COMBINING TEXT STRUCTURE AND MEANING TO SUPPORT TEXT MINING

by

Daniel Merrill McDonald

A Dissertation Submitted to the Faculty of the

COMMITTEE ON BUSINESS ADMINISTRATION

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN MANAGEMENT

In the Graduate College

THE UNIVERSITY OF ARIZONA

2006

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation prepared by Daniel Merrill McDonald
entitled Combining Text Structure and Meaning to Support Text Mining
and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy

_____ Date: 11/13/2006
Hsinchun Chen

_____ Date: 11/13/2006
Jay F. Nunamaker Jr.

_____ Date: 11/13/2006
Mohan Tanniru

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

_____ Date: 11/13/2006
Dissertation Director: Hsinchun Chen

STATEMENT BY AUTHOR

This dissertation has been submitted in partial fulfillment of requirements for an advanced degree at The University of Arizona and is deposited in the University Library to be made available to borrowers under rules of the Library.

Brief quotations from this dissertation are allowable without special permission, provided that accurate acknowledgment of source is made. Requests for permission for extended quotation from or reproduction of this manuscript in whole or in part may be granted by the head of the major department or the Dean of the Graduate College when in his or her judgment the proposed use of the material is in the interests of scholarship. In all other instances, however, permission must be obtained from the author.

SIGNED: Daniel Merrill McDonald

ACKNOWLEDGEMENTS

I thank Dr. Hsinchun Chen for his guidance, for being open to a variety of research areas, and for having my best interests in mind. His feedback and high standards have greatly helped me develop professionally. I wish to thank the members of the Artificial Intelligence Lab. Their friendship and esprit de corps helped lighten the mood during trying times and provide encouragement to meet deadlines and goals. In particular, I would like to thank Byron Marshall for the countless times we bounced ideas back and forth and for his energy that was so contagious. I also want to thank Cathy Larson and Kira Joslin for the friendly way they took care of so many details. I benefited greatly from feedback and insight from multiple professors at the University of Arizona. I wish to thank Drs. Jay Nunamaker, Kurt Fenstermacher, Terry Langendoen, and Mohan Tanniru in particular for their feedback, support, and for being available to students. I also thank Dr. Olivia Sheng at the University of Utah for her encouragement and support in my professional development. Also, I am appreciative of the grants by NSF and NLM/NIH that supported this work.

Most of all, I wish to thank my wife Karina McDonald. She was the sanity and balance in my life to help me see this work to its end. Her support and willingness to cover for me so many times allowed me to pursue this dream. Finally, I thank my children Andrew, Blake, Mariah, and Natalie for being an inspiration to me.

TABLE OF CONTENTS

TABLE OF CONTENTS – *CONTINUED*

TABLE OF CONTENTS – *CONTINUED*

TABLE OF CONTENTS – *CONTINUED*

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

Text mining methods strive to make unstructured text more useful for decision making. As part of the mining process, language is "processed" prior to analysis. Processing techniques have often focused primarily on either text structure or text meaning in preparing documents for analysis. As approaches have evolved over the years, increases in the use of lexical semantic parsing usually have come at the expense of full syntactic parsing. This work explores the benefits of combining structure and meaning or syntax and lexical semantics to support the text mining process.

Chapter two presents the Arizona Summarizer, which includes several processing approaches to automatic text summarization. Each approach has varying usage of structural and lexical semantic information. The usefulness of the different summaries is evaluated in the finding stage of the text mining process. The summary produced using structural and lexical semantic information outperforms all others in the browse task. Chapter three presents the Arizona Relation Parser, a system for extracting relations from medical texts. The system is a grammar-based system that combines syntax and lexical semantic information in one grammar for relation extraction. The relation parser attempts to capitalize on the high precision performance of semantic systems and the good coverage of the syntax-based systems. The parser performs in line with the top reported systems in the literature. Chapter four presents the Arizona Entity Finder, a system for extracting named entities from text. The system greatly expands on the combination grammar approach from the relation parser. Each tag is given a semantic and syntactic component and placed in a tag hierarchy. Over 10,000 tags exist in the hierarchy. The

system is tested on multiple domains and is required to extract seven additional types of entities in the second corpus. The entity finder achieves a 90 percent F-measure on the MUC-7 data and an 87 percent F-measure on the Yahoo data where additional entity types were extracted.

Together, these three chapters demonstrate that combining text structure and meaning in algorithms to process language has the potential to improve the text mining process. A lexical semantic grammar is effective at recognizing domain-specific entities and language constructs. Syntax information, on the other hand, allows a grammar to generalize its rules when possible. Balancing performance and coverage in light of the world's growing body of unstructured text is important.

1 INTRODUCTION

1.1 Prevalence of Textual Data Stores

Text is a common reporting, storage and communication format in business. Tan reported that 80 percent of a company's knowledge stores are found in textual databases (A.-H. Tan, 1999). More recently, Computerworld reported that textual data accounted for 85 percent of companies' information assets (Robb, 2004). Email communication contributes substantially to unstructured data in business. Information on best practices, lessons learned, corporate policies, and standard operating procedures also contribute to the total of unstructured data in managed organizations. In 1999, BAE Systems PLC, formerly British Aerospace, invested $150,000 to study the time spent by employees in gathering and processing information. They reported that almost 25 percent of a project's completion time was spent searching for best practices information (Hoffman, 2002). The large amount of time spent searching through text is an indication of the large quantities of text within the company.

The prevalence of unstructured textual data is not restricted to the proprietary content of businesses. The popularity and growth of the World Wide Web continues to make more and more text available to users. Through book scanning initiatives, Google is adding libraries of text from existing manuscripts to the Web. The blogging phenomenon has made Web publishing easier, motivating many Web users to publish personal and professional information online. Popular blogs also elicit many follow-up responses to original postings or to other threads. Popular social networking sites, such as MySpace

and Facebook have also boosted online text communication. These developments on the Web contribute to the increasing amount of publicly available text.

Research publications have also become more available due to online posting of journals and government sponsored publication databases, such as MEDLINE. This body of published research is growing at a very rapid pace. The National Library of Medicine reports that 1,500 to 3,500 completed abstracts are added every day to the MEDLINE database (Medicine, 2005). Particularly in medical research texts where the speed of discovery in very fast, more available research means more relevant research.

1.2 The Knowledge Discovery Challenge

Is the increasing amount of unstructured text, both inside and out of organizations, translating into user knowledge acquisition and a more productive workplace (Huber, 1991; Nonaka, 1994)? Are the tools for knowledge discovery and filtering keeping pace with the growth of relevant unstructured information? While the answer to these questions may be no, the challenge of converting rather low-level information into formats fit to support decision-making is not new. Knowledge management researchers recognize a process in which data becomes information, information becomes knowledge, and knowledge becomes wisdom (Ackoff, 1989; H. Chen, 2001). In data mining research, the progression from data selection, cleaning, transformation, and data mining to interpretation and evaluation is referred to as the knowledge discovery in databases (KDD) process (Welge, 1998). With regards to text, there are two main approaches to converting unstructured textual data into formats fit to support decision making: top-down approaches and bottom-up approaches.

Initiatives such as the semantic web and Wikipedia-style web sites strive to improve retrieval and processing of information in a top-down fashion. Top-down in this context refers to the reliance upon humans to organize unstructured text. For example, the burden of annotating web pages for the semantic web initiative mostly falls upon human annotators. In the case of Wikipedia-style tools, willing experts define and organize ideas and concepts in ways that facilitate knowledge acquisition. In both cases, humans organize text into a pre-defined structure.

Text mining or knowledge discovery in textual databases (Feldman & Dagan, 1995), on the other hand, is a bottom up approach to knowledge discovery. In a bottom-up approach all published text, from blogs to research papers, can be mined whether annotated or not. Researchers and practitioners have utilized text mining to analyze unstructured text. In science, text mining has been shown to assist the hypothesis-driven research process (D.R. Swanson, Smalheiser, & Bookstein, 2001). In the commercial sector, businesses are using enhanced text processing to speed knowledge acquisition. Google, for example, has implemented entity identification techniques and clustering to help augment their retrieval process. When a user searches for a medication, information about dosage, possible interactions, and other relevant information is placed at the top of the result list. Supporting this bottom-up text mining approach is the focus of this work.

1.3 Text Mining Definition

Text mining refers to a process where non-trivial, interesting patterns are extracted from text (A.-H. Tan, 1999). Figure 1.1 shows a diagram of the text mining process. There are three main steps in the text mining process. The first is the finding

step. For patterns and relationships to be "interesting" they must come from text that is relevant to the user's task. Finding relevant texts is the primary domain of information retrieval. The finding process may involve search, but can also involve identifying pre-assembled focused collections. For example, the entire MEDLINE collection could be

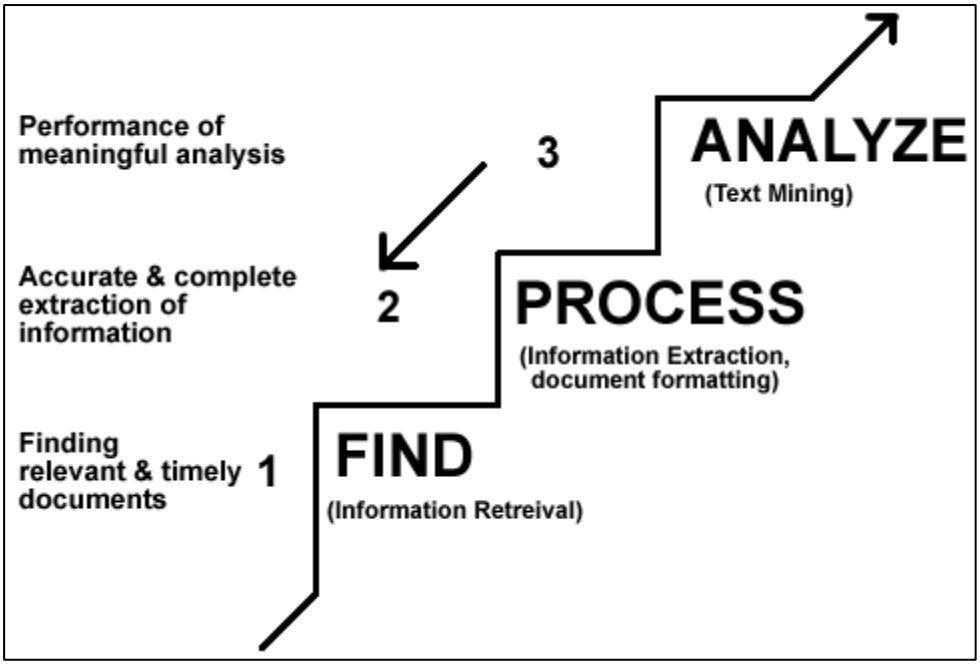

Figure 1.1 – The Text Mining Process

considered relevant for finding gene to gene relationships. A collection of Securities and Exchange Commission filings could be considered relevant for analyzing relationships between companies based on their boards of directors. Text from brainstorming sessions may be relevant for identifying relationships between discussed concepts. Another component of extracting "interesting" relationships is making sure the documents in a collection are timely. For example, when interested in current relationships between companies extracted from news articles, the news articles should be current. News is

time-sensitive so relationships and patterns extracted from current news may not only be more “interesting”, but also more accurate.

The second step in the text mining process involves the processing of the documents. Processing documents is typically the main concern of the information extraction field. Information extraction includes tasks that vary from identifying noun phrases (Tolle & Chen, 2000) and named entities (Sekine & Nobata, 2003) to filling information templates that involve many related pieces of information (Cowie & Lehnert, 1996; DARPA, 1998). The processing step is concerned with textual representation. Representations created during the processing stage can be passed along to the analysis stage or back to the finding stage. As shown in Figure 1.1, there are two arrows, one that suggests the output of processing is passed to analysis and another arrow suggesting processing output can be past back down to the finding step. Examples where additional processing is applied to the finding stage includes the indexing of noun phrases or events in a search index. The MIPT terrorism incident database, for example, allows users to search for terrorism events by location (MIPT, 2006). This search functionality is made possible due to the processing of the documents to recognize location entities in addition to event keywords.

The final step in the text mining process is the analysis step. Proper analysis takes the output of the processing stage, which may be a number of “trivial” relationships and finds relationships that were not obvious or perhaps not explicitly stated in the text. Insight from the analysis can come as a result of aggregating a number of relationships to produce patterns (Jenssen, Laegreid, Kmorowski, & Hovig, 2001). Insight can also take

the form of a single hypothesis or relationship that was unexpected (Blagosklonny & Pardee, 2002; Srinivasan, 2004). A successful example of hypothesis generation via text mining was Swanson's report of 11 connections between migraine and magnesium that were not previously recognized in the medical world (D.R. Swanson, 1988). The patterns or hypothesis that result from such relationship analysis can be used to improve the finding process (Houston et al., 2000) or interface with humans in the knowledge acquisition process (Morinaga, Yamanishi, Tateishi, & Fukushima, 2002).

1.4 Text Mining Background

The three main stages of the text mining process each have evolved separately over time. The background of each stage follows.

1.4.1 The Finding Stage

The finding stage of text mining or information retrieval owes much to the work of Gerald Salton (Salton, Wong, & Yang, 1975). His work with text indexing and the vector-space model is still the foundation of most information retrieval systems today. In addition to listings of relevant documents ranked by the vector-space model, additional text processing has been used to assist users in the finding stage of the text mining process. A common artifact used to assist users is a text summary. Early summarization work focused on the use of surface-level heuristics to locate summary sentences (Luhn, 1958). Later work included topic boundary identification (Hearst, 1997) and the development of summaries biased by queries submitted by users (Carbonell & Goldstein,

1998). Work is also being done on true text abstracting where ideas are identified and merged and grammatical sentences are created (Hovy & Lin, 1999).

1.4.2 The Processing Stage

The theory and early approaches to text processing were inspired by Noam Chomsky and his work with language syntax (Chomsky, 1957) as well as by Roger Schank and his work with semantics (Schank, 1972). Prior to the Message Understanding Conferences, text processing research was largely considered the domain of Natural Language Processing community. In late 1970s and early 1980's the first information extraction systems started to appear (DeJong, 1979, 1982). Information extraction provided well-defined tasks, real world texts, and performance metrics (Cowie & Lehnert, 1996). These practical benefits helped attract government funding, which helped achieve substantial improvements to text processing. The first MUC conference was held in May 1987 and the last took place in 1998 (DARPA, 1998). The conferences created a forum for the cross-fertilization of ideas which helped advance the field.

In the early MUC conferences, participants attempted full syntax parsing. In later conferences, however, systems moved to processing only pieces of sentences completely or applying only shallow parsing techniques to entire sentences. Full syntax parsing typically ran in polynomial time, while shallow parsing techniques could be converted into finite state automata for faster performance. Another important trend in the MUC conferences was the move to "shallow knowledge". As opposed to incorporating general expert-created rules, many domain-dependent and ad-hoc rules were used in systems (Cowie & Lehnert, 1996). The rules could be created so cheaply that it did not matter that

they could not be reused in other systems. At the same time, as full syntax and rich semantic analysis attempts were decreasing, the use of machine learning and statistical algorithms were increasing. Indeed, many of the rules based on shallow knowledge were obtained through statistical analysis (Riloff, 1993). Performance in some MUC tasks, such as entity extraction, achieved a performance that was considered commercially viable. In MUC-7, three groups were able to achieve a 90 percent f-measure, though approaches still suffered from domain dependence.

While tasks from MUC included among others entity extraction, co-reference resolution, and template tasks, medical text mining focused on relationship extraction. While there was some overlap in extraction approaches between the MUC and medical information extraction, differences started to emerge. For example, syntax parsing again became more prevalent in medical systems. As the number of medical abstracts in MEDLINE exceeded 10 million, creating completely domain-dependent approaches became less feasible. Syntax grammars were able to generalize better than rules generated from shallow knowledge.

1.4.3 The Analysis Stage

As information extraction has improved, the text mining possibilities have become more numerous. A good deal of text mining utilizes co-occurrence algorithms on text concepts to compute relationships (Houston et al., 2000; Jenssen, Laegreid, Kmorowski, & Hovig, 2001). Such techniques are built upon noun phrasing technology or pre-identified concepts or entities. Text mining also utilizes the output of entity extraction algorithms to identify product reputation trends (Morinaga, Yamanishi,

Tateishi, & Fukushima, 2002) and conduct environmental scanning (S. S. L. Tan, Teo, Tan, & Wei, 1998). As the number of entities and events that can be accurately identified in text increases, so should the types of analysis that can be performed.

1.5 Text Structure and Meaning in the Processing Stage

Information extraction systems, those performing the processing stage of the text mining process, usually have a greater reliance on either syntax parsing or lexical semantic templates and shallow knowledge. This tendency was observed in the evolution of the MUC systems away from heavy syntax parsing toward the use of shallow knowledge and lexical semantics (Cowie & Lehnert, 1996). The same tendency was observed in relation extraction systems in the medical domain (McDonald, Chen, Su, & Marshall, 2004). Some medical extraction systems focused on the use of syntax, while other focused on lexical semantics. In cases where both substantial syntax and semantic analysis are being performed, the semantic analysis is a more formal analysis that involves logical forms and goes beyond lexical semantics (Gaizauskas, Demetriou, Artymiuk, & Willett, 2003). While rules based on syntax tagging tend to apply to larger document sets, rules relying on lexical semantics tend to be more precise. Integrating these two types of information is important so that systems will be independent of any topical domain, but also accurate enough to be useful.

In this dissertation, three different studies are presented that are based on systems that have combined the use of structural information and lexical semantics to perform document processing in support of the text mining process.

1.5.1 The Arizona Summarizer

The Arizona Summarizer uses a combination of structural information along with lexical semantic heuristics to identify sentences from documents to include in summaries. The structural information comes from discourse analysis where a document is separated into its main topic areas. The summarizer strives to represent all topic areas in the summary. The lexical semantic information takes the form of heuristic cue phrases and term frequency analysis. The additional text processing involved in summarization is then used to support the finding process of text mining. Users select relevant documents based on the different types of summaries produced.

