing and analyzing the characteristics, actions and events of the comparable entities should be similar to the text about the main entity. Based on this idea, we define the *word context* of a named entity as the terms and phrases co-occurring with the entity within the same sentence. A *word context vector* is built for the main entity in the original news article by collecting all the terms and phrases co-occurring with the main entity. The weight of the context terms is borrowed from the original vector representation of generic situation produced by the News Story Modeler. So the word context vector bears information about the word context of the main entity, as well as the document-level information about the importance of the terms in the whole news story.

The potentially relevant web pages are preprocessed in the same way as described in the previous subsection, including content extraction, sentence splitting, named entity recognition, tokenization, and stop word removal. To compute the similarity of word context, each sentence in the potentially relevant web pages is scored using the word context vector (Equation 2). In the equation, *S* is the set of all the terms in the sentence; *C* is the set of all the terms in the word context vector.

$$simScore(sentence) = \frac{|S \cap C|}{|S|} \times \frac{\sum_{term \in (S \cap C)} weight_{term}}{\sum_{term \in C} weight_{term}} \quad (2)$$

Entities of the same type as the main entity in a potentially relevant page are considered as candidates for comparable entities. The similarity score of an entity is computed using the score of all the sentences they appear in (Equation 3). In the equation, *E* is the set of sentences which containing the name entity and *numSentence* is the total number of sentences in the article.

$$simScore(entity) = \frac{1}{numSentence} \times \sum_{sentence \in E} simScore(sentence) \quad (3)$$

We normalize the similarity score by the total number of sentences in the article, so the score of entities in different articles can be compared in a meaningful way.

After this process, each potentially relevant page has a set of candidates for comparable entities with their *simScore*s. However, we observed that among these potentially relevant pages, there are usually some web pages describing the same events. It would be beneficial to cluster the articles about the same events together. Influenced by the study by Gabrilovich et al. [7], which suggests that the metric counting named entities can be an effective mechanism in detecting new stories, we developed our method for clustering articles according to the overlap of the important entities in the articles. *simScores* of the same named entities within a cluster are added together. The named entity with the highest score within the cluster is identified as the comparable entity for the cluster.

In summary, comparable entity discovery takes 4 steps:

1. Retrieve potentially relevant pages, which are web pages with the generic situation keywords but without the main entity;

2. Construct the word context vector of the main entity from the original news story;

3. Compute the *simScore* of the entities in the potentially relevant pages according to word context similarity;

4. Cluster the web pages describing the same events and grouping the comparable entities;

The first three steps are similar to the Local Context Analysis (LCA) technique proposed by Xu and Croft [22] for query expansion. LCA selects expansion terms based on co-occurrence with the query terms within the initial set of top-ranked documents. However, for comparable entity discovery, we use the context vector of the main entity in our analysis, which is much richer then the query terms, as well as more closely related to the main entity. Furthermore, our system measures co-occurrence at the sentence level to capture the similarities between the descriptions of the entities.

After Compare&Contrast identifies the comparable entities, the system uses the comparable entities as seeds to retrieve comparable cases with query "+'comparable entity' –'main entity' {generic situation keywords}" (i.e. "+'IBM' –'Oracle' 'open source' software acquisition"). To verify the comparable cases, the system uses the search results counts returned by web search engines to calculate the relevant score of comparable cases according to Equation 4, where *entity* is the comparable entity involved in the comparable case.

$$relScore(case) = simScore(entity) \times \log_{10} queryCount \quad (4)$$

The benefit of taking into account the search result count is twofold. Firstly, more hits on the Web of comparable cases means that the cases have larger coverage in public. They should naturally be ranked as more important than others with lower hits. Secondly, the process in analyzing web pages, such as named entity recognition and content extraction, are subject to mistakes and the system may produce false comparable entities. However, there are usually very few web pages describing the false comparable entities with the generic situation keywords. Therefore, combining the similarity score and search result count eliminates some noise.

4.3 Page Filtering to Remedy Noise on the Web

To maximize the coverage of the system, Compare&Contrast uses the general web search engine. This choice also brings the system a lot of noise in the search results. Within the set of potentially relevant pages, there are some irrelevant web pages that are useless or even detrimental for selecting comparable entities. We identified two different categories of harmful pages that impact the performance of Compare&Contrast and developed filters accordingly.