1.5.2 The Arizona Relation Parser

The Arizona Relation Parser combines word structure and meaning together by expanding the typical set of Penn Tree Bank tags to include many more semantically oriented tags. The result is a set of combination tags that are assigned in the tagging process. Instead of using rules with shallow knowledge, the tagging process assigns both the syntax and lexically semantic tags. The parsing of sentences is carried out much like a syntax parser with a generative grammar only with many more types of tags. Relationships are extracted based on the parsing structures dictated by the combination grammars. Relation parsing is a part of the processing stage of the text mining process. The relationships that are extracted, in this case, feed the analysis process as opposed to the finding process as with the summarization processing. At the analysis step, the

extracted relations are decomposed and aggregated to identify gene pathways (Byron Marshall, Su, McDonald, & Chen, 2006).

1.5.3 The Arizona Entity Finder

By adding additional tags with lexical semantic properties, the Arizona Relation Parser required individual parsing rules be added for each different type of noun tag used and for each different preposition tag used. This was feasible, however, because the number of additional lexical semantic tags numbered less than 200. Identifying entities in text, however, was a task that required much more lexical semantic information. While most systems incorporate this information via lexicons, the Arizona Entity Finder added new combination tags to recognize the additional lexical semantics. In order to minimize the duplication of rules, all the tags were placed in a hierarchy. The tag hierarchy allows parsing rules to be written using tags high in the hierarchy and still have them match their children tags. The entity finder combined structure and meaning by not just having both syntax and lexical semantic tagging unified in a single parsing process, but also having each tag contain some structure and some meaning information. As texts required, rules could be written that would focus on varying degrees of structure and meaning. Similar to the summarizer and relation parser, the entity finder is part of the processing stage of the text mining process. The output could be passed along to the analysis stage to identify trends or back to the finding stage to improve the finding process.

2 THE ARIZONA SUMMARIZER

2.1 Introduction

Today's web search engines use text summaries to help users make relevance decisions. Most summaries, such as those used by Google, are based on the query-terms from the user's search. Basing a summary on the query-terms used in the search makes the summary more focused to the user's query and less so to the context of the page. Page-level context information, however, may be helpful to the user when making relevance decisions. Context has been shown to be more important when a user is not as familiar with a search topic and their ideas are still fuzzy or when a user seeks background information (Carmel, Crawford, & Chen, 1992; Kuhlthau, 1991; Marchionini & Shneiderman, 1988). Despite these well-researched needs for context, text summaries provided by web search engines remain consistently query-based at the expense of page-level context.

Web searchers have been shown to distinguish between searching and browsing tasks (Hearst et al., 2002). Given such user ability, search engines could provide various types of summaries based on the user's need given the information-seeking task being performed. To the best of our knowledge, generic summaries have not been extrinsically evaluated given an information-seeking task. In the Document Understanding Conference (DUC), which started in 2001 (Over & Yen, 2004), generic summaries are evaluated intrinsically, being compared to human-generated summaries. In the TIPSTER Text Summarization Evaluation Conference (SUMMAC) (D. Mani et al., 1998), the first large

scale summarization evaluation conference held in May 1998, only query-based summaries were evaluated in a search scenario. Generic summaries were given an extrinsic categorization task. The value of page-level context, as provided by a generic summary, in searching has thus not been explored. In this chapter, we start by reviewing the relevant summarization and information-seeking literature. We then present the design and intrinsic evaluation of a generic summarizer. The generic summarizer is then used along with query-based and hybrid summarizers in an information-seeking experiment involving 297 subjects. The subjects complete search and browse tasks, each using a different type of summary. The paper ends with results and conclusions from the experiment.

2.2 Literature Review

A common way to separate research in automatic text summarization is by the focus or scope of the summary being produced (Firmin & Chrzanowski, 1999). Summaries that ignore the summary user, focusing more on the essence of the document are "generic" summaries. Summarizers that focus the summary on a topic or user are "query-based" summaries. This distinction is important as it often determines what summarization techniques to apply, what type of document information to utilize, and what type of evaluation method to use. Below, we review the literature on summarization techniques for each type of summary. We then review evaluation methodologies for each type of summary, focusing specifically on information retrieval tasks. We end the literature review by discussing the different types of information seeking tasks and how document context has been shown useful in browsing tasks.

2.2.1 Generic Summaries

Because generic summaries are not based on particular topics, techniques for creating generic summaries strive to include the most important sentences from a document based on indicators from the document itself. Techniques for identifying sentences fall into two general categories. The first category of generic summarization techniques creates groups of text that represent topic areas. Such textual areas usually span sentences, not relying on sentence boundaries to determine their size. Various types of discourse analysis and information extraction techniques create the textual groupings. The second technique groups text based on sentence boundary markers. This sentence-selection heuristic approach selects sentences for a summary based on pre-established features or characteristics of a sentence. These heuristics are often calculated using surface-level analysis and information extraction techniques. We will review techniques that place document text into topic groups first, followed by sentence-level techniques. We then review how information extraction techniques contribute to both approaches.

Discourse analysis refers to utilizing document meaning and structure that comes from text longer than one sentence (Liddy, 1998). If topic areas can be identified, then representative sentences from various topics can be included in the summary. Topic areas can be identified by analyzing document content or document structure. Content analysis focuses on the keywords in the document and the identification of co-referents. Structural analysis, also called coherence analysis, uses regularities in document structure to group sentences together.

2.2.1.1 Topic Finding using Document Content

Cohesion analysis, a type of discourse analysis, refers to measuring the connectivity or reliance sentences have on one another. In text summarization, word co-occurrence information and co-reference analysis approximate the cohesion of a document. In SUMMAC, nine of the sixteen participants utilized co-occurrence information including researchers from TextWise, Umass, and Cornell (D. Mani et al., 1998). Sentences with large numbers of co-occurring words have a greater likelihood of sharing a topic. Lexical chaining (Barzilay & Elhadad, 1999) is similar to co-occurrence analysis. Words between sentences are chained together based on their shared semantics. In co-reference resolution, sentences are linked together through anaphora or other types of reference resolution. The existence of pronouns from different sentences having the same referent shows cohesion between the sentences. An example of such approach is found in Boguraev & Kennedy (Boguraev & Kennedy, 1997). In SUMMAC, seven of the sixteen participants utilized co-reference techniques to produce summaries. TextWise, General Electric Research, and British Telecom used both co-reference and co-occurrence (D. Mani et al., 1998).

Another technique used for breaking documents into topic groups is text segmentation. TextTiling (Hearst, 1997) is an example of a popular text segmentation algorithm. In linear text segmentation, topic groups occur adjacent to one another. Other researchers have improved upon Hearst's algorithm. Kan et al. considers, among other things, the word classes of terms and semantic clustering of terms using WordNet 1.5 in their segmenting algorithm, SEGMENTER (Kan, Klavans, & McKeown, 1998). More

recently, Choi introduced a new algorithm for text segmentation that increased the accuracy of the TextTiling and SEGMENTER algorithms (Choi, 2000). Additional techniques such as sentence clustering have been used to group sentences into topics. Nomoto (Nomoto & Matsumoto, 2001) and Radev (Radev, Jing, & Budzikowska, 2000) present topic identification strategies based on clustering. Once topic areas have been identified, summary sentences can be selected from different topics (McDonald & Chen, 2002). Summarization techniques that rely on segmentation usually do not combine traditional surface-level analysis with the topic boundary information.

2.2.1.2 Topic Finding using Document Structure

A document's structural information is largely obtained using coherence analysis. Coherence analysis, a type of discourse analysis, captures structural information by identifying relationships between sentences or clauses. Marcu used rhetorical structure theory (RST) trees to represent the relationships between elementary clauses in text. The nucleus nodes on the tree were considered more salient (and thus relevant) than the satellite nodes for a summary (Marcu, 2000). Strazalkowski et al. utilized the concept of Discourse Macro Structure (DMS) to capitalize on regularities of organization and style in text to choose the best summary sentences (Strzalkowski, Wang, & Wise, 1998). Again, by separating sentences into groups, sentences that cover different topics can be added to the summary.

2.2.1.3 Surface-level Analysis

The remaining techniques of generic text summarization are the oldest (Luhn, 1958) and most commonly used methods. These techniques use surface and entity-level

analysis. At this level of analysis various features of sentences are calculated. Feature values can add to or subtract from the weighting of a sentence in a document. Researchers have validated individual features as well as combinations of features that can be identified at this level of analysis. Luhn in 1958 first utilized word-frequency-based rules to identify sentences for summaries (Luhn, 1958). Edmundson (1969) added three rules in addition to word frequencies for selecting sentences to extract, including cue phrases (e.g., "significant," "impossible," "hardly"), title and heading words, and sentence location (words starting a paragraph were more heavily weighted) (Edmundson, 1969). The ideas behind these older approaches are still referenced in modern text extraction research. Teufel and Moens found the use of cue phrases to be the best individual method (Teufel & Moens, 1999). Kupiec et al. reported that the best mix of extraction features included the position of a sentence within a paragraph, the existence of cue phrases in a sentence, and the sentence length (Kupiec, Pedersen, & Chen, 1995). They found first sentences of paragraphs to be good summary sentences along with sentences containing common cue phrases such as "in summary" and "in conclusion". Sentences that were simply longer also summarized better. In other research, summary sentences were found to have 90 percent more proper nouns per sentence (Goldstein, Kantrowitz, Mittal, & Carbonell, 1999). Identifying the number of 'signature words' in each sentence by using the common $tf \times idf$ calculation was also shown to be a positive contributor to summary quality as well (C. Aone, Okurowski, Gorlinsky, & Larsen, 1999).

2.2.1.4 Information Extraction Techniques

Information extraction techniques are used in summarization research to enhance topic finding and surface-level extraction techniques as well as to create task-based summaries. Noun phrasing can be used instead of single words to calculate a sentence's $tf \times idf$ value and entity extraction techniques are used to identify proper nouns in a sentence. Such techniques enhance the calculation of surface-level measures. Also, as mentioned above, generic summarization techniques have used co-reference resolution to chain together sentences of the same topic.

More significant, however, is the role information extraction techniques play in identifying information that can be extracted to support a particular task. In the biomedical domain, for example, genetic interactions can be extracted to assist biologists in constructing gene pathways (McDonald, Chen, Su, & Marshall, 2004). The extracted interactions summarize gene pathway information from the document. More general information extraction techniques including the extraction of template relations, template elements, and scenarios templates were tested in 1998 at the Message Understanding Conference 7 (MUC-7) (Chinatsu Aone, Halverson, Hampton, & Ramos-Santacruz, 1998). In general, extracted information must match a pre-defined structure or template, whether semantic, syntactic or some combination in order to be extracted (McDonald, Chen, Su, & Marshall, 2004).

Task-based summaries differ in several important ways from generic and query-based summaries. First, task-based summaries are informative summaries that replace the need to view the entire document. The information extracted (whether facts or events)

contains the information sought by the user. On the other hand, generic and query-based summaries are indicative summaries, which are created with the intent of helping users make relevance decisions. In addition, information extraction techniques are well suited for multi-document summary creation because they extract uniform information structures. The topic finding techniques and sentence-selection heuristics we have described above, however, apply mainly to single document summary creation.

2.2.2 Query-based Summaries

Query-based summaries focus their analysis at the entity or term level of a document. Words and/or phrases from sections of text are compared to a user's query terms for similarity (Sanderson, 1998). Query-based summaries typically ignore the cohesion or structural similarity between different sections of text, focusing only on a section's query-term similarity. Another approach that utilizes keywords from documents is Maximum Marginal Relevance (MMR) (Carbonell & Goldstein, 1998). MMR views sentences as having redundancy and diversity (Katz, 1996). MMR-based summaries have high redundancy (or relevance) to the query terms and low redundancy to each other. An MMR-based approach strives to present sentences as diverse as possible, but still related to the query.

Keyword-in-context (KWIC) or snippet summaries are query-based summaries that show which query-terms appear in a document and the words around those query-terms. KWIC summaries used to be rare in information retrieval systems and search engines because of the need to cache the documents locally in order to compare them to query terms (M. Hearst, 1999). However, Google has greatly popularized this

summarization method. Some research has shown that text fragments that occur near the beginning of a document and those that contain the largest subset of query terms are the best to include (Kupiec, Pedersen, & Chen, 1995). To promote the readability of the summary, the extracted fragments of text are shown in the original document order. The highlighting of the query terms in the summary has been shown to be a useful feature for displaying a KWIC summary (Landauer et al., 1993).

2.2.3 Evaluating Summaries

Generic and query-based summaries are usually evaluated in different ways. Text summary evaluations fall generally into two categories, intrinsic and extrinsic (I. Mani & Maybury, 1999).

2.2.3.1 Generic Summary Evaluation

Generic summaries are most often evaluated intrinsically. Intrinsic evaluations measure the quality of a summary against an ideal. Evaluators compare the summary to human-generated summaries (Kupiec, Pedersen, & Chen, 1995). The evaluator can scan the summary for the inclusion of key "ideas" from the original text (Brandow, Mitze, & Rau, 1994) or check for fluency (Minel, Nugier, & Piat, 1997). However, problems arise when comparing machine-generated summaries to an "ideal" because there is no single correct summary. Techniques for reducing the subjectivity of the process include creating the "ideal" summary by taking the majority opinion of the experts. Jing et al. (Jing, Barzilay, McKeown, & Elhadad, 1998) provide a more complete discussion of intrinsic evaluations and their challenges. In the Document Understanding Conference (DUC), generic summaries are evaluated intrinsically, being compared to human-generated

summaries. In the Summarization Evaluation Conference (SUMMAC), generic summaries were evaluated using an extrinsic categorization task.

2.2.3.2 Query-based Summary Evaluation

Query-based summaries are most often evaluated with extrinsic tasks. Extrinsic evaluations are task-based evaluations that measure the quality of a summary by testing the user's ability to complete some other task. For example, users may have to answer questions based on the summary text alone (Morris, Kasper, & Adams, 1992) or determine the relevance of a document using only the summary (Brandow, Mitze, & Rau, 1994). In DUC, query-based summaries are evaluated using an information-seeking task. A similar information-seeking task was used at the SUMMAC conference. Because selection of summary sentences is based on the query-terms used in a search, information-seeking tasks are an intuitive way to evaluate query-based summaries.

2.2.3.3 Review of Summaries for Information Seeking

Text summaries have frequently been evaluated using tasks common to information retrieval (D. Mani et al., 1998). Informative summaries, for example, have been evaluated as document replacements for creating search indexes and producing relevance feedback (Lam-Adesina & Jones, 2001). More common, even, is the evaluation of indicative summaries using an information-seeking task. Indicative summaries are not meant to replace the original document, but to provide enough information for the user to judge the relevance of the document (Firmin & Chrzanowski, 1999). In information-seeking tasks, summaries should help users judge document relevance. The type of information-seeking task being performed, however, affects the relevance decisions

(Harter, 1992; Schamber, Eisenberg, & Nilan, 1990). We will describe two distinct types of tasks in the information retrieval domain, review how subjects have utilized document context in browsing tasks, and discuss the use of document context in information seeking tools.

2.2.3.4 Information Seeking Tasks

The process of information seeking involves more than one task. Two general tasks have largely been agreed upon in the information retrieval literature, namely search and browse. Kuhlthau's work with High School students showed that students perform different tasks when looking for information (Kuhlthau, 1991). Kuhlthau found that at assignment inception users engaged in more general browsing with more directed search occurring as topic understanding increased. She found users' thoughts to be general and vague in the beginning of the search process while focused near the end. Other researchers have characterized differences between search and browse in similar ways. Cove and Walsh present a three-stage model with 'knowledge of the goal' being the primary factor dictating the information seeking stage (Cove & Walsh, 1988). Directed search is performed when the information need is well understood while general and serendipitous browsing is done when the information need is less clear. Marchionini and Sniederman characterized the difference between searching and browsing by the focus of the task. Searching was characterized as more directed and focused, while browsing was described as "an exploratory strategy…involving serendipity" (Marchionini & Shneiderman, 1988). Browse goals had general objectives, while search goals had

specific objectives. We end by reviewing the use of page context in browsing scenarios and discussing the lack of context in textual summaries.

2.2.3.5 Document Context

Document context normally contains content not related to the user's query terms, but rather related to the purpose and meaning of the document as a whole, somewhat like the author's perspective. Document context has been shown to be relevant in certain information seeking tasks. Carmel et al. used GOMS analysis to study browsing behavior in a hypertext system (Carmel, Crawford, & Chen, 1992). Browsing patterns observed were placed into three categories separated in part by how deep the user would delve into any one page. Of the three browsing tasks observed, the task involving deeper open-ended analysis of a document was the most common. In other words, while browsing, users relied on page context to study a document. Analyzing page context appeared to help users affect their own mental context for a topic. Also, in research from natural language processing, Black et al. note that humans effectively deal with the ambiguities of natural language sentences by examining the context (E. Black et al., 1992).

2.2.3.6 Single Document Context Ignored in Browsing Tools

Web browsing tools are largely based on two different types of web mining: web structure mining and web content mining (Etzioni, 1996). Web structure mining, also referred to as citation analysis, involves using the hyperlink information between documents to show similarity, relevance to queries, or other relationships between documents. Two well-known algorithms that utilize web structure are the HITS and Page Rank algorithms (Brin & Page, 1998; Kleinberg, 1999). In web content mining,

documents are related to each other or ranked by comparing keywords between documents. The keywords of a single page are all treated equally and compared to keywords from other documents. Tools that group documents together based on the page content include the hierarchical categories of YAHOO and visualization techniques such as the self-organizing map (SOM) (Hsinchun Chen, Houston, Sewell, & Schatz, 1998; H. Chen, Schufels, & Orwig, 1996). Thus, existing browsing tools mix all the content (both query-relevant and context) from each page together to represent the search "landscape". In such a representation, the document context (or author's perspective) of a single document contrasted with its query-relevant (searcher's perspective) content is lost in the aggregation process. To utilize document-level context, an algorithm must separate the context from query-related content.

2.3 Research Questions

The main question in our research is whether summaries that include more document context, such as generic summaries, are better suited than query-based summaries in either a browse or directed search information-seeking task. We explore whether document context as part of a summary helps users make relevance decisions better than adding additional information similar to a user's query. We first develop and intrinsically evaluate a generic automatic text summarizer, the Arizona Summarizer. Such evaluation is meant to legitimize our techniques for generic summary creation and show validity. Second, we add different variations of query-based summaries and conduct a user study of 297 subjects comparing their performance given search and browse tasks.

Such a large user study is conducted because generic summaries have not been evaluated using directed-search tasks before.

2.3.1 Summarizer Development

We have two research questions regarding the summarizer development. First, we explore how to construct a generic summarizer that balances the goal of drawing sentences from multiple document topics and at the same time containing good summarizing sentences. Systems at SUMMAC most often used co-occurrence and co-reference for cohesion analysis. We explore how to use a segmenting algorithm to identify topic areas and then how to choose sentences from within topics once the topic areas are identified. Second, we study what the resulting performance is of such a summarizing tool. We measure the performance of the generic summarizer intrinsically by benchmarking it against the human-generated summaries of TREC documents compiled by Jing et al. in their research (Jing, Barzilay, McKeown, & Elhadad, 1998). We test whether the performance of the Arizona Summarizer makes it generalizable to other systems and thus a good candidate for use in a large user information seeking study.

2.3.2 Summary Type and Task Experiment

We explore the usefulness of document context given search and browse tasks. More document context means users see content most related to the document overall and not that just related to the user's query. Generic summaries provide more context than query-based summaries because they include sentences from multiple topic areas or structural areas and do not focus solely on the content most related to the user's query.

Other summaries reduce context even more by producing a query-based summary that does not use full sentences, but rather only a small amount of text around the query terms, known as a snippet summary. Users are very often uncertain about their choice of query terms, especially during the early stages of the information seeking process (Kuhlthau, 1991). Showing topics only related to query terms might over-bias a summary. In addition, providing content only relevant to the query, may exclude information central to the document's meaning or purpose. We explore the question of how the reduction of page-level context impacts user information-seeking performance given search and browse tasks.

2.4 Arizona Summarizer Design

The Arizona Summarizer utilizes the boundaries of a document's topic areas along with the common surface heuristics from summarization research. The summarizer uses the TextTiling algorithm to calculate topic area information. Because we also rely on surface level summarization techniques, the TextTiling algorithm offered adequate performance for linear segmentation. Segmenting algorithms have not included sentence-selection heuristics to select individual sentences beyond that of using term-level calculations (Kan, Klavans, & McKeown, 1998). The combination of segmentation with surface-level heuristics represents the contribution of our algorithm. The number of topic areas in a document reflects the document's diversity of topics. Because generic summaries are not biased by a particular topic, we implement an algorithm that extracts equal numbers of sentences from each of the text segments. The pseudo code for the algorithm is shown in Figure 2.1. Because ideal compression levels (ratio of summary

```
sort sentences by weight
while (desiredSumLength is not met and there are unused sentences)
        for (all sentences x )
                if (sentence x not already in summary)
                        if (segment of sentece x has the lowest or equally low use)
                                set sum_sentence = x
                                break out of for loop
                        end if
                end if
        end for
        add sum_sentence to the summary
        record sum_sentence as having been used
        increment sum_sentence's segment use
        increment currentSumLength
end while
```

Figure 2.1 – Pseudo Code for Extraction Algorithm.

length to source length) for a document may change as users' tasks change, flexibility to handle changes in summary length is built into the summarization algorithm. After sorting the sentences by weight and entering the while loop, in the pseudo code, the algorithm extracts a sentence from a topic segment if it has not already been used and if its text segment is the least represented in the summary. The highest-ranking sentence from the least used segment is always added to the summary first. Two sentences cannot be extracted from the same topic segment until all topic segments have produced a sentence for the summary. This process continues until the summary contains the desired number of sentences or the document contains no unused sentences. Once the algorithm has extracted the desired number of sentences, it positions them in their original document order to promote fluency. Figure 2.2 shows an example of how the summarizer would select sentences from a document given it was segmented into three topic areas. A dashed line represents each sentence in the document (there are 15 total). The topic segment number to which each sentence belongs is listed on the right.

AZ Summarizer					
Sentence Rank	Sentence No.	Document	Topic Segment	Order Extracted	Summary Order
8	1		1		
1	2		1	1	1
15	3		1		
14	4		2		
12	5		2		
9	6		2		
3	7		2	3	2
6	8		2	5	3
13	9		3		
7	10		3		
10	11		3		
5	12		3		
2	13		3	2	4
11	14		3		
4	15		3	4	5

Figure 2.2 – Example of the Extraction Process for a Five-Sentence Summary.

The rank of each sentence as scored by the selection heuristics is listed in the first column titled ‘Sentence Rank’. The pseudo code in Figure 2.1 utilizes the sentence rank to sort the array. The algorithm keeps track of the most underutilized segment by using the topic segment information also listed in Figure 2.2. The ‘Order Extracted’ column shows what iteration of the pseudo code extracted what sentence. Once sentences are extracted, they are re-ordered to match their original document order, shown in ‘Summary Order’.

2.4.1 Structural Analysis

We use text segmentation to partition the document into multiple topic sections that span from one to many sentences. In particular, the TextTiling algorithm is the technique we use for segmentation. The TextTiling algorithm determines where topic boundaries are located. A topic boundary is the point where the document transitions

from one topic to the next. The first step in the TextTiling algorithm is to divide the text into token-sequences, removing any words that appear on a stop list. We used a token-sequence length of 20 and the same stop word list used by Hearst in her implementation. Token-sequences are then combined to form blocks. The first block contains the $(k+1)^{th}$ token-sequence plus k token-sequences before it. The second block contains the $(k+2)^{th}$ token-sequence and the k token-sequences after it. The value for k used in the Arizona Summarizer is 10, the same as used by Hearst. A similarity algorithm compares blocks to adjacent blocks. After each comparison, both blocks advance one token sequence, where the algorithm makes another comparison. The similarity algorithm returns the similarity as a percentage, which is derived from the number of times the same terms appear in the two blocks being compared. The Jaccard coefficient is used for the similarity equation, which differs slightly from the normalized inner product equation used by Hearst. The Jaccard coefficient is shown in Eq. (1). $S_{i,j}$ is the similarity between the two blocks of grouped token-sequences *i* and *j*. The variable w_{ik} is thus the total of the occurrences of term *k* in block *i* and w_{jk} is the total of the occurrences of term k in block *j*. In the numerator, the occurrence of unique terms (*k*) in both token-sequences *i* and *j* are multiplied together and summed over the set of total unique terms from both blocks, *L*. In the denominator of the equation, the occurrence of all words in the two token-sequences

$$S_{i,j} = \frac{\sum_{k=1}^{L} (w_{ik} w_{jk})}{\sum_{k=1}^{L} w_{ik}^2 + \sum_{k=1}^{L} w_{jk}^2 - \sum_{k=1}^{L} w_{ik} w_{jk}} \qquad (1)$$

is squared and totaled and the value of the overlapping words is subtracted out. Thus the similarity is a ratio of shared words to non-shared words. Where the similarity between two token-sequences is under a threshold, a segment boundary is created. Once segment boundaries are identified, sentences are assigned to a segment. The segment containing the first word in the sentence is the segment for the entire sentence when segment boundaries appear in the middle sentences.

2.4.2 Sentence and Entity-level Analysis

Sentence and entity-level analysis begins with a rule-based sentence boundary detector or sentence splitter. The sentence splitter recognizes 208 common abbreviations and uses rules to find periods that have been placed inside quotation marks or parenthesis. Because abbreviations present a specific problem there are also rules to anticipate abbreviations not in our lexicon. Once sentence boundaries are determined, the order of the sentences is recorded and each sentence is scored based on five sentence-selection heuristics. In selecting which heuristics to utilize, we incorporated as many as possible, knowing some heuristics would be more or less useful in different domains. Equation (2) shows the five heuristics that make up the sentence value (SV) for sentence k, SV_k. $S_{cp}(k)$ is the cue phrase value for sentence k, $S_{pn}(k)$ is the proper noun value for sentence k, $S_{sw}(k)$ is the signature word value for sentence k, $S_{sp}(k)$ is the sentence

$$SV_k = a_1 \times S_{cp}(k) + a_2 \times S_{pn}(k) + a_3 \times S_{sw}(k) + a_4 \times S_{sp}(k) + a_5 \times S_{sl}(k) \qquad (2)$$

position value for sentence k, and $S_{sl}(k)$ is the sentence length value for sentence k. The total of each heuristic is multiplied by a weighting factor (a1…a5) to normalize the

impact of any one score. Table 2.1 shows the points allotted to a sentence given the corresponding heuristic values.

Table 2.1 – Impact of Sentence Selection Heuristics

Sentence-selection Heuristics	Points allotted
1 Cue phrase	40 points
Proper nouns make up 27% of sentence words	34 points
$tf \times idf$ normalized for the length of the sentence	20 – 50 points is common range
The sentence begins a document	30 points
The sentence begins a paragraph	20 points
Sentence length of 358 characters	40 points

We manually adjusted the weights through a process of summarizing a training set of Information Technology articles and adjusting the weights after assessing each document to balance the impact of any one heuristic. The cue phrase heuristic was slightly weighted above the others. The other four heuristics were adjusted to have somewhat similar impact on the ranking of a sentence. Because the value of the weights is dependent on the domain of articles being summarized, we wanted to avoid over-committing to a particular domain. Also it has been reported that the weighting of different summarization features does not significantly affect the average precision (Lam-Adesina & Jones, 2001). Details of each heuristic are described below.

2.4.2.1 Cue Phrases

Each sentence is checked for the existence of ten different cue phrases (e.g. “in summary,” “in conclusion,” “in short,” “therefore”). Cue phrases indicate where an author may intend to summarize an idea. This heuristic has been weighted heavily because cue phrases are rare and have been shown to be good indicators of summary

sentences (Teufel & Moens, 1999). If a sentence has a cue phrase, the sentence value should be high regardless of the totals from the other four heuristics. The calculation of the cue phrase value for sentence k is shown in Eq. (3). The value $w_{cue}(t_i)$ is the weight of

$$S_{cp}(k) = \sum_{t_i \in s_k} w_{cue}(t_i) \tag{3}$$

the term t_i found in the cue phrase dictionary. Values in the dictionary range from –1 to 1. Currently, all cue phrases in the dictionary have a weight of one. All phrases not found in the dictionary are given a weight of zero. We may include phrases with a negative cue weight value in the future. A sentence's cue phrase value is the sum of the cue weights of the sentence's phrases, *sk*. The cue phrase value is then multiplied by a weight a1 to normalize its impact on the value of the sentence.

2.4.2.2 Proper Nouns

An indicative summary is meant to provide enough information for a user to decide whether to read the original document in its entirety. Proper names and places impact such relevance decisions. The importance of proper nouns, however, varies between domains. To obtain a rough estimate of proper nouns in a sentence we count capitalized words, not including the opening word in the sentence. While capitalized words do not always equate to proper names and places in our formatted Information Technology news domain, the correspondence is acceptable. A full entity-extraction system could be used to replace the reliance on word capitalization and even differentiate between the types of proper nouns in each sentence. The calculation for the scoring of

$$S_{pn}(k) = \frac{\sum_{t_i \in s_k} p_{noun}(t_i)}{|s_k|} \quad (4)$$

proper nouns found in a sentence is shown in Eq. (4). The value $p_{noun}(t_i)$ is the proper noun value of term t_i. Proper noun values are either one if the proper noun is recognized or zero otherwise. The proper noun value for sentence k is the sum of the proper noun values given to each term, t_i, in the sentence, s_k. The sum of the proper noun values is divided by the total number of words in the sentence, $|s_k|$. Thus, shorter sentences are not penalized for having fewer proper nouns than longer sentences.

2.4.2.3 Signature Words

The formula $tf \times idf$ or term frequency multiplied by inverse document frequency measures how common the words in a sentence are relative to the entire document. Signature words are words that are common to a sentence/document, but not to a document/collection. Sentences with more signature words are scored higher. The formula to calculate scores of average signature words per sentence is shown in Eq. (5), where w_{ik} is the $tf \times idf$ score for term t_i in sentence k. The $tf \times idf$ score for each term

$$S_{sw}(k) = \frac{\sum_{t_i \in s_k} w_{ik}}{|s_k|} \quad (5)$$

t_i from sentence s_k is summed and divided by the total number of words in the sentence $|s_k|$ to create an average. The calculation of w_{ik} is shown in Eq. (6), the $tf \times idf$ score for

term *i* in sentence *k*. In the formula, tf_{ik} is the term frequency of term *i* in sentence *k*. N is the total number of sentences in the document and n is the number of sentences in the document, which contain t_i (term *i*). Before term frequencies and document frequencies are totaled, each word is made lower-case and stemmed using the Porter stemmer. The Porter stemmer is one of the most widely used stemming algorithms (Jurafsky & Martin,

$$w_{ik} = tf_{ik} \times \log_2\left(\frac{N}{n}\right) \quad (6)$$

2000) and can be thought of as a lexicon-free stemmer because it uses cascaded rewrite rules that can be run very quickly and do not require the use of a lexicon. Stemming is performed so that words with the same stem but different affixes may be treated as the same word when calculating the frequency of a particular term.

2.4.2.4 Sentence Position in a Paragraph

As the sentences are extracted from the original document, new lines and carriage returns signal the beginning of new paragraphs. The beginning sentence of a document and the beginning sentence of each paragraph are scored higher due to their greater summarizing potential. The calculation of the sentence position score for sentence k is shown in Eq. (7). P_k is the position of sentence k in the document and $S_{pos}(P_k)$ is the value given to that position. Position values range between zero and one. Currently, sentences that start a document or a paragraph are given the value of one. All others are given the value of zero.

$$S_{sp}(k) = S_{pos}(P_k) \quad (7)$$

2.4.2.5 Sentence Length

The length of a sentence can provide clues as to its usefulness in a summary (C. Aone, Okurowski, Gorlinsky, & Larsen, 1999; Kupiec, Pedersen, & Chen, 1995). The sentence length calculation is shown in Eq. (8). L_k is the number of characters in

$$S_{sl}(k) = S_{len}(L_k) \tag{8}$$

sentence k and $S_{len}(L_k)$ is the value given to the length of sentence k. Before increasing sentence score based on sentence length, we tried to achieve the same effect by not averaging $tf \times idf$ scores over the number of words in the sentence. Longer sentences received higher scores by virtue of their greater number of signature terms. However, this approach disproportionately weighted long sentences. Sentence length became its own heuristic to remedy this problem.

2.5 Arizona Summarizer Extensions

In addition to the generic Arizona Summarizer, we developed a full-sentence, hybrid summarizer and a snippet, query-based summarizer. The hybrid and the query-based summaries were created to study the impact of varying amounts of discourse analysis and document context within summaries that were still based on the query-terms. The performance of all three summaries could then be compared in different information-seeking tasks.

2.5.1 Arizona Full-sentence, Hybrid Summary

In the full-sentence, hybrid summary, the five sentence-selection heuristics were replaced with a single $tf \times idf$ score of the query terms in the document. Each sentence

was treated as a document with all the sentences together composing the collection. The $tf \times idf$ equation used is shown in (6). Term frequency, tf_{ik}, was the occurrence of query term i in sentence k. N was the total number of sentences in the document and n was the number of sentences with the query term. The $tf \times idf$ score was summed over all the query terms. The topic information provided by the TextTiling algorithm was still utilized in this summary. Summary sentences were still chosen from different topic areas, but sentence ranking was based solely on the similarity of the sentence to the query (the $tf \times idf$ score). Because of its query-focus, this summary provides less document-level context than the generic summary, but more than the snippet summary because it uses full sentences coming from different topic areas as determined by discourse analysis.

2.5.2 Arizona Snippet, Query-based Summary

The snippet, query-based summary did not use any structural information obtained through TextTiling, nor did it include full-sentences. The snippet summary, also referred to as keyword in context (KWIC) (M. A. Hearst, 1999) in other research, was created by locating the query terms in the document and including the adjacent 55 characters of text on each side of the query term. To match the length of the full-sentence summaries, three total instances of query terms and their snippets were extracted from the document. Depending on how often the different query terms appeared in the document, the summary would include snippets from all the query terms or three different snippets using the same query term. The snippets were concatenated together, separated by 3 dots. The query terms in the snippets were bolded as is common with Internet search engines.

This type of summary was included in the experiment because it utilized no discourse analysis and contained less document context than the full-sentence summaries. In addition, query-based summaries are commonly used on the Internet by search engines such as Google.

2.6 Research Hypotheses

Our research hypotheses fall into two major categories. The first involves the evaluation of the generic Arizona Summarizer using an intrinsic study. The second involves the extrinsic evaluation of four different types of summaries using two types of information-seeking tasks.

2.6.1 Generic Arizona Summarizer Performance

We have one formal hypothesis to test regarding the performance of the Arizona Summarizer design.

Hypothesis 1 (H1): The Arizona Summarizer will perform the same or better than at least 2 published summarization systems evaluated on the TREC dataset at the 20 percent compression level.

2.6.2 Summaries of Varying Page-level Context in Information Seeking Tasks

We test five formal hypotheses regarding the value of page-level context found in text summaries given different information seeking tasks.

Hypothesis 2 (H2): The generic summary, which has more page-level context than the hybrid and query-based summaries, will perform better than the hybrid and query-based summaries in browse tasks.

Hypothesis 3 (H3): The hybrid summary and query-based summary will perform better than the generic summary in search tasks.