The first category of harmful pages is directory pages, such as news website portals and lists of products. Rather than containing one consistent block of content text, this kind of page contains information about multiple entries with each entry being relatively short. Our content extraction algorithm usually mixes the various entries together. Using directory pages to find comparable entities produces false results because of the mixture of contexts. We observed that the directory pages often contain a lot of headlines or sub headlines with capitalized words. Therefore, the percentage of upper case characters on directory pages is often higher then other pages, which can be a good feature for filtering out the pages. We randomly selected 102 news articles from our news archive and 40 directory pages from the Web and calculated the percentage of upper case characters in the content of these pages. Figure 5 shows the distributions of the percentages for the two types of web pages. The percentages of the directory pages are apparently higher than those of news pages. According to the result of this experiment, the system filters out web pages with more than 28% of the characters in upper case.

The second category of harmful pages is irrelevant pages. Although the query contains the generic situation keywords, the situations described in some pages are very different from those of the original news story and it is unlikely that any comparable entities can be identified. The system handles the irrelevant web pages in a manner similar to that used by human users. As human users view the results returned by web search engines, they glimpse at the summaries and judge the relevance before they retrieve the web page for in-depth reading. We developed a web page filter similar to this process. The summaries of results returned by web search engines are compared with the vector representation of the generic situation. The relevance of a result is measured according to Equation 5, where *S* is the set of terms appearing in the summary and *V* is the set of terms in the generic situation vector.

$$relScore(result) = \frac{\sum_{term \in (s \cap v)} weight_{term}}{\sum_{term \in v} weight_{term}} \quad (5)$$

Like human users, the system does not bother retrieving and processing the web page if the *relScore* of the search result is below certain threshold.

These two filters are built specifically for dealing with noise on the Web. They are executed before the Comparable Entity Identifier. Our experience shows that multi-step procedure with different levels of sophistication is an effective way of processing content on the Web.

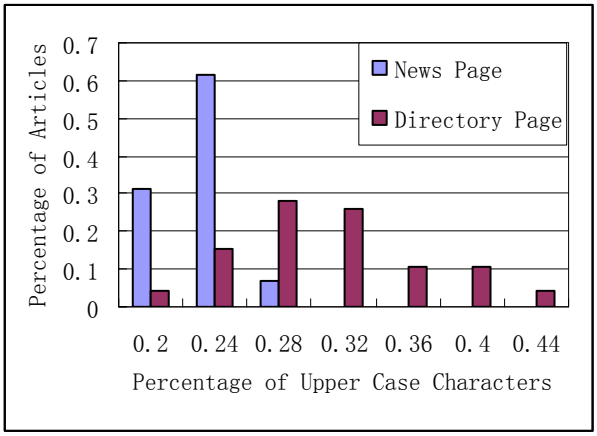

Figure 5. Distribution of the Percentage of Upper Case Characters in Two Types of Web Pages

5. EVALUATION

To evaluate the performance of our system, we needed a collection of news articles for which comparable cases can be found on the Web. Not all news articles suggest comparable cases. Some news articles describe and discuss general phenomena, rather than specific events and, as a result, no focused entities can be detected from this kind of news. These articles do not fit into our definition of comparable cases which should share similar situation but involve different entities. On the other hand, even for some news articles about some specific events, the events may be too odd that no comparable cases exist or there is very little information about the comparable cases on the Web. Testing Compare&Contrast on this kind of news is also unproductive.

In our effort to build an appropriate news collection, we noticed that there is a moderate portion of news articles that contain comparisons or contrasts *inside* the articles. The Oracle story in Figure 1 shows an example of this kind of news article. These news articles can be good candidates for our test cases. Moreover, the comparable cases mentioned in these news articles can be used as answer keys for evaluation.

With this observation in mind, we built a collection of test cases by gathering news articles mentioning comparable cases. For the preliminary evaluation, we collected 40 news articles from various news websites, including MSNBC, BBC NEWS, CNN.com, Yahoo! NEWS and CNET News.com. These articles cover a wide range of topics. According to their original categorization on the news websites, we divided the test cases into three categories: politics, business, and technology. In the preparation of the test cases, the texts about the comparable cases of the main story are extracted from the content and set aside as answer keys, as shown in Figure 6. Therefore, Compare&Contrast only uses the text for describing the main story to find comparable cases for the story.