Hypothesis 4 (H4): Full sentence, hybrid summaries have more page-level context and thus will outperform query-based snippet summaries in browse tasks.

Hypothesis 5 (H5): Snippet, query-based summaries will outperform full-sentence, hybrid summaries in search tasks.

Hypothesis 6 (H6): The generic summary will outperform all other summaries overall.

2.7 Experimental Design

In order to test the six hypotheses above, we designed two experiments.

2.7.1 Arizona Summarizer Experiment: Intrinsic Evaluation

Using human-generated summaries as a gold standard is termed intrinsic evaluation and is the most common summarization evaluation technique. We obtained a well-established corpus of human-generated summaries of TREC documents used in previous research (Jing, Barzilay, McKeown, & Elhadad, 1998; Marcu, 2000). In this experiment, the computer-selected sentences are compared to the human-selected sentences for recall and precision at two different levels of compression. Compression is

the ratio of summary length to the length of the source document. Previous research has recognized the shortcomings of this approach explaining there is no single correct summary (Jing, Barzilay, McKeown, & Elhadad, 1998). To minimize such shortcomings, Jing et al. had five subjects create summaries on 40 news articles from the TREC collection. Each subject created a summary at 10 percent and 20 percent compression levels resulting in 400 total summaries. A summary of 10 percent compression is 10 percent the length of the original document. Sentences with the highest percent agreement were used to create the ideal summary for the 40 articles. Percent agreement is the ratio of observed agreements with the majority opinion to the possible agreements with the majority opinion. Refer to (Jing, Barzilay, McKeown, & Elhadad, 1998) for the details of the corpus creation. Summary length has been shown to impact the recall and precision performance of a summarizer. To control for variations of the 10 and 20 percent compression calculations, we added the same number of sentences to a summary that had been added by human subjects. In other words, the number of relevant sentences in the summary was also the same as the number of retrieved sentences. As a result, our precision and recall numbers are identical. If the human-generated summary had 6 sentences, so did ours. If 4 of ours were relevant, both precision and recall were 67 percent (4/6).

Given the corpus of human generated sentences, we calculated recall and precision for three different scenarios. First, we compared the Arizona Summarizer sentences to the consensus of ideal sentences. Second, we compared the Arizona Summarizer sentences to any of the human-generated summary sentences. Third, we

compared sentences randomly extracted to the sentences from the consensus ideal summaries. Because lead sentences in the news domain tend to be very good summarizing sentences, we also compared the lead sentences from the documents to the consensus ideal summary sentences.

Next, we compared the intrinsic performance of the Arizona Summarizer to other summarizers that have used the same data to calculate recall and precision. The comparison is made to determine support of our hypothesis that the Arizona Summarizer performs the same or better than at least 2 published summarization systems evaluated on the TREC dataset at the 20 percent compression level and is therefore generalizable enough to be used in an information-seeking experiment.

2.7.2 Using Page-level Context in Information Seeking Tasks

In order to test our hypotheses on information seeking using summaries, we utilized a collection of approximately 300,000 business Information Technology documents. We had obtained the collection through meta-searching various news magazine and editorial web sites (B. Marshall, McDonald, Chen, & Chung, 2004). We tested the effectiveness of four different types of summaries in identifying material relevant to two search tasks and two browse tasks. Such task-oriented approach is an accepted methodology for evaluating information retrieval systems (Hersh, Pentecost, & Hickam, 1996). Search tasks were narrow, requiring users to find documents with specific facts of one to two words. Browse tasks were broad and open-ended, which required users to understand more of the topic in order to choose relevant documents (Marchionini & Shneiderman, 1988). The tasks were patterned after the TREC Ad Hoc

Retrieval Tasks, but tailored to the Information Technology domain. The four tasks are listed in Figure 2.3.

Describe the open source movement. (Open-ended) What is XML? (Open-ended) What is the name of Novell's annual user conference? (Search) What is Bill Joy's position at Sun Microsystems? (Search)

Figure 2.3 – Tasks for Summary Experiment

Six tasks in total were considered for the experiment, 2 browse and 4 search tasks. Two query terms were used to represent each task and all the documents in the collection with the query terms were retrieved. The total number of documents retrieved varied between 20 and 40. Two search tasks were eliminated before the pilot because there were not enough irrelevant documents within the retrieved set. We reviewed the list of document results and selected 12 documents that had varying levels of relevance. At least 3 documents for each task were highly relevant. We attempted to include at least 7 documents that were less-relevant, which was difficult in some cases because all the documents contained the two query terms and were part of a highly focused collection that had been constructed through a meta-search collection-building methodology (B. Marshall, McDonald, Chen, & Chung, 2004).

Users were required to select only the three most relevant documents from the twelve listed. Setting the number of documents to select differs from traditional recall/precision experiments, yet allows subjects to compare the relevance of documents against each other as opposed to a relevance standard that may vary more between

subjects. The library expert read the full text of all the documents and selected the three most relevant documents of the twelve for each task. In one task, the expert found five documents to be equally relevant so all five were considered correct for all conditions in the experiment. Two small pilot studies were run to test the reliability of the web-based experimental platform, the clarity of the pre and post questionnaires, and the understandability of the tasks and experimental instructions. One browse task was altered after the pilot to be more open-ended.

A total of 297 subjects participated in the experiment. The subjects participated in the experiment as part of an introductory course in Management Information Systems. Participation in this experiment was a required part of the course. After logging onto the system, the user was presented with specific instructions for the experiment and then an 8 question pre-questionnaire. After the questionnaire, the user was presented with the first of the four tasks. Each of the four tasks was assigned to each subject in random order. Under each task, a list of 12 results was presented in random order. For each of the four different tasks, the result listing used different types of summaries. Each task-summary combination was also randomly assigned to the different subjects. Four tasks were chosen to match the four types of summaries. Using the same number of tasks as summaries was done to balance the impact of any one person on a particular task-summary combination. Every subject saw the same four tasks and used all four summaries. All tasks appeared with the four different types of summaries an equal number of times. Each summary type was selected a total of 891 times (see Table 2.4). A post-questionnaire followed the completion of the four information-seeking tasks.

Subjects were instructed to identify and mark the top three documents (of the 12 results) that provided the most relevant information to complete the task. The correct response to each task was therefore not the answer to the question, but rather to select the top three documents that would be most relevant to answering the question. Of course, if answers to the task did appear in the summary, this was strong evidence on which to base relevance decisions. Subject's relevance decisions were based on the summaries alone. The time spent on each task given the summary type was kept. To serve as the gold standard, an expert identified the top three relevant documents for each of the four tasks based on the entire document without referring to the summaries. The expert had previously conducted such relevance evaluations of documents from medical research and from general Internet searching. The expert had a Masters degree in Library and Information Science and also a BFA and an MA in Art History. Her library experience also included being a faculty member at The University of Arizona and Clemson University Libraries. Results from the experiment were tallied by totaling the correctly identified documents according to the type of summary used in the decision.

In order to control for different abilities in query formation, two query terms for each task were pre-determined and the resulting 12 documents were retrieved. All summaries were generated and stored before the experiment to normalize system response time. The hybrid summary and the query-based summary used the pre-determined query terms to generate the summaries.

The first type of summary was a 2-sentence generic summary created by the Arizona generic summarizer. The generic summary also included the top five keywords

from the page and the number of sentences in the page. The second summary was a 2-sentence hybrid summary created by the Arizona full-sentence, hybrid summarizer. In addition, this full-sentence, hybrid summary contained how many times the query terms appeared in the document and which of the query terms were found. The third summary was a 3 to 5 line snippet query-based summary created by the Arizona query-based summarizer. The length of the snippet summary was calculated to be as close to the length of the full-sentence summaries as possible. The fourth summary was obtained from the IT news sites that had published the documents. We refer to the fourth summary as the original summary and we included it to serve as a baseline for the performance of the Arizona summarizers. The "original" summaries were created using different techniques. Summaries from IDG, Infoworld, and sites powered by Google used snippet query-based summaries. Summaries from C|Net used the lead sentences from the articles and summaries from Computerworld and PCWorld appeared to be human-generated summaries. The length of the original summaries was usually shorter than the three summaries created by the Arizona Summarizers. All summaries also included the title of the webpage and the date of the article, when available. The query-terms found in the summaries were all bolded with the exception of the original summary.

Of the 297 subjects that participated in the experiment, 157 were male and 140 were female. The students had majors somewhat evenly distributed between Finance, Marketing, Management Information Systems, Management, and Accounting. Fifty-six percent of the subjects used mainly Google in their daily searching, followed by 31 percent for YAHOO, and 3 percent for MSN. Seventy-seven percent of the subjects were

native English speakers, while the other 23 percent had a different native language (20 languages represented in total).

2.8 Experimental Results

The results and discussion of the intrinsic experiment along with results and discussion of our information-seeking experiment with summaries of differing page-level context now follow.

2.8.1 Results of Intrinsic Evaluation Experiment

According to our experimental design, we evaluated intrinsically how the Arizona generic summarizer performed using human-generated summaries as the gold standard. In this experiment, we fed the automatic summarization routine the number of sentences allowed in each summary to meet the compression requirements. As a result, the percentages for recall and precision are identical (described in Section 2.7.1). The results of the experiment are summarized in Table 2.2. First, we compared the Arizona generic summarizer to the consensus of human-selected sentences. We achieved 50 percent precision and recall at the 20 percent compression level and 47 percent precision and recall at the 10 percent compression level.

Table 2.2 – Performance of Automatically Generated Summaries

Technique	Compression	Precision / Recall
AZ Summarizer compared to consensus ideal summaries	20%	50%
	10%	47%
AZ Summarizer compared to all human-generated sentences	20%	73%
	10%	58%
Random sentences compared to consensus ideal summaries	20%	20%
	10%	5%

Next, we compared the Arizona Summarizer to the most similar subset of human-selected sentences. This gold standard was therefore not a consensus of human selected sentences, but rather the set of human-selected summaries that best matched those of the Arizona Summarizer. Performance went up to 73 percent precision and recall and 58 percent precision and recall for the 20 percent and 10 percent compression levels respectively. Therefore, 73 percent of the sentences selected by the Arizona generic summarizer at the 20 percent compression level were also selected by at least one of the five human summarizers. Finally, a summarizer that randomly selected sentences for extraction achieved poor performance with a 20 percent precision and recall total at the 20 percent compression level and a 5 percent precision and recall total at the 10 percent compression level. Also, taking the lead sentences to generate a summary proved quite effective because the TREC corpus used was made up of news documents[1].

Next, we compare the performance of the Arizona Summarizer to that of four other summarizers that were evaluated on the same TREC data set (Jing, Barzilay, McKeown, & Elhadad, 1998; Marcu, 2000). While the significance of head to head comparisons has been questioned (Jing, Barzilay, McKeown, & Elhadad, 1998), we present this comparison only to support our hypothesis that the Arizona generic summarizer performs the same or better than at least 2 published summarization systems at the 20 percent compression level. Table 2.3 summarizes the results of the comparison.

[1] Lead sentence summaries achieved 58% precision and recall at the .20 compression level.

Table 2.3 – Summarizer Performance on TREC Corpus

Technique	Compression	Precision / Recall
Arizona Summarizer	20%	50% / 50%
	10%	47% / 47%
Marcu's Summarizer (Marcu, 2000)	20%	50% / 47%
	10%	N/A
Jing et al. Summarizer A (Jing, Barzilay, McKeown, & Elhadad, 1998)	20%	32% / 39%
	10%	33% / 37%
Jing et al. Summarizer B (Jing, Barzilay, McKeown, & Elhadad, 1998)	20%	47% / 64%
	10%	62% / 67%
Jing et al. Summarizer C (Jing, Barzilay, McKeown, & Elhadad, 1998)	20%	36% / 55%
	10%	46% / 64%

At the 20 percent compression level, the Arizona generic summarizer matched the best-reported precision measure of 50 percent. Recall performance of the Arizona Summarizer was ranked right in the middle compared to the recall numbers reported by the four other summarizers, confirming our first hypothesis.

2.8.2 Discussion of Intrinsic Summarization Experiment

The fact that 73 percent of the sentences extracted by the Arizona Summarizer were also extracted by at least one human is promising given the difficulty of reproducing human summaries. In addition, the Arizona generic summarizer easily outperformed a random summarizer in recall and precision. The Arizona generic summarizer performed better at the 20 percent compression level compared to the 10 percent level in part because the heavy reliance on lead sentences in the human summaries was lessoned as the length of the summary increased. The Arizona generic summarizer sought a diverse summary by adding sentences from different topic areas. Such a strategy is less likely to produce summaries with sentences appearing next to each other as occurs with lead-

sentence summaries. Based on the comparable performance of the Arizona generic summarizer compared to other summarization tools reported, we can confirm H1. The Arizona generic summarizer is generalizable and well suited for an experiment studying the information-seeking benefits of different types of summaries. In addition, the lower concentration of news articles in our IT collection compared to the TREC collection will limit the bias towards lead sentences, which may help our summarizer perform better.

2.8.3 Experimental Results of Information Seeking Experiment

We tallied the number of relevant documents selected by summary type and by information seeking task, whether browse or search. The total numbers are reported in Table 2.4. The grand total of 3,564 was the total number of summaries that were selected by the subjects in the experiment as being relevant (3 documents per task). From that total, an expert confirmed 1,263 or 35 percent of the documents were actually among the top three relevant documents for the task. By separating browse and search tasks and the four different types of summaries, we measure the tendency to make better relevance decisions given a certain type of information seeking scenario and summary.

Table 2.4 – Documents Selected by Subjects as Relevant to Browse and Search Tasks

	Browse tasks				
	generic	hybrid	original	query-based	Browse
Relevant	227 (51%)	144 (33%)	154 (34%)	127 (29%)	652 (37%)
Not relevant	220 (49%)	294 (67%)	305 (66%)	311 (71%)	1,130 (63%)
Total	447	438	459	438	1,782
	Search tasks				
	generic	hybrid	original	query-based	Search
Relevant	118 (27%)	145 (32%)	185 (43%)	163 (36%)	611 (34%)
Not relevant	326 (73%)	308 (68%)	247 (57%)	290 (64%)	1,171 (66%)
Total	444	453	432	453	1,782
Relevant Total	345 (39%)	289 (32%)	339 (38%)	290 (33%)	1263 (35%)
Total Selected	891	891	891	891	3,564

Results showed that given the browse task, the generic summary guided users to relevant documents 51 percent of the time compared to 29 percent for the snippet summary. Given a search task, the original summary successfully guided users 43 percent of the time compared to 27 percent for the generic summary. When ignoring the type of task, the subjects scored 39 percent accuracy with the generic summary, while achieving 32 percent accuracy with the full-sentence, hybrid summary. Overall, users were slightly more successful using the summaries in browse tasks (37 percent accuracy) than in search tasks (34 percent accuracy).

2.8.3.1 Summaries with Browse Tasks

We conducted a two-tailed t-test comparing the mean totals of relevant documents in the browse task by summary type to see if the totals could have come from the same distribution. The results from the statistical analysis are listed in Table 2.5. The generic summary produced significantly more relevant documents than the hybrid, full-sentence

summary (p-value = .0000), the query-based snippet summaries (p-value = .0000), and the original summaries from our content providers (p-value = .0000). These conclusions were also confirmed using analysis of variance (ANOVA), which showed between-group variation having significance less than .000 (with an F-score of 20.66 and 3 degrees of freedom).

Table 2.5 – Summary Type Results Given a Browse Task

Browse Task	T-stat	P-value	Result
Generic is better than the hybrid summary	7.747874	0.00000	H2 Confirmed
Generic is better than original summary overall	7.455501	0.00000	H6 Pending
Generic is better than query-based snippet summary	9.502109	0.00000	H2 Confirmed

Table 2.4 contains details of the page totals. Given this statistical evidence, we confirmed our second hypothesis that generic summaries with more page-level context more effectively lead users to relevant documents than query-based and hybrid summaries in browsing tasks. This finding highlights the usefulness of page-level context for browsing tasks. Often, such page-level context is ignored in favor of query-based summaries in Internet search engines. Interesting as well is that the generic summaries outperformed the summaries actually used by content providers in the browsing tasks.

2.8.3.2 Summaries with Search Tasks

Next, we conducted a statistical analysis using a two-tailed t-test comparing the mean performance of the different summary types in the "search" task. The results showed that users were able to identify significantly more relevant documents given the hybrid, full-sentence summary (p-value = .0001) and query-based, snippet summaries (p-

value = .0057) than they were given a generic summary. The statistical analysis is summarized in Table 2.6. These conclusions were also confirmed using analysis of variance (ANOVA), which showed between-group variation having significance less than .000 (with an F-score of 13.44 and 3 degrees of freedom). Our third hypothesis that both full-sentence, hybrid summaries and snippet, query-based summaries would outperform

Table 2.6 – Summary Type Results Given a Search Task

Search Task	T-stat	P-value	Result
Query-based snippet is better than generic summary	4.11075	0.0001	H3 confirmed
Hybrid full-sentence is better than generic summary	2.804726	0.00570	H3 confirmed

generic summaries was confirmed by the results. Thus summaries that are based on the queries alone are more useful given focused search tasks.

2.8.3.3 Full-sentence Hybrid Versus Query-based Snippet

Our fourth hypothesis was that full-sentence, hybrid summaries would outperform snippet, query-based summaries in a browse task. We reasoned that hybrid summaries contain slightly more page-level context than query-based summaries, which is of greater use in a browse scenario. Also, the full sentence hybrid summaries utilized some discourse analysis to identify and select sentences from diverse topical areas, while the snippet, query-based summaries only considered similarity to the query terms as selection criteria. Likewise, our fifth hypothesis stated that the snippet, query-based summaries would outperform full-sentence, hybrid summaries in search tasks. The reasoning was that snippet summaries are more focused to the query and therefore are more useful given specific tasks, where users are less concerned about page-level context. The results from

our statistical analysis (two-tailed t-test) are listed in Table 2.7. Given a significance level of .10, both hypotheses four and five were confirmed, the full-sentence hybrid summaries where more useful in browsing tasks (p-value = .0678) and the snippet query-based summaries were more useful in search tasks (p-value = .0802). The confirmation is only marginal, however, because the resulting p-values did not reach the same .05 standard as did the values in the other statistical tests.