We conducted two experiments on the collection. First, we ran Compare&Contrast on all the test cases and use the comparable cases given in the articles to evaluate the effectiveness of the technique for discovering comparable entities. Second, we randomly selected 6 test cases from the collection and invited 5 people to judge whether the web pages Compare&Contrast found are about relevant cases comparable to the original news stories.

<article>

<title> Germany tries 'Holocaust denier' </title>

<url> http://news.bbc.co.uk/2/hi/europe/6147400.stm </url>

<content> A German man deported from the US has gone on trial in the Germany city of Mannheim for alleged Holocaust denial.

Germar Rudolf published a study saying the Nazis did not use gas to kill Jews at the Auschwitz concentration camp.

The prosecution says he "represented the Holocaust as invention" and used the internet to spread his documents.

If found guilty, Mr Rudolf will face up to five years in prison. He has already been given an jail sentence in a similar case but fled to the US.

A chemistry graduate, 42-year-old Mr Rudolf also faces charges of defaming the memory of the dead.

He was sentenced to 14 months in prison in a similar case in 1995 but fled the country.

His 2000 application for political asylum in the US was rejected and he was deported back to Germany to serve the earlier sentence.</content>

<comparison> In a similar case in February 2005, British revisionist historian David Irving was found guilty of denying the Holocaust by an Austrian court and sentenced to three years in prison. </comparison>

</article>

Figure 6. Exemplar Test Case

5.1 Effectiveness of Comparable Entity Discovery

Compare&Contrast works by first dynamically discovering comparable entities from the Web and then using the comparable entities to retrieve relevant web pages about comparable cases. Therefore, comparable entity discovery is a critical step for the system. To evaluate the effectiveness of our proposed technique for finding comparable entities, we used the comparable cases mentioned in the test cases as answer keys. The 40 news articles are fed into the Compare&Contrast. For each news article, the system returned its top five, or fewer, comparable entities with their score above certain threshold. If some of the comparable entities are mentioned by the comparison part of the test case, the test case is counted as a hit. Out of the 40 test cases, the system has found comparable entities mentioned in 23 test cases. The overall recall is 57.5%. Table 1 reports the performance of Compare&Contrast on the three categories of news.

Table 1 Performance of Compare&Contrast

	Articles	Hits	Recall
Politics	15	9	60%
Business	13	8	62%
Technology	12	6	50%

Though for 17 test cases Compare&Contrast didn't find the comparable entities mentioned in the comparison part of the original article, this doesn't mean that the system found no relevant comparable entities for those cases. As it can be shown in the second experiment, many of the comparable entities found by the system are valid and useful for retrieving relevant comparable cases. This is especially true for technology news. For example, some technology articles report on companies introducing new products to market. Compare&Contrast found many similar companies and similar products in the same market. The comparable entities mentioned in the comparison part of the original articles were simply not ranked highly enough to be in the top five comparable entities.

Analyzing times in which the system clearly failed revealed some problems in discovering comparable entities. Firstly, for some of the failed cases, the system found very few or even not any comparable entities. This is a sign that the query used to retrieve the potentially relevant pages is not good enough. The mechanism of query formulation can be improved to intelligently detect this kind of failure and modify the queries accordingly. Secondly, some comparable cases mentioned in the article contain some of the entities involved in the original story. For example, a comparable case for the acquisition made by American Express on Harbor Payments is the investment of Oak Investment Partners on Harbor Payments. However, the system does not take into account entities other than the main entity, either in query formulation or comparable entity identification. However, the comparable cases involving some of the same entities as the original story may be more closely relevant. To capture this dimension of comparability, the system needs to make more sophisticated use of the named entities in the original article.

5.2 Relevance of Retrieved Pages

After the comparable entities are identified and selected, Compare&Contrast constructs a query for each comparable entity to find web pages about the comparable case involving the comparable entity. To evaluate the relevance of the web pages the system found, we randomly selected 6 test cases from the collection and invited 5 people to judge the relevance of the retrieved web pages of the test cases. The 5 users consisted of 2 graduate students, 2 staff members and 1 undergraduate from our department. For the 6 test case, 4 are hit cases, within which Compare&Contrast found the comparable entities mentioned in the original article, and the other 2 are failures. For each test case, there were 5 comparable entities and the system presented 3 or fewer web pages retrieved for each comparable entity. Altogether there were 85 web pages. For each web page, the users were asked whether the web page contains a relevant comparable case for the original news story. The web page is given 1 point if one user answers yes and 0 point if no, so by summing the judgment of all users, each web page would get between 0 and 5 points. Table 2 shows the distribution of the scores of the web pages. The average score for all the 85 web pages is 3.13. We consider a web page with score equal or higher than 3 to be relevant, because it was judged so by a majority of the users. 59 of the 85 web pages contains relevant comparable cases, thus the precision is 69.4%.