Table 2.7 – Full Sentence versus Snippet Summary

Browse	T-stat	P-value	Result
Full-sentence hybrid is better than snippet query-based summary	1.840058	0.0678	H4: Marginal confirmation
Search	T-stat	P-value	Result
Snippet query-based is better than full sentence hybrid summary	1.76175	0.0802	H5: Marginal confirmation

2.8.3.4 Overall Summary Performance

To test our sixth hypothesis, we statistically compared (using a two-tailed t-test) the total number of relevant documents selected using the generic summary to the number selected by each of the other three types of summaries. Table 2.8 summarizes the statistical analysis for this comparison. While the generic summarizer outperformed both the hybrid and query-based summarizers (p-values = .00005), the overall performance of the generic summary closely paralleled that of the original summary (p-value = .661). Thus, we rejected our sixth hypothesis that the generic summary outperformed all other summaries overall.

Table 2.8 – Generic versus Other Summaries Overall

Overall	T-stat	P-value	Result
Generic is better than original summary	0.439075	0.6612	H6: Rejected
Generic is better than snippet query-based summary	3.95	.00005	H6: Supported
Generic is better than full-sentence hybrid summary	4.17	.00005	H6: Supported

2.8.3.5 Time Spent on Summaries

The time spent on summaries is also an important factor in evaluating their performance. We attempted to control for time by making the summaries the same length. In the experiment, however, subjects were not limited to the time they could spend on any one task, though they could not return to any task or lookup information not contained in the summary. We conducted some analysis on the time the users spent on each task given the type of summary. Table 2.9 lists the average time spent by users given each type of summary. Time did vary between tasks, but did not vary significantly from the mean time per task of 2 minutes and 11 seconds, as shown by the p-values in Table 2.9. Original summaries showed the greatest deviation from the mean (13 seconds).

Table 2.9 – Differences by Summary Type on Time Spent per Task

Summary Type	Ave. time per task	T-stat	P-value
Time spent on generic summary different from the average (mean)	2.23	1.491	.137
Time spent on full-sentence hybrid different from the average (mean)	2.21	1.242	.215
Time spent on query-based snippet different from the average (mean)	2.02	1.12	.264
Time spent on original summary different from the average (mean)	1.58	1.61	.107
Time spent on browse tasks	2.09	-	-
Time spent on search tasks	2.13	-	-

Original summaries were typically shorter than the others, however, as we could not control their length. In addition, the time spent on browse tasks was also very similar to that spent on search tasks. The average time for browsing tasks was 2 minutes and 9 seconds. The average time spent on searching task was 2 minutes and 13 seconds. Because the average task time spent given different types of summaries did not vary significantly from the mean, time is not considered a contributing factor to better performance in the information seeking experiment.

2.8.4 Discussion of Information Seeking Experiment

In the following sections, we discuss the experimental results in terms of the importance of page-level context. We also discuss other findings of the information seeking experiment.

2.8.4.1 Summarization in Context

The most useful text summary for web searching depends on the user's task. Our primary finding is that given a browsing task, users can better select relevant documents given a generic summary, while users make better relevance decisions using query-based and hybrid summaries in a search scenario. This finding is significant given the almost non-existent use of generic summaries on the Internet today. This finding, however, is consistent with theories on information seeking. While users are more vague and uncertain about a search topic, as is more common in browse scenarios, their query formulation will be less precise. During this search stage, less focus should be placed on those query terms for summarization because it may or may not be what the user really wants to see. The user is still in the process of refining their query.

We have suggested the difference between the generic summaries and query-based summaries to be one of page-level context. Such difference arises because generic summaries commonly utilize discourse analysis and query-based summaries use term or entity-level analysis. Also full-sentence summaries have more page context than snippet summaries. When a task is focused, as in a search task, there is less need for context and subsequently less robust discourse and language analysis. On the other hand, context becomes more important when the task is less focused and more open-ended. Figure 2.4 depicts the relationship between the importance of page-level context and the focus of the user's task. The less focused a task is, the more important contextual information will be to help the user understand the document. Researchers have also characterized user's feelings during the search process and found user's browse more when uncertain about a topic (Kuhlthau, 1991). Given this connection in research, we have added 'knowledge of the goal' to Figure 2.4, even though we did not directly test user's knowledge of the tasks given. This addition was made to show a connection between the knowledge levels characteristic to particular types of tasks. The addition also hints of future research that studies a users' familiarity with a topic, and how that familiarity affects the type of summary desired by a user given a particular type of task.

Full-sentence summaries were more beneficial in browse tasks, while snippet summaries were more beneficial given search tasks. In snippet summaries, users are shown where the query terms are in the document more so than any other summary. Search tasks are well summarized by their query terms. In open-ended browse tasks, query-terms insufficiently summarize what information is sought. Prominent page-

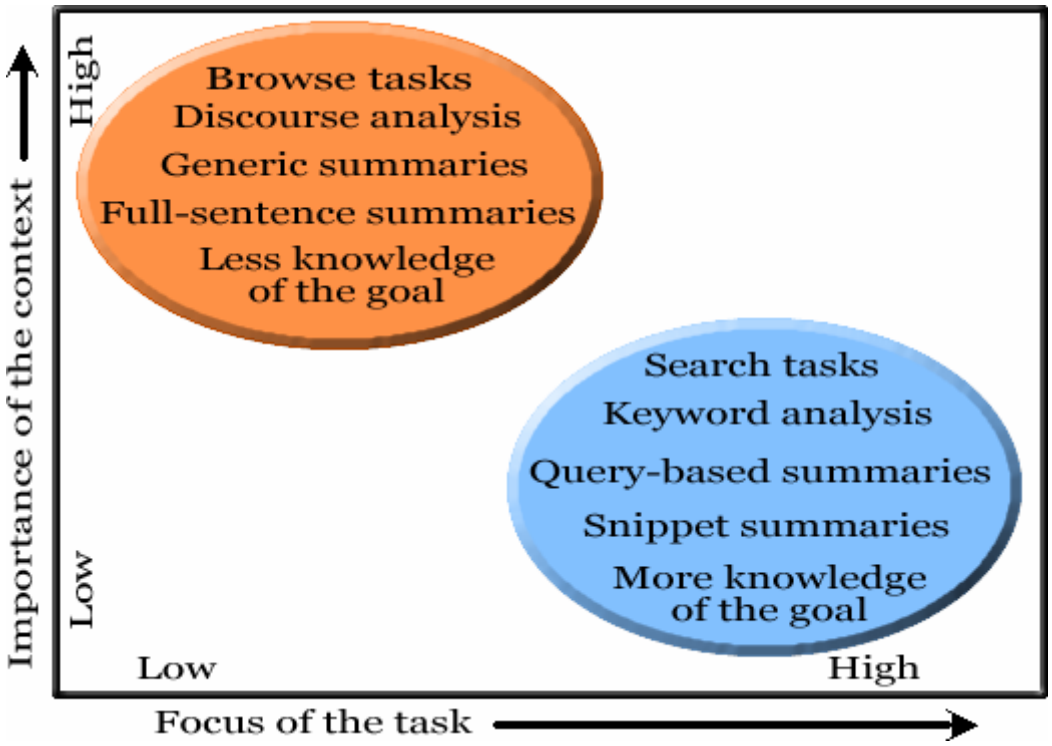

Figure 2.4 – Importance of the Context Given the Focus of a Task

level context may not be what information is sought, but it does help the user better judge the page's overall relevance and refine their query terms for the next search.

The fact that query-based and hybrid summaries outperformed generic summaries in search tasks also provides some insight into the difference between search and browse tasks. The scope of information sought appears to be different. Searchers appeared less concerned with the relevance of an entire page and more concerned about whether they got the bit of information they were seeking. To users performing a search task, the boundaries between documents are less important. More information about the query terms is the desired information. Acquiring the needed information is the ultimate goal and page-level context does not speed the location of such information. Users that are browsing, however, are more interested in page-level context. Not being as familiar

with the topic, users who are browsing want to make sense of each page to help them reformulate their search and better understand their topic. For them, a generic summary along with its greater use of natural language processing provides the context needed to make relevance decisions.

2.8.4.2 Implications for Information Retrieval

The importance of page-level context for users when browsing has some implications for information system design. As mentioned in the introduction, users have been shown to distinguish between their searching and browsing behavior. With this knowledge, users could indicate the type of information-seeking task they are performing and the summaries displayed could adjust accordingly to best support their task. In addition, browsing tools could incorporate the distinction of query-related content in a document versus document-level context. Browsing tools may only include phrases from document-level context areas or tools may cluster pages based on similarity of the query-related content and similarity of the document-level context as separate dimensions in the clustering algorithm. Finally, ranking algorithms could rank higher documents that have the greatest overlap between query-related content and document-level context information. If a document's most prominent sentences (in a generic summarization sense) were also related to the query, then the document may have higher relevance.

2.8.4.3 Native vs. Non-native English Speakers

While analyzing the results from the experiment, we also discovered some unanticipated differences between native and non-native English speakers in performance using the snippet summaries. Non-native English speakers did not find as many relevant

documents using the query-based snippet summary as they did using the three other types of summaries compared to native English speakers (p-value = .001). This finding was unanticipated, but appears to be consistent with our findings that page-level context is more helpful when users are less familiar with a topic or perhaps even the language of the source material. P-values in a t-test comparing native and non-native English speakers were all above .10 for the use of generic, hybrid, and original summaries.

2.8.4.4 Limitations of User Study

In this research, we presented the same number of search tasks as browse tasks to our subjects. In doing so, we ignore the prevalence of one task over the other on the Internet. However, browsing is a common information seeking process performed by users and given such tasks, generic summaries more effectively guide users to relevant documents. In addition, a limitation of our study was the use of a single expert in evaluating the relevance of each document.

2.8.4.5 Performance of Original Summaries

The original summaries from the content providers on the whole performed well. In the search task, the original summary was statistically the best performer. In the browse task, the original summary was statistically better than the snippet, query-based summary and statistically the same as the full-sentence, hybrid summary. As mentioned, the original summaries contained a combination of human-generated, lead sentence(s), and snippet summaries. The mix of summaries found in the original might account for the high and more balanced results in the experiment. A topic of future research will further

explore combining both types of information, page-context and query-focused content, into one summary.

2.8.4.6 Post-Questionnaire Analysis

After completing the experiment, users were asked to complete a short questionnaire. Users consistently choose a browse question as the most difficult and a search question as the easiest. This is interesting given that there was no statistical difference in performance or in time spent by users between browse and search questions in the experiment. Users were quite varied in their preference for different summaries. Overall, 34 percent of the users preferred the full-sentence, hybrid summary, 31 percent preferred the original summary, 22 percent preferred the generic summary, and 12 percent preferred the snippet, query-based summary. Participants who preferred the query-based summaries felt the computer was doing more work for them, which they liked. Those who preferred the original summaries liked how concise the summaries were. Finally, participants that preferred the generic summaries felt they were getting a better idea of what the document was really about.

2.9 Conclusions and Future Directions

In this section, we summarize our findings as well as present future directions for this research.

2.9.1 Conclusions

We have developed a generic text summarizer, using a blend of discourse structural information as well as sentence-selection heuristics. The summarizer achieved performance equivalent or better than two other published systems using the same corpus

of human-generated summaries as the gold standard. The summarizer was then modified to create two additional types of summaries, one full-sentence hybrid summarizer and one snippet, query-based summarizer. The hybrid summarizer used some discourse analysis and query-term information, while the query-based summarizer used only query-term similarity information.

Conferences on automatic text summarization including TIPSTER's SUMMAC in 1998 and the ongoing Document Understanding Conference (DUC) evaluate query-based summaries in information seeking environments, while generic summaries are evaluated intrinsically or using non-searching tasks. We have explored the use of four different summary types in search and browse information-seeking scenarios. Our findings indicate generic summaries are more useful than query-based summaries in open-ended tasks, despite their scarce use on the Internet. On the other hand, query-based summaries outperformed the generic summaries in more focused search tasks. We concluded page-level context helps users find more relevant documents when they are less focused in their task. Page-level context can be provided in summaries by not focusing on the query-terms, using discourse-level information, and using full sentences as opposed to snippets in summary creation.

In highlighting the role of page-level context in information seeking, we utilized a very large sample size of 297 subjects. Such a large sample was important given the lack of research comparing generic and query-based summaries. This sample size is the largest we have seen exploring the use of summaries in information-seeking tasks.

2.9.2 Future Directions

We have compared the performance of summaries with high page-level context to those with high query focus. We are interested in testing summaries with both types of information together in an information seeking experiment. Researchers in the field of information visualization have shown that "Focus + Context" interfaces (Greene, Marchionini, Plaisant, & Shneiderman, 2000; Plaisant, Carr, & Shneiderman, 1995) lead to faster user navigation (Börner & Chen, 2002; Pirolli, Card, & Wege, 2001). These interfaces deal with many documents or data records. We are interested in testing the "Focus + Context" paradigm in terms of single-document summaries that would provide some information that is focused to the query and other information that is relevant to the document-level context. The impact of "Focus + Context" in single-document summaries may help users make better relevance decisions and save time.

3 THE ARIZONA RELATION PARSER

3.1 Introduction

The MEDLINE database is a valuable source of biomedical research findings. The collection contains information for over 12 million articles and continues to grow at a rate of 2,000 articles per week. The rapid introduction of new research makes staying up-to-date a serious challenge. In addition, because the abstracts are in natural language, significant findings are more difficult to automatically extract than findings that appear in databases such as SwisProt, InterPro, and GenBank. To help alleviate this problem, several tools have been developed and tested for their ability to extract biomedical findings, represented by semantic relations from MEDLINE or other biomedical research texts. Such tools have the potential to assist researchers in processing useful information, formulating biological models, and developing new hypothesis. The success of such tools, however, relies on the accuracy of the relations extracted from text. We will review existing techniques for generating biomedical relations and the methodologies used to evaluate the techniques. We then propose a relation extraction tool, the Arizona Relation Parser and present the results of a thorough evaluation of the parser by an expert in biology. Finally, we draw conclusions and summarize our contributions.

3.2 Literature Review

Published systems that extract biomedical relations vary in the amount and type of syntax and semantic information they utilize. Syntax information for our purposes consists of part-of-speech (POS) tags and/or other information described in a syntax

theory, such as Combinatory Categorical Grammars (CCG) or Government and Binding Theory. Syntactic information is usually incorporated via a parser that creates a syntactic tree. Semantic information, however, consists of specific domain words and patterns. Semantic information is usually incorporated via a template or frame that includes slots for certain words or entities. The focus of this review is on the syntax and semantic information used by various published approaches. We first review systems that use predominantly either syntax or semantic information in relation parsing. We then review systems that more equally utilize both syntax and semantic information. Such systems, however, are constructed in a pipelined approach using one set of rules that utilize syntax information and another set of rules that utilize semantics. Syntax and semantic constraints have not yet been combined in a single rule in the bioinformatics domain. In our review, we will also draw connections between the amount of syntax and semantic information used and the size and diversity of the evaluation.

3.2.1 Syntax Parsing

Tools that use syntax parsing seek to relate semantically relevant phrases via the syntactic structure of the sentence. In this sense, syntax serves as a bridge to semantics (Buchholz, 2002). However, syntax parsers have reported problems of poor grammar coverage and over generation of candidate sentence parses. In addition, problems arise because important semantic elements are sometimes widely distributed across the parse of a sentence and parses often contain many syntactically motivated components that serve no semantic function (Jurafsky & Martin, 2000). To handle these challenges, some filtering is performed to eliminate non-relevant parses. Also, sentence relevance is judged

before parsing to avoid parsing irrelevant sentences. As reported in the literature, however, systems relying primarily on syntax parsing generally achieve lower precision numbers as compared to relations extracted from systems using full semantic templates.

In the following predominantly syntactic systems, key substances or verbs are used to identify relevant sentences to parse. Park et al. used a combinatory categorical grammar to syntactically parse complete sentence structure around occurrences of proteins (Park, Kim, & Kim, 2001). Sekimizu et al. used partial parsing techniques to identify simple grammatical verb relations involving seven different verbs (Sekimisu, Park, & Tsujii, 1998). Yakushiji et al. used full syntax parsing techniques to identify not just relations between substances, but the sequence of the relations as they occurred in events (Yakushiji, Tateisi, Miyao, & Tsujii, 2001). Others, while still predominantly syntactic, have incorporated different types of semantic information. Leroy et al. used shallow syntax parsing around three key prepositions to locate relevant relations (2003). Thomas et al. reported on their system Highlight that used partial parsing techniques to recognize certain syntactic structures (Thomas, Milward, Ouzounis, Pulman, & Carroll, 2000). Semantic analysis was then incorporated afterwards by requiring certain syntactic slots to contain a certain type of semantic entity. In addition, the system extracted only relations that used one of the verb phrases interact with, associate with, or bind to.

With the exception of Leroy et al. that reported a 90% precision, the highest precision reported from the syntax approaches did not exceed 83 percent. Park et al. reported 80 percent precision. Sekimizu et al. reported 83 percent precision. Yakushiji et al. reported a recall of 47 percent. The Highlight system reported a high of 77 percent

precision. Semantic approaches on the other hand have achieved precision as high as 91 and 96 percent (Friedman, Kra, Yu, Krauthammer, & Rzhetsky, 2001; Pustejovsky, Castano, Zhang, Kotecki, & Cochran, 2002).

Despite the lower performance numbers, the evaluations of syntax approaches typically involved a large number of documents. Park et al. evaluated their parser on 492 sentences, while Leroy et al. used 26 abstracts, and Thomas et al. used 2,565 abstracts. Using a greater number of documents in an evaluation shows the parser's performance when faced with different writing styles and topic content. Because syntax tags do not consider the semantics of words, rules that describe tag combinations do not require as much information as semantic rules and can thus be created with fewer resources.

3.2.2 Semantic Templates

Other systems rely more on semantic information than on syntax. Semantic parsing techniques are designed to directly analyze the content of a document. Rules from semantic grammars correspond to the entities and relations identified in the domain. Semantic rules connect the relevant entities together in domain-specific ways. Rindflesch et al. incorporate a greater amount of semantic information in their system EDGAR (Rindflesch, Tanabe, Weinstein, & Hunter, 2000). Documents are first shallow parsed and important entities are identified using the Unified Medical Language System (UMLS). Biomedical terms are then related together using semantic and pragmatic information. Performance was described as "moderate". GENIES (Friedman, Kra, Yu, Krauthammer, & Rzhetsky, 2001) and a system reported by Hafner and colleagues (Hafner, Baclawski, Futrelle, Fridman, & Sampath, 1994) rely primarily on a semantic

grammar. GENIES starts by recognizing genes and proteins in journal articles by using a term tagger. The terms are then combined in relations using a semantic grammar with both syntactic and semantic constraints. The system was tested on one journal article with the reported precision of 96% and a recall of 63%. In the system developed by Hafner and colleagues, a semantic grammar was developed to handle sentences with the verbs measure, determine, compute, and estimate. The grammar contained sample phrases acceptable for the defined relations. The system was in an early state of development when reported. Pustejovsky et al. used a semantic automaton that focused on certain verbal and nominal forms. Precision of 91 was reported along with a recall of 59 percent. The evaluation, however, only extracted relations that used the verb inhibit.

Semantic approaches, while more precise, are subject to poorer coverage than syntax approaches. As a result, tools that predominantly use semantic analysis are often evaluated using a smaller sample of documents or a smaller sample of relevant sentences. Templates in semantically oriented approaches require greater specification than the rules in syntax parsers. This requirement for more rules may be a practical reason for evaluating the parsers on smaller subsets of documents using fewer target verbs. GENIES was evaluated using one full text article. Pustejovsky et al. limited their relations of interest to inhibit relations, and Hafner et al. and Rindflesch et al. did not submit precision or recall numbers. However, the precision of semantic systems has been high. Both systems from Pustejovsky et al. and Friedman et al. (GENIES) reported precision above 90 percent.

3.2.3 Balanced Approaches

Balanced approaches utilize more equal amounts of syntax and semantic processing. Syntax processing takes place first, often resulting in an ambiguous parse. More than 100 parses can be generated for a single sentence (Novichkova, 2003). Semantic analysis is then applied to eliminate the incorrect syntactic parse trees and further identify domain words such as proteins and genes. In this fashion, systems combine the flexibility of syntax parsing with the precision of semantic analysis. Such combination has resulted in systems that have been evaluated over a large numbers of documents. Despite the use of both syntactic and semantic processing, however, problems specific to syntactic and semantic analysis persist in part because the analysis are still separate. Syntax grammars remain subject to poor coverage. Because semantic analysis only occurs after syntactic processing, a syntax grammar with poor coverage cannot be improved by the semantic analysis. At the same time, a syntax grammar with good coverage can still generate more parses than can be effectively disambiguated using semantic analysis.

Gaizauskas et al. reported on PASTA, a system that included complete syntax and semantic modules (Gaizauskas, Demetriou, Artymiuk, & Willett, 2003). The relation extraction component of PASTA was evaluated using 30 unseen abstracts. Recall was reported at 68 percent, among the highest recall number published, and a precision of 65 percent. The high recall and larger number of documents in the experiment suggest relatively good coverage of their syntax grammar. The relatively lower precision number

reflects the sparser coverage of the semantic module given the incoming syntactic parses and their task of extracting protein interactions.

Novichkova et al. reported on their system MedScan which involved both syntax and semantic components (Novichkova, Egorov, & Daraselia, 2003). Their first evaluation focused on the coverage of their syntax module, which was tested on 4.6 million relevant sentences from MEDLINE. Their syntax grammar produced parses for 1.56 million sentences out of the 4.6 million tested resulting in 34 percent coverage. Their relatively lower recall number suggests their syntax grammar lacks the coverage of the PASTA system. In a more recent study, Daraselia et al. reported a precision of 91 percent and a recall of 21 percent when extracting human protein interactions from MEDLINE using MedScan (Daraselia et al., 2004). Such a high precision supports the robustness of their semantic analysis given their task. However, the recall of 21 percent still shows the problem of a syntax grammar with relatively poor coverage. Balancing the use of syntax and semantic analysis contributed to MedScan's ability to be tested over a large sample size. Adding semantic analysis to the pipe, however, did not improve the coverage of the syntax grammar.

3.3 System and Methods

In the Arizona Relation Parser, syntax and semantic analysis are applied together in one parsing process as opposed to the pipelined approach that applies syntax and then semantic analysis in sequence. Others have shown such a combination to be effective for information extraction (Ciravegna & Lavelli, 1999), but we have not seen such a combination in the biomedical domain. We propose that the benefits of combining syntax

and semantic analysis can be realized by using a greater number of word classes or tags that reflect the relevant properties of words. Constraints limiting the type of combinations that occur are thus implicit by the absence of such parsing rules. With a greater number of tags, parsing rules must be explicitly written for each word class. When a rule is created and added to the system, it may be correct based on syntax, semantics, or some combination. The theory behind the rules is only implicit. While many rules have to be written to support the numerous tags, semantic constraints do not have to be specified in the system's lexicon. In addition, while explicitly stated theoretical constraints are absent, the use of more word classes may better lend itself to machine learning techniques, a topic of future research.

We report on two main research questions in this paper. First, how can a combined syntax-semantic parsing process be implemented for biomedical texts? Second, how does our implementation of the combined syntax-semantic parser compare in precision and recall to other published systems? Our study focuses on extracting relations that contain specifically genetic regulatory pathway information. A successfully extracted relation has a gene, protein or hormone as arguments in the predicate relation as well as a relevant predicate as judged by our expert. All the information extracted is in this form of predicate(argument1, argument2). Relations extracted are used by a gene network visualizer and are intended to help researchers construct gene pathways.

3.3.1 Representation and Rules

We use a notation adapted from Ciravegna and Lavelli to describe our parsing representation (Ciravegna & Lavelli, 1999). Every lexical element a in an input sequence

k is represented by a token T. Grammar rules have a binary form ($\gamma\alpha\delta$, Γ_R) where $\gamma\alpha\delta$ is a non-empty string of tags and is called the rule pattern; α is called the rule core and cannot be empty; γ and δ are called the rule context and can be empty. Γ_R is a set of rule transformations that act on the rule core only. A data structure called a token chart is dynamically maintained. The token chart is a directed graph where initial vertices correspond to the lexical entities in k. Arcs represent relationships among vertices from some finite set. At first, arcs between tags represent the original order of the corresponding lexical elements from k. During the parsing process, vertices are added to the graph to reflect constituency relationships among the vertices. Figure 3.1 shows the token chart from the parsing process (levels one through three) combined with a similar data structure, a knowledge pattern chart (KPC) (level four) drawn on top. The parser outputs the top two levels of the parse chart to the relation extraction process. The relations are then extracted by using knowledge patterns on the output of the parser. The boxed numbers correspond to the output of the four major processing steps of the parser. At level 1, words are broken into their lexical tokens and tagged. Shown at levels 2 and 3 are the parses from the token chart. The output of the parsing process (level 3) is the input to the relation extraction process, which is shown in level 4. The parsing process outputs at most the top two parse layers from each sentence.

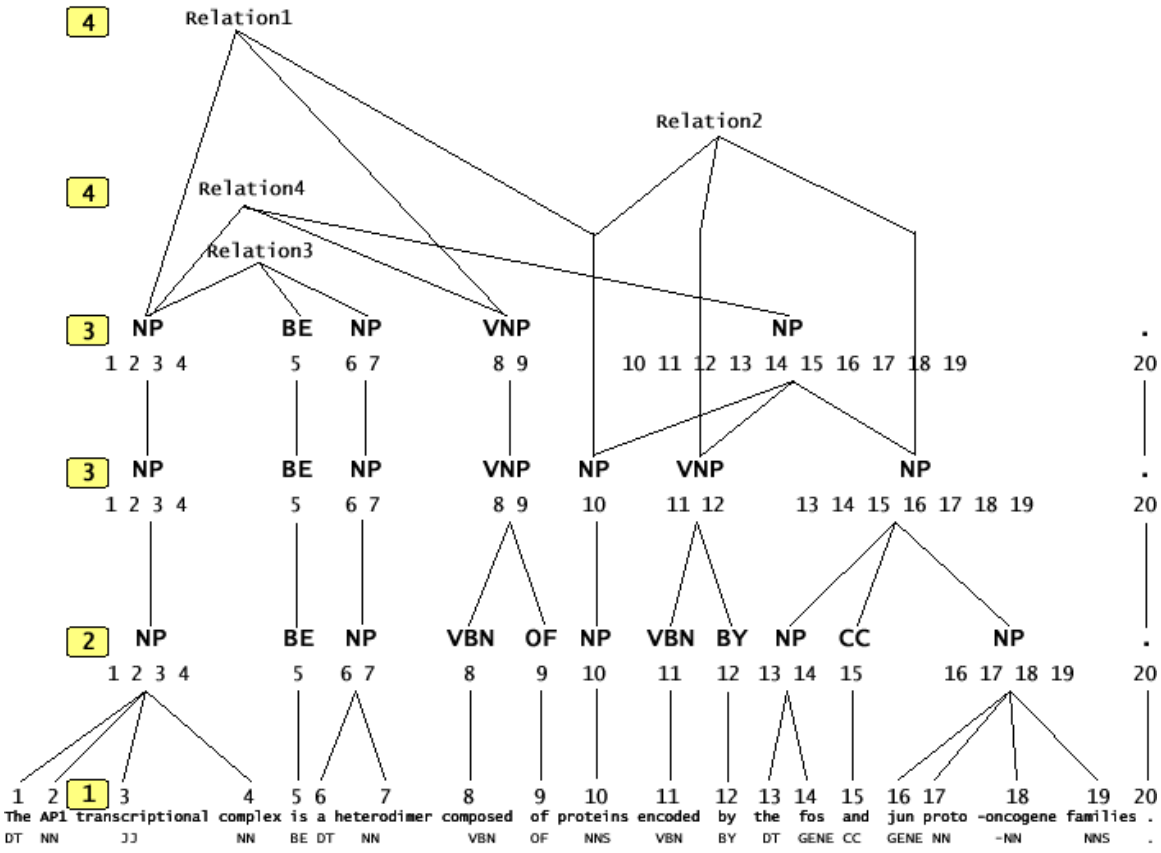

Figure 3.1 – A Token Chart (levels 1 – 3) with the Knowledge Pattern Chart (level 4) Applied on Top

3.3.2 Combination Grammar

As mentioned in the literature review, most systems that use syntax and semantics use the pipelined approach where semantic analysis follows syntactic analysis. Some systems have combined syntax and semantic analysis together by specifying semantic constraints in a parser's lexicon. We propose combining syntax and semantic analysis together by introducing over 150 new word classes to separate words with different properties. In comparison, the PENN TREE BANK has approximately 36 word classes. The majority of the new word classes added are semantically or lexically oriented, while

we also carry over a subset of the syntax tags from the PENN TREE BANK tag set. A sample of the tags is shown in Table 3.1. The set of tags was chosen using three primary methods. First, we started with a complete lexicon extracted from the PENN TREE BANK and BROWN corpora. The most common prepositions and verbs from our 40 abstract training set that had been assigned multiple part-of-speech tags from the PENN TREE BANK lexicon were then assigned a unique tag in our new lexicon. The existence or absence of later parsing rules would resolve the ambiguity between the function of the actual tag. The role of new tags would thus be determined by the way they could be parsed. As tags take semantic and lexical properties, as well as syntax, rules that apply to

Table 3.1 – Sample Tags from Combination Grammar

Selection Method	Tags
Unique tags in our lexicon obtained by observing ambiguous tags from the PENN TREE BANK	BE, GET, DO, KEEP, MAKE, INCD (include), COV (cover), HAVE, INF (infinitive), ABT (about), ABOV, ACROS, AFT (after), AGNST, AL (although), AMG (among), ARD (around), AS, AT, BEC, BEF (before) , BEL (below), BTN (between), DUR (during), TO, OF, ON , OPP (neg/opposite), OVR (over), UNT (until) , UPN (upon), VI (via), WAS (whereas) , WHL (while), WI (with), WOT
Domain relevant noun classes	DATE, PRCT, TIME, GENE, LOCATION, PERSON, ORGANIZATION
PENN TREE BANK syntax tags	IN, NP, VBD, VBN, VBG, VP, NN, NNP, NNS, NNPS, PRP, PRP$, RB, RBR

the tags, thus are reflecting semantic and syntactic phenomena. Second, domain relevant nouns were sub classed into groups of relevant substances or entities. Third, many of the original 36 PENN TREE BANK syntax tags were included in the new tag set. Using over 150 new tags with a regular grammar eliminates the problem of over generating parses.

Two different parsing rules are not allowed to act on the same input token sequence. Only one parse tree is generated for each sentence, with only two levels being analyzed for relation extraction. The particulars of the entire parsing process now follow.

3.3.3 Arizona Relation Parser

The general architecture of the Arizona Relation Parser is shown in Figure 3.2. The boxed numbers correspond to the boxed numbers shown in Figure 3.1 and indicate when new arcs are added to the parse chart. Each main component found in Figure 3.2 will now be explained in more detail.

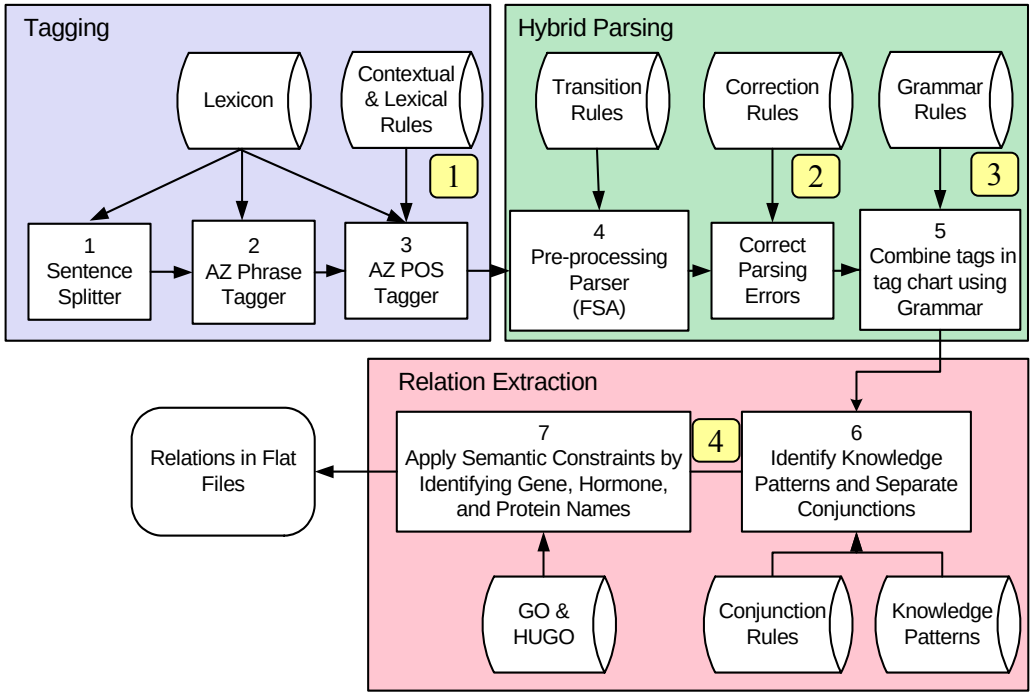

Figure 3.2 – Architecture Diagram for the Arizona Relation Parser Consisting of Three Main Stages: Tagging, Parsing, and Relation Extraction.

3.3.4 Sentence Splitter

The parsing process begins with tokenization, where word boundaries and sentence boundaries are recognized. The sentence splitting relies on a lexicon of 210 common abbreviations and rules to recognize new abbreviations. Documents are tokenized generally according to the PENN TREE BANK tokenizing rules. In addition, words are also split on hyphens, a practice commonly performed in the bioinformatics domain (Gaizauskas, Demetriou, Artymiuk, & Willett, 2003).

3.3.5 Arizona Phrase Tagger

The phrase tagger is based on a finite state automaton (FSA). The common idiomatic and discourse phrases (i.e. “for example” and “on the other hand”) are grouped together in this step along with other compound lexemes, such as compound gene names. We also mark the discourse phrases identified by Marcu in previous discourse research (Marcu, 2000). Such phrases receive a single initial tag, despite being made up of multiple words. We currently tag over 25,000 phrases. At the end of this process, the boundaries of the initial tags have been established.

3.3.6 Arizona Part-of-Speech/Lexical Semantic Tagger

We developed a Brill-style tagger (Brill, 1993) written in Java and trained on the Brown and Wall Street Journal corpora. The tagger was also trained using 100 MEDLINE abstracts and its lexicon was augmented by the words and tags from the GENIA corpus (Ohta, Tateisi, Hideki, & Jun'ichi, 2002). The tags used to mark tokens include over 150 new tags (generated by methods described earlier) along with the

original tags from the PENN TREE BANK tag set. A mapping between the new lexical/semantic tags and the original part-of-speech tag exits so that the Brill transformation rules function as normal. The tag lexicon consists of over 150,000 entries.