Table 2 Relevance of Web Pages

Score	0	1	2	3	4	5
Num of pages	10	10	6	14	23	22

We also computed a score for each comparable entity by averaging the score of the web pages associated with the entity. Figure 7 shows the distribution of the score of the entities. It should be noticed that the distribution of the score of the entities takes the form of a U-curve. That is because that the relevance of web page is highly dependent on the relevance of the entity. If the entity is indeed a comparable entity involved in some comparable cases to the original story, most of the web pages retrieved according to the entity are relevant pages. Specifically, the score of the 4 comparable entities mentioned in the comparison part of the 4 hit test cases are: 4, 4.7, 4.7 and 5.

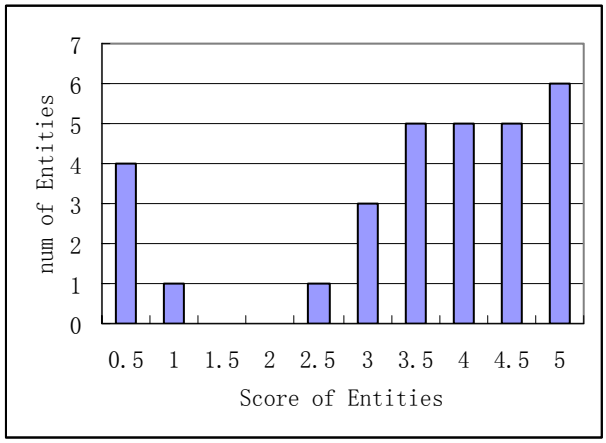

Figure 7. Distribution of Score of Entities

The dependency between the comparable cases and the comparable entities illustrates that the score of the entities computed above can be used as a measurement of the usefulness of the comparable entity. If we consider the entities with score equal to or higher than 3 as correctly identified comparable entities, there are 24 correctly identified comparable entities out of 30 entities, which is 80%. It should be noticed that although Compare&Contrast didn't find the comparable entities mentioned in the 2 failure cases, there are 3 and 2 correctly identified comparable entities in the two cases, which are the comparable cases out of the knowledge space of the reporters.

6. CONCLUSION AND FUTURE WORK

In this paper, we analyzed the problem of finding comparable cases for news stories, characterizing comparable cases as those that share a similar situation as the original story but involve different entities. Based on this idea, we presented Compare&Contrast: a system for finding the comparable cases by automatically formulating queries based on the story model derived from the original article and dynamically discovering comparable entities involved in comparable cases. Specific techniques are implemented in Compare&Contrast to deal with the complexity of natural language texts and the noise of the Web. The system does not require any domain-specific knowledge engineering. The web pages the system finds can be used by human researchers to gain understanding of the events they are interested in and support their decision-making processes. The system may also supply documents to be further processed for automatically or semi-automatically building case-bases for case-based reasoning systems. We evaluated the system with an experiment on a collection of news articles and a user study.

The research can be extended in several directions. As indicated by our evaluation, we plan to investigate a more sophisticated use of named entities so that the system will be able to find and identify comparable cases within more narrowly constrained contexts, such as comparable cases in the same country or involving some of the same actors. We also plan to develop more intelligent query formulation mechanism so that the system can dynamically change the queries depending on analysis of the results. Finally, in our preparation for test case, we found that there are many news articles that provide some kind of comparison and contrast. The current system should be extended to make use of these existing analyses, so that more interesting results can be found and presented to the user.

Compare&Contrast is part of our larger goal of exploring intelligent information retrieval to support the user by finding useful, not just similar, information. Underlying this effort is the view that documents with useful information should be similar to the user's current document in certain aspects, but systematically different from the original document in certain other meaningful aspects. In addition to finding comparable cases, we are also planning to explore other dimensions of systematic differences in our future work.

7. ACKNOWLEDGMENTS

This research was supported in part by the National Science Foundation under grant no. IIS-0325315/004. We thank our colleagues at our InfoLab at Northwestern, especially Sara Owsley, Sanjay Sood, Nathan Nichols, and Kris Hammond. We also thank Eleftherios Gdoutos for building the test case collection for evaluation. Finally, we are grateful to the volunteers who participated in the user study.