3.3.7 Pre-process Parsing

Abney first presented the idea of phrase chunking as a way of combining word groups with related semantics together (Abney, 1991). We perform a preliminary phrase-chunking step that makes simple parsing decisions. The phrase chunking combines straightforward nouns and verbs using limited context, thus reducing the number of rules required in our combination grammar and saving us from more computationally expensive parsing. The more ambiguous parsing problems, including the handling of conjunctions and prepositions, are done in the hybrid-parsing step.

The phrase chunking algorithm is similar to the Finite State Automaton (FSA) approach introduced by Church (Church, 1988). After phrase chunking, a transformation-based chunking correction routine corrects chunking errors using up to 6 tokens of context. Such a correction algorithm is similar to the transformation based rules used in the Brill POS Tagger. Figure 3.3 shows an example of the phrase chunking output. This output corresponds to boxed-number 2 in Figure 3.1 and Figure 3.2. In Figure 3.3, the words from the sentence are shown only for understandability. The tags alone (marked with an asterisk) are processed internally. The output from the pre-process parser is then fed to the combination parser.

```
[The AP1 transcriptional~complex/NP] [is/BE] [a heterodimer/NP]
[composed/VBN] [of/OF] [proteins/NP] [encoded/VBN] [by/BY]
[the fos/NP] [and/CC] [jun proto -oncogene families/NP] ./.
* NP BE NP VBN OF NP VBN BY NP CC NP .
```

Figure 3.3 – Phrase Chunking Output.

3.3.8 Combination Parser

While the use of 150 new tags helps reduce the over-generation problem by allowing more fine-grained rules, the poor grammar coverage problem remains. We attempt to address this challenge by relaxing some of the parsing assumptions made by full parsers. Our combination parser must accommodate the extraction of knowledge patterns. Full sentence parsing up to a root node of a binary branching tree is therefore not necessary. In contrast, the combination parser uses a shallow parse structure with n-ary branching, such that any number of tokens up to 24 can be combined into a new node on the tree. The input to the combination parser is the pre-parsing output, the sequence of tags after the asterisk shown in Figure 3.4. Internally, a cascade of four finite state automata attempts to match adjacent nodes from the parse chart to rules found in the grammar. Each FSA handles specific grammatical constructs:

Level 1: Conjunctions are recognized and combined as noun phrases or treated as discourse units.

Level 2: Prepositions are attached to verb phrases where possible and made into prepositional phrases elsewhere.

Level 3: This level catches parsing that should have taken place at level 1 or 2, but did not because of embedded clauses or other preprocessing requirement.

Level 4: Relative and subordinate clauses are recognized.

The output from both levels 3 and 4 are then passed to the relation extraction step. An example of the parsing output is shown in Figure 3.4. The sequence of tags after the asterisk represents the adjacent nodes in the parse chart.

```
[The AP1 transcriptional~complex/NP] [is /BE] [a heterodimer/NP]
[composed of /VNP] [proteins encoded by the fos and jun proto -
oncogene families /NP] [. /.]
* NP BE NP VNP NP .

[The AP1 transcriptional~complex/NP] [is /BE] [a heterodimer/NP]
[composed of /VNP] [proteins/NP] [encoded by /VNP] [the fos and
jun proto -oncogene families /NP] [. /.]
* NP BE NP VNP NP VNP NP .
```

Figure 3.4 – Parsing Output Including Two Levels from the Parse Chart

The combination parser utilizes a regular grammar that handles dependencies up to 24 tags or phrase tags away. Most sentences end before reaching this limit. Regular grammars have previously been used to model the context-sensitive nature of the English language, notably in FASTUS and its medical counterpart HIGHLIGHT (Hobbs et al., 1996). Different in our approach, however, is that the grammar rules are constrained by surrounding tags and thus are only fired when rule core and rule context tags are filled. Figure 3.5 gives and example of a grammar rule. In the rule from Figure 3.5, the rule pattern consists of the string “BY NP CC NP .”, the rule core equal to “NP CC NP”. The

```
<GRAMMAR LEVEL= “1”>
<RULE NUM=1>
<RULEPATTERN>
<PREVIOUSCONTEXT TAG=”BY” />
<RULECORE>NP CC NP</RULECORE>
<FUTURECONTEXT TAG=”.” />
</RULEPATTERN>
<TRANSFORMATION>
NP
</TRANSFORMATION>
</RULE>
```

Figure 3.5 – A Parsing Rule with a Rule Pattern and Transformation

rule core is transformed to a “NP” when the entire rule pattern is matched. Therefore, the rule core has to follow a “BY” tag and be followed by a “.” tag to be combined into a new noun phrase. The GRAMMAR LEVEL designation, in Figure 3.5, refers to which cascade of the four uses this rule. There are a total of 1,778 grammar rules applied in the four cascades. Currently 843 of those rules are unique, while the others apply in two or more levels. Figure 3.6 shows several of those rules with the rule core in bold and the

INP VDP NP . transforms to>>INP
IT VP **NP CC NP** *:* transforms to>>NP
ITS **VBN NP** transforms to>>NP
JJ , JJ , CC JJ . transforms to>>NP
AFT NP transforms to>>WHENP
AGNST NP transforms to>>AGPP
AMG NP CC NP II transforms to>>AMGP
ARD NP , transforms to>>ARDP
, **NP CC NP** *VBD* transforms to>>NP
, **NP CD NP** , transforms to>>NP
BE **NP VDP NP** transforms to>>NP
BE **RB JJ** transforms to>>JJ
WI NP CC NP , transforms to>>MOD

Figure 3.6 – Parsing Rules with Rule Core Bolded

rule context italicized. Some of the rules listed have empty rule context slots.

3.3.9 Relation Identification

The top two levels of the parse chart are passed to the relation identification step, boxed number 3 from Figure 3.1 and Figure 3.2. Relations can be loosely compared to subject, verb, object constructs and can be recognized in text using what we have called knowledge patterns. Knowledge patterns refer to the different syntactic/semantic patterns used by authors to convey knowledge. The relations extracted in this step are recorded in a directed graph called a knowledge pattern chart (KPC). Like the parsing rules, knowledge patterns consist of rule patterns ($\gamma\alpha\delta$), with rule context (γ, δ) and rule core (α) and transformations (Γ_R) that are applied on the rule core. Different from the parsing rules, however, are the actions that take place on the rule core. First, the rule core does not get transformed into a single new tag, but rather each tag in the rule core is recognized as playing a role, from a finite set of roles R defined in a knowledge pattern. Currently, there are 10 different roles defined in the set R. Each role is defined below. Roles 0 - 3 account for the more directly expressed knowledge patterns. Roles 4 - 9 recognize knowledge patterns that are less straightforward and require some additional processing steps. Examples are included with the definitions below.

Role 0: The tag plays no role

Role 1: The tag fills argument slot 1 of the predicate.

Role 2: The tag acts as the predicate.

Role 3: The tag fills argument slot 2 of the predicate

Role 4: The Role 4, like Role 6 applies to tokens that are prepositional phrases. Unlike Role 6, however, a Role 4 can only fill the second argument slot of a relation.

Role 5: This role captures relations expressed in a between phrase.

Role 6: The tag contains both Roles 2 and 3, separated by an "of" (Example 1) or a gerund verb (Example 2). Role 6 constructions occur in prepositions, unlike Role 8 constructions.

Example 1: [for involvement of c-Abl /FOR] [in recombinational repair of DNA strand breaks /INN]

Example 2: [K12/NP] [may increase /VP] [aggressiveness/NP] [not by altering proliferative pathways /HOW]

Role 7: Reformats verbs in the agentive form to serve as predicates.

Role 8: Role 8 captures verbs in nominal form (example 1) and also agentive form (example 2) when they appear in noun phrases.

Example 1: [induction of c-myc expression/NP] [by/BY] [GM-CSF/NP]

Example 2: [The Mdm2 oncoprotein/NP] [is /BE] [a potent inhibitor of p53 /NP]

Role 9: Predicates can contain at times many verbs. Role 9 signifies that only the main verb of a predicate be used in the relation.

Sentences can contain multiple overlapping knowledge patterns. In other words, a tag may play one role in the first knowledge pattern and a different role in a second knowledge pattern. The parse shown in Figure 3.1 used the knowledge pattern rule shown in Figure 3.7. The input to the rule was a subset of the output from the parsing step, "NP BE NP VNP NP VNP NP" (only the . is missing). The input was matched against the rule pattern from Figure 3.7. In this particular rule, there is no rule context. Given a match of the rule pattern, the transformation takes place that defines the relations that can be

```
<KNOWLEDGEPATTERN NUM="1">
<RULEPATTERN>
<PREVIOUSCONTEXT />
<RULECORE>NP BE NP VNP NP VNP NP</RULECORE>
<FUTURECONTEXT />
</RULEPATTERN>
<TRANSFORMATION>
   <RELATION SEQ="1">
        <SLOT ROLE= "1" TOKEN= "1" />
        <SLOT ROLE= "2" TOKEN= "2" />
        <SLOT ROLE= "3" TOKEN= "3" />
   </RELATION>
   <RELATION SEQ="2">
        <SLOT ROLE= "1" TOKEN= "1" />
        <SLOT ROLE= "2" TOKEN= "4" />
        <SLOT ROLE= "3" TOKEN= "5" />
   </RELATION>
   <RELATION SEQ="3">
        <SLOT ROLE= "1" TOKEN= "5" />
        <SLOT ROLE= "2" TOKEN= "6" />
        <SLOT ROLE= "3" TOKEN= "7" />
   </RELATION>
</TRANSFORMATION>
</KNOWLEDGEPATTERN>
```

Figure 3.7 – A Knowledge Pattern Rule

extracted, the tags participating in the relationship and the role they play. There were a total of 210 knowledge pattern rules at the time of this evaluation.

Knowledge pattern parsing mitigates the challenge of poor grammar coverage. Knowledge pattern parsing can ignore the majority of prepositional attachment. Some knowledge patterns (such as patterns involving roles 6 & 8) probe into prepositional phrases, but the majority do not. Also simplifying knowledge pattern parsing is the one-level deep limit placed on embedded clauses (example "p53 -induced cell death involves a Bax-dependent caspase-3 activation"). Only two levels of the parse tree are passed to the relation identification step to be matched against knowledge patterns. In addition, with the exception of relations involving embedded clauses, relations are independent of other relations. In other words parse trees do not have to combine together into a single root node. Therefore main clauses of sentences currently are not explicitly distinguished from subordinate clauses, further simplifying the parse. Future research involves incorporating a tag ontology to improve grammar coverage.

3.3.10 Conjunctions

When Role 1 or Role 3 contains a conjunction of noun phrases, the noun phrases are split after the relation identification phase. For example, the relation induce(p53, both growth arrest and apoptosis) would be split into the following two relations: induce(p53, growth arrest) and induce(p53, apoptosis).

3.3.11 Applying Semantic Constraints

Once potential relations are identified, each relation has to meet a number of semantic constraints in order to be extracted. In the current system, at least one word in Role 1 (the first argument role) and at least one word in Role 3 (the second argument role) had to exist in a gene/gene products lexicon, such as those from the Gene Ontology and HUGO. In addition, we used a list of 147 verb stems to filter the connectors in Role 2 (the predicate role). At least one of the words in Role 2 had to contain the verb stem. Examples of verb stems from the lexicon include activat, inhibit, increas, suppress, bind, catalyz, block, augment, elicit, promot, revers, control, coregulat, encod, downregulat, destabilize, express, hydrolyze, inactivat, interfer, interact, mimic, neutralize, phosphorylat, repress, trigger, and induc. Our biology expert generated the list of relevant verb stems by examining verbs appearing in MEDLINE.

3.4 Experimental Results

The goal of the relation parser design was to increase the recall and precision of relation extraction by combining semantic and syntax analysis by utilizing many fine-grained tags. The performance of the combination grammar together with the semantic filtering was tested in an experiment involving 100 unseen abstracts. Fifty abstracts were used to test precision and 50 for recall. We had extracted 23,000 abstracts related to the AP-1 family of transcription factors from MEDLINE as part of a test bed for Cancer researchers on our campus. We randomly selected 100 abstracts from the collection. For the precision experiment, a PhD in biology separated the parser-generated relations into

four different categories. Relations were separated by their ability to assist a biologist in creating a gene-regulatory pathway. Because the relations extracted feed a pathway visualization module, we sought a large variety of interaction relations. The first category (category A) consisted of genetic regulatory pathway relations, which included relations between two substances. An example of a category A relation is "inhibit(MDM2, SMAD3)" The second category (category B) consisted of partial gene-pathway relations. Category B relations had to contain a substance for at least one argument. A process was acceptable for the second argument. An example of a category B relation is "regulates(counterbalance of protein-tyrosine kinases, activation of T lymphocytes to produce cytokines)." Category C relations did not contain related substances, but provided relevant information for the task of constructing gene-pathways. Thus category C consisted of more general biologically relevant relations, an example being "mediate(target genes, effects of AP-1 proteins)." The fourth category (category D) consisted of partially relevant relations and/or incorrect relations.

The four categories were chosen by the expert beforehand to best characterize the content of the relations. The last category is the only one without sufficient relevance to gene pathways. The first two categories showed the greatest relevance for constructing gene-pathways. The results from the experiment are listed in Table 3.2. The precision of the parser was 90.8 percent after semantic filtering when considering relations from the first three categories as correct. The precision of the parser after semantic filtering was 63.6 percent when considering only category A and category B relations as correct.

Table 3.2 – Parser Performance Results

Precision (categories A, B & C)	90.8%
Precision (categories A & B)	63.6%
Recall before filtering	61%
Recall after filtering	34.7%

To perform the recall experiment, the expert in biology manually identified all gene pathway relations from 50 randomly selected unseen abstracts. We used the standard definition for recall where recall equals the ratio of system-identified relations to the expert identified relations. Two different numbers for system-identified relations were used to calculate recall. The first system total included all relations recognized before the semantic constraints described earlier were applied. The second total included all relations after semantic filtering. Results are shown in Table 3.2. The recall before semantic filtering was 61 percent. The recall after semantic filtering was 34.7 percent. The reasons why relations were missed are listed in Table 3.3. The largest number of relations was missed due to imprecise filtering, due to ambiguity caused by synonyms

Table 3.3 – Why the Parser Missed Relations

Reason	% Missed
Removed at semantic filtering stage	26.3%
Incomplete extraction rules	23.6%
Required co-reference information	12.5%
Parsing error	2.7%

of gene and protein names. We are in the process of replacing our semantic filtering step with a biological named entity extraction module to overcome this problem. The next largest group of relations was missed due to incomplete extraction rules. Since this evaluation, the number of extraction rules has more than doubled and their scalability is

being more formally tested elsewhere. The third largest group of relations was missed due to the parser's inability to handle co-reference. The expert identified relations that required co-reference resolution 12.5 percent of the time. The co-reference usually occurred between sentences. Along with the entity identification module, a co-reference module is being developed to address this problem. Finally, the parser missed 2.7 percent of the expert identified relations due to parsing errors.

3.5 Discussion of Results

When including biologically relevant relations (category C) in the total of correct relations, the precision of the parser (90.8%) is among the top performers that we reviewed. Such performance provides substantial support for the effectiveness of the combination grammar. However, when including just categories A & B as correct relations, the parser's precision drops significantly. We attribute this drop to the lacking sophistication of our semantic filtering approach. Our semantic filtering approach represents an efficient way to utilize current resources in the bioinformatics community, such as GO and HUGO, to approximate the identification of gene and gene products within noun strings. Such an approach is an appropriate substitute for gene and gene product identification only in the short term and will be replaced by more accurate algorithms for matching biological entities in the future.

The recall of gene-pathway relations before semantic filtering was 61%. This total also rates among the top reported recall numbers for biomedical relation extraction. To our knowledge only Friedman et al. (2001) and Gaizauskas et al. (2003) have reported a higher recall number. The high performance of the pre-filtered recall number attests to

the coverage of our combination grammar. Such coverage was shown over a large number of abstracts (50) and without any verb restrictions, such as extracting only inhibit relations. Despite the coverage of the combination grammar, however, a number of good relations were removed at the semantic filtering stage. The recall total after semantic filtering was 34.7 percent. The large decrease in recall performance relates again to the poor performance of the semantic filtering step. The "semantic constraints" described earlier that we applied turned out to be too strict in that many relevant relations identified were filtered out before extraction. The next large decrease in recall was a result of missing extraction rules that could have theoretically extracted the correct relation. While the coverage of the parser represents a strong point compared to recall numbers of other published systems, we expect more extraction rules to improve performance. The number of extraction rules has more than doubled since the experiment and now total nearly 500 rules (up from 210 at the time of the experiment). We are conducting additional experiments to calculate the coverage of the extraction rules and their potential to converge. A final positive outcome of the experiment was the low number of parsing errors, which totaled 2.7 percent. The low ambiguity of the grammar is a result of the increased number of semantic tags along with the grammar being specified to include rule context along with rule core.

3.6 Conclusions

The use of a combination grammar for biomedical relation extraction has shown promise for extracting relations with high recall and precision. The combination grammar seeks to decrease the ambiguity of syntax grammars, and at the same time increase the

coverage of the grammar above a semantic approach. With parsing errors reported at 2.7 percent, the grammar has been shown to be quite accurate. The combination grammar also performed well extracting biologically relevant relations with a precision of 90.8 percent and a recall before semantic filtering of 61%. The semantic filtering of the relations turned out to be less precise than expected. Recall of the relations declined to 34.7 percent after applying semantic filtering. As a result, more sophisticated algorithms are required to identify biological entities of interest in order to improve the recall and precision of the relation parser. Future directions for this research include such an undertaking.

4 THE ARIZONA ENTITY FINDER

4.1 Introduction

The quantity of unstructured text-based content continues to grow in various domains from business to education. Some have estimated that 80 percent of a company's knowledge assets are found in text. The ability to better understand the content of textual databases holds the promise of improving retrieval and analysis of relevant documents and subsequently decision making. Named entity extraction is an important step in understanding textual content. Identified entities can feed monitoring and tracking systems for entities and can form the foundation for the extraction of more structured events or relationships from text.