8. REFERENCES

[1] Brüninghaus, S. and Ashley, K. D. 2006. Progress in Textual Case-Based Reasoning Predicting the Outcome of Legal Cases from Text. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*.

[2] Brüninghaus, S. and Ashley, K. 2001. The role of Information Extraction for Textual CBR. In *Proceedings of the 4th International Conference on Case-Based Reasoning*, LNCS 2080, Springer.

[3] Budzik, J. and Hammond, K. J. 2000 User Interactions with Everyday Applications as Context for Just-in-time Information Access. In *Proceedings of Intelligent User Interfaces*.

[4] Budzik, J., Hammond K., and Birnbaum, L. 2001. Information access in context. *Knowledge based systems* 14 (1-2), pp 37-53, Elsevier Science.

[5] ClearForest Semantic Web Services (SWS) http://sws.clearforest.com/.

[6] Finkelstein, L., Gabriolovic, E., Matias, Y., Rivilin, E., Solan, Z., Wolfman, G., and Ruppin, E. 2001. Placing search in context: The concept revisited. In *Proceedings of the 10th International World Wide Web Conference*, Hong Kong, China.

[7] Gabrilovich, E., Dumais, S. and Horvitz E. 2004 Newsjunkie: Providing personalized newsfeeds via analysis

of information novelty. In *Proceedings of the 13th International World Wide Web Conference.*

[8] Google http://www.google.com/.

[9] Hammond, K. J. 1990 Case-based planning: A framework for planning from experience. *Cognitive Science*, 14(3):385–443.

[10] Hinkle, D. and Toomey, C. 1995. Applying case-based reasoning to manufacturing. *AI Magazine*.

[11] Kumaran, G. and Allan, J. 2004. Text classification and named entities for new event detection. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, New York, NY, USA.

[12] Lenz, M. and Burkhard H. D. 1997. CBR for Document Retrieval - The FAllQ Project. In *Proceedings of the 2nd International Conference on Case-Based Reasoning Research and Development*..

[13] Lin C-Y and Hovy E. 1997. Identify Topics by Position. In *Proceedings of the 5th Conference on Applied Natural Language Processing*.

[14] Ma, L., Goharian, N. and Chowdhury, A. 2003. Automatic Data Extraction From Template Generated Web Pages In *Proceedings of International Conference on Parallel and Distributed Processing Techniques and Applications* (PDPTA 03), Las Vegas, Nevada.

[15] Morgan, A. P., Cafeo, J. A., Gibbons, D. I.. Lesperance, R. M., Sengir, G. H. and Simon, A. M. 2003. The General Motors Variation-Reduction Adviser: Evolution of a CBR System. In *Proceedings of 5th International Conference on Case-Based Reasoning*, Trondheim, Norway.

[16] Patterson, D., Dobrynin, V., Galushka, M. and Rooney, N. 2005. SOPHIA: A Novel Approach for Textual Case Based Reasoning. In *Proceedings of the 18th International Joint Conference on Artificial Intelligence.*

[17] Rhodes B. J. and Maes, P. 2000. Just-in-time information retrieval agents. *IBM System Journal* 39(4): 685-704.

[18] Schank, R. 1982. *Dynamic Memory: A Theory of Learning in Computers and People*. New York: Cambridge University Press.

[19] Van Dijk, T. A. 1988. *News as discourse*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

[20] van Rijsbergen, C.J., Robertson, S.E. and Porter, M.F. 1980. New models in probabilistic information retrieval. London: British Library. British Library Research and Development Report, no. 5587.

[21] Weber, R. O., Ashley, K. D. and Brüninghaus, S. 2006. Textual case-based reasoning. *Knowledge Engineering Review, Special Issue: Readings on Case-based reasoning*.

[22] Xu , J. and Croft, W. B. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems*, 18(1):79–112.

[23] Yang, Y., Zhang, J., Carbonell, J., and Jin, C. 2002. Topic-conditioned novelty detection. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining*.

[24] Yi, L., Liu, B. and Li, X. 2002. Visualizing web site comparisons. In *Proceedings of the 11th international conference on World Wide Web*.

[25] Zhang, L., Pan Y. and Zhang, T. 2004. Focused named entity recognition using machine learning. In *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, Sheffield, United Kingdom.