The task of named entity extraction was popularized in the Message Understanding Conferences (MUC) sponsored by DARPA. The conferences spanned ten years and were terminated in 1998. One task in MUC focused on identifying seven different types of entities, namely people, locations, organizations, dates, money, percent, and time. While such entities are quite relevant, especially for business domains, there are subclasses of these entity types and new entity types that are also important. For example, distinguishing between a business organization, a government organization, and terrorist organization is an important distinction for a terrorism-relation application. More recent research in named entity extraction has focused on increasing the number of entities that are identified from text. The ACE program added two new entity types: geographical and political entity (GPE) and facility. Other researchers have proposed entity hierarchies (Sekine & Nobata, 2003) and name classes based on WordNet (Morarescu & Harabagiu,

2004). As the number of entity classes grows, however, the accuracy of the extraction has declined. High performing algorithms that can identify new entity types and sub-classes of existing entity types without requiring manual rule additions would be a contribution to the field. As named entity extraction is an important foundational technology for Information Extraction, Question Answering, Summarization, and Information Retrieval, the greater number of entities that can be accurately identified, the greater improvement in the usefulness of the technologies that rely upon accurate extraction.

In this essay, we first review the literature of named entity extraction, grouping approaches by the algorithms used. We then present an approach to recognizing 14 different types of entities in financial news text. Central to our approach is a combination syntax-semantic tag hierarchy. We suggest that the use of extensive syntax-semantic tags as input to the extraction task is the novelty of our approach and a gap in published entity extraction literature. We then evaluate our approach first on the standard corpus from MUC-7 and then on a corpus of 50 financial news documents downloaded from the Web. We finish the paper with conclusions and future directions.

4.2 Literature Review

Approaches to extracting named entities have varied primarily by the algorithms used to transform the inputs into decisions. We have separated the main approaches to entity extraction into three main groups, namely approaches that utilize independent features, template-based approaches with shallow knowledge, and grammar-based approaches.

4.2.1 Independent Feature-based Approach

The independent feature-based approach refers to the use of rules or a classification algorithm that makes decisions based on inputs that are treated independently. For the sake of this approach taxonomy, the weights or rules for the classification can be either learned automatically or generated manually. Independent feature-based approaches are flexible and subsequently utilize various types of data as inputs. Inputs into the classification include such data as capitalization and orthographic information, part-of-speech information, binary values related to the existence of words in lexicons/gazetteers, and actual word sequences in various sizes (unigram, bi-gram, & tri-gram). Strengths of independent feature-based approaches include their flexibility. Features of various granularities can be included together as input to the classification. Because of the algorithm's flexibility, features can be chosen that generalize well to the corpus. Weaknesses of the independent feature-based approach include the need for a data set to contain all possible combinations of feature values when performing automated training. For example, a model with five features, each feature containing 100 possible values would result in a feature space of 100^5. As the number of relevant entity types increases, the need for more features to distinguish the entity classes drives up the size of the required training corpus. Also, a challenge is including lexical lookup as a feature because of the difficulty matching the boundaries of the text to boundaries used in the lexicon. For example, a lexicon that contains "Microsoft Corporation" would not match to the single word "Microsoft" found in the text.

In MUC-6, New Mexico State University submitted a system based on Quinlan's ID3 that was primarily a independent feature-based system (Cowie, 1995). The system used as features to the classification five adjacent words in a document. The algorithm would classify the center word as the beginning or ending of an entity. A different decision tree was built for each entity type. Despite using the actual words in the classification the words were not considered together as a single instance, but individually as separate inputs. NYU's entry into MUC-7 is another example of a independent feature-based system (Borthwick, Sterling, Agichtein, & Grishman, 1998). For MUC-7, NYU used a maximum entropy algorithm to compute tags for each token that preserved the greatest amount of entropy. Individual tokens could be assigned start, continue, or end tags for each entity type. Feature values in the model were independent of one another and then multiplied together in the entropy calculation. The Language Technology Group from the University of Edinburgh achieved the highest F-measure in MUC-7 (Mikheev, Grover, & Moens, 1998). Their system applied iterations of relaxing rules based on independent features along with partial matching of entities within a single document. Rules were iteratively relaxed as the possibility of phrases being conflicting entity types were resolved. The rules were made up of word context that provide clues to the entity type. The LTG system did resemble template systems in that their rules resulted from "shallow knowledge" and statistical analysis as opposed to machine learning, however the rules included only single independent features as opposed to treating variables together, which is why the system is included among the independent feature group. Other at least partially independent feature-based approaches include the FACILE

system by the University of Manchester (W. Black, Rinaldi, & Mowatt, 1998) and the DX system by SAIC used in MUC-6 (L. A. Miller, 1995). Both of these systems create vectors of feature values for each token. Rules are then manually created to recognize patterns that are indicative of certain types of entities. To the extent that rules include multiple variables on multiple tokens, the techniques are better classified as template-based systems.

4.2.2 Template-based Approach

Between independent feature-based and grammar-based approaches are the template–based approaches. Template approaches group a greater number of input variables together than do independent feature-based systems, where variables are treated independently. In a template, all conditions together must be satisfied before a tag assignment or transformation is made. However, unlike grammar-based approaches, the inputs to the template system do not have to address every token in the text being analyzed. Templates can represent a partial memory which includes wild cards for flexible slots unlike pure grammar-based systems that do not have gaps in input sequences. Grammar-based systems also have generative capabilities because of their complete definition of all inputs. Where systems have incorporated independent features and well as some grouped variables, we have tried to categorize those systems under their dominate strategy.

SRA's NetOwl finished first place in MUC-6 and second in MUC-7 and is a good example of a system employing rules than may include both independent features as well as some grouped variables (Krupka & Hausman, 1998). Rules in NetOwl are based on the

internal structure of an entity as well as the entity's surrounding context. An interesting part of NetOwl's approach is rule competition phrase that decides which rule is fired when conflict exits. Mitre's Alembic system for MUC-6 and MUC-7 used the transformation-based learning algorithm to name entities. Such an approach uses contextual clues to properly classify an entity and may also include independent and grouped variables (Aberdeen et al., 1995; Day, Robinson, Vilain, & Yeh, 1998). In this type of algorithm, phrasal context becomes the input variable. To the extent that multiple variables are included in the context, the system can be classified as a template system. In addition, the rules are statistical "shallow knowledge" rules which are common in template systems. The FACILE system from MUC-7 is another example of a template system made up of grouped inputs (W. Black, Rinaldi, & Mowatt, 1998).

CRYSTAL is an information extraction system that would automatically learn the attribute values that should be grouped together in a template in order to extract an entity (Soderland, Fisher, Aseltine, & Lehnert, 1995). The system utilized semantic information from the UMLS as well as verb and contextual information as constraints in the templates it would generate. LIEP is another system that would learn extraction patterns for an information extraction tool, in this case called ODIE (Huffman, 1995). Lockheed Martin's LOUELLA, used in MUC-6 is another example of a template-based system. LOUELLA used a pattern-matching algorithm to identify phrases that would meet a collection of constraints as defined by the language engineer. Constraints could be both syntactic and semantic (Childs et al., 1995).

Figure 4.1 shows visually some of the distinctions we have made between the techniques utilized for named entity and other types of information extraction. In the upper left of the graph are the independent feature-based systems where rules/actions are based on independent constraints. Closer to the middle of the graph are template-based systems that incorporate rules involving multiple constraints treated together.

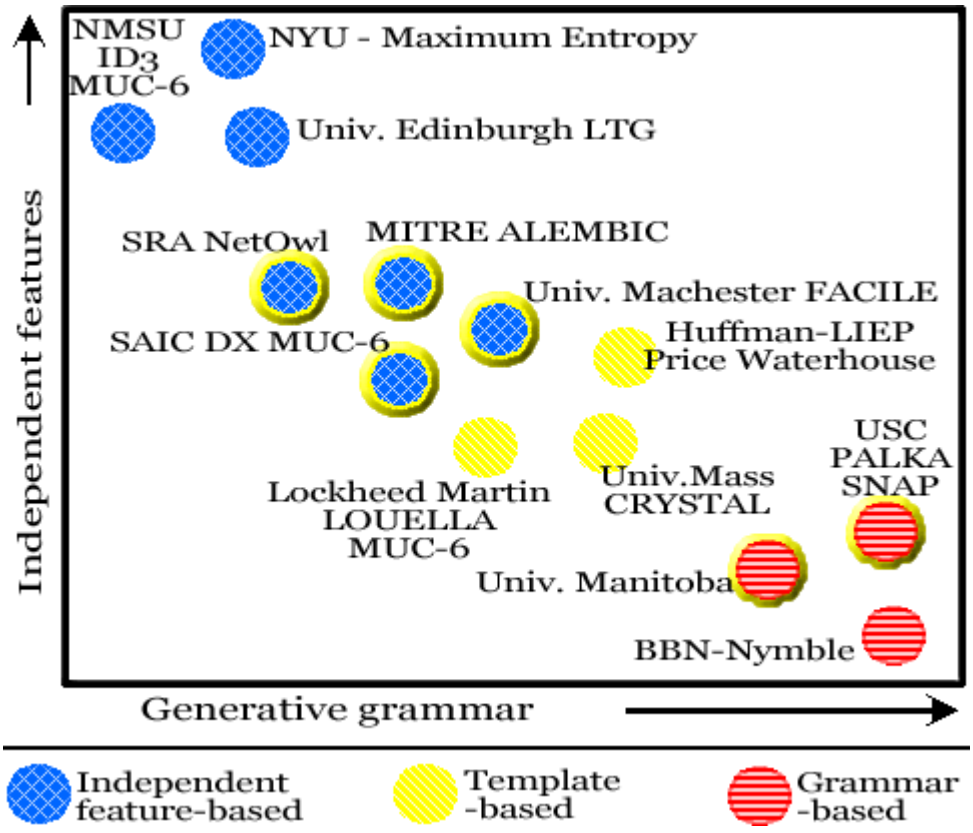

Figure 4.1– Taxonomy of Information Extraction Approaches.

These systems use both learned and human-generated rules. Finally, in the lower right corner are the grammar-based systems that must match an entire sequence of values to a grammar rule for tagging or transformations to take place.

4.2.3 Grammar-based Approach

A grammar-based algorithm typically tries to match an entire input string to a grammar rule from a repository. A grammar rule repository could also be referred to as memory or case base. Grammars can not only parse text, but have generative capabilities as well because they translate complete input strings for output tags, that process can be reversed. When the input matches a grammar rule input, the action to pursue is retrieved. Inputs to the algorithm are treated together as a whole instead of independently. Grammar-based algorithms tend to use more homogeneous representations of the input text. Grammar-based approaches include Hidden Markov Models that map input text sequences to their most probable output entity tags (Bikel, Miller, Schwartz, & Weischedel, 1997). Grammar-based algorithms also include sequence matching algorithms that map an input sequence to an output parse (Appelt et al., 1995; Buchholz, 2002). Grammar-based algorithms need only positive examples for training as opposed to an entire feature space as is required with machine learning on independent features. In addition, grammar-based algorithms that operate on sequences of words contain lexical semantic information, which is quite valuable in naming entities. The strength of the grammar approach is that when an entire input sequence can be matched to a "case base", high accuracy results. On the other hand, when word sequences are not found in the trained model, a back-off algorithm is used which is often considerably less accurate. To limit using the back-off algorithm considerable amounts of domain-specific training text is required.

BBN's Nymble system achieved the third highest score in MUC-7 (Bikel, Miller, Schwartz, & Weischedel, 1997). The system used a Hidden Markov Model (HMM) to match sequence of words in order to predict the most probable tag for the next word in the sequence. Categories for words could be the start middle or end of any particular entity. Because the words were non-independent and did not allow gaps in sequences, this approach is good example of a grammar-based approach. The University of Manitoba system for MUC-7 used an interesting variation of a template system and grammar-based algorithm (Lin, 1998). Their algorithm stored complete entity contexts in memory. New text segments were matched to contexts from the context memory. When contexts were matched, the phrase being analyzed was assigned the corresponding entity type. The SNAP system used in MUC-4 is another example of a memory-based parser (Moldovan et al., 1992). The PALKA system was created to automatically extract semantic phrasal patterns to support the SNAP system (Kim & Moldovan, 1993).

4.3 Research Questions

Our research aims to increase the number of entity types that can be accurately recognized in various domains of text. Current grammar-based systems for named-entity extraction, such as Nymble and Palka, focus primarily on semantic grammars. These systems have reported high performance in single-domain studies (Bikel, Miller, Schwartz, & Weischedel, 1997). Training a model on actual words relies to a greater degree on training data that overlaps with the testing data. When previously unseen sequences are encountered, a less accurate back-off algorithm must be used. Our research is focused on the impact of combining a syntax component alongside the semantic

component in the entity extraction grammar. Would building a grammar using the syntax-semantic combination tags allow the grammar to generalize better to texts from new domains? We seek to balance the rich representation and high performance of the language model approaches with the generic nature of structure or syntax information. We propose to explore the following two research questions in this regard:

1. What are the components of a tag that would be more generic than using words themselves and yet still expressive enough to be accurate?
2. How does such an extractor compare in performance to other algorithms?

4.4 Arizona Entity Finder Design

We have implemented a grammar-based algorithm and designed a combination syntax-semantic tag hierarchy to serve as the representation of words for our grammar. The algorithm acts like a sliding window, advancing through the text while matching sequences of tags from a document to a "grammar-base" of tag sequences. Input tag sequences then correspond to types of entities as well as other non-entity tags. Sequences of input tags may include entity context and must contain the entire sequence of tags that make up the entity. We refer to the extraction grammar pattern for different types of entities to be a lexical profile.

Figure 4.2 shows the overall system design of the Arizona Entity Finder. The tag hierarchy is shown at the top. Tag hierarchy tags are assigned to text in the "tagging" stage. Tags are corrected from their most common tag to their correct tag in the "correction" stage. Finally in the "parsing" stage, tag hierarchy tags are the input to the grammar rules that produce the output entity type, also a tag from the tag hierarchy.

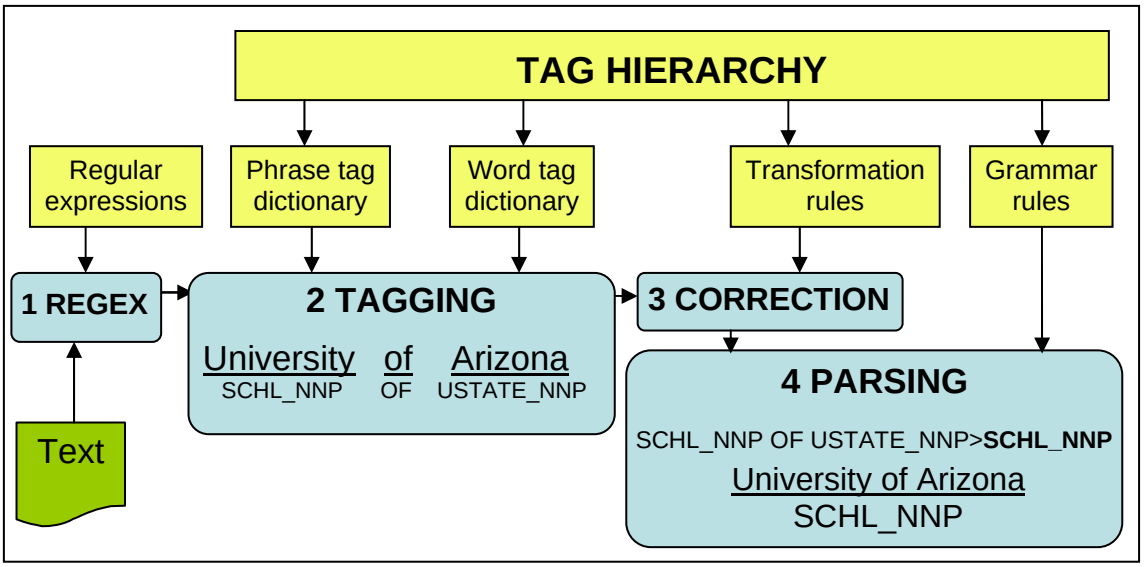

Figure 4.2 – System diagram of the Arizona Entity Finder. The system consists of four main processes, which include the applying of regular expressions, tagging of words and phrases, correction of the tags, and finally parsing the entities.

4.4.1 Combination Syntax-Semantic Tag

The idea of combining syntax and semantic information into one tag is inspired by WordNet, where a word is considered to be a combination of "a lexicalized concept and…a syntactic role." (G. A. Miller, Beckwith, Fellbaum, Gross, & Miller, 1990). In other words, nouns will never be in a synonym set with verbs regardless of the similarity of their semantics. We extend this concept, so that not only are nouns separated from verbs, but are also separated from different types of nouns as well as adjectives. This distinction between nouns is not found in WordNet. The nouns in WordNet are primarily non-proper nouns with names and proper nouns not being that common.

In our representation, each tag is a composite of syntax and semantic information. The tag begins with semantic information and ends with syntax information. For example, the tag "BLDG_NNP" begins with semantic information in the "BLDG" tag,

which means the noun is a type of building. The tag ends with syntax information "NNP" which is the PENN TREE BANK notation for singular proper noun. An example of a phrase that would get tagged with this tag is "Empire State Building." An underscore divides the semantic and syntax portion of the tag. For each semantic portion of a tag, there are many syntactic endings. For example, the tag "BLDG_PL" means the building is a plural non-proper noun. An example of such a noun is "public libraries." In some cases semantic tags used with noun syntax endings are also used with adjective endings "EMOTION_JJ" (for proud) and adverb endings "EMOTION_RB" (for proudly).

In total, ignoring the syntax components, the hierarchy has over 1,200 semantic word classes. 275 verbs classes, 220 adjective classes 100 preposition classes 50 adverb classes, 27 pronoun classes, 26 punctuation classes, 10 number-related classes, 7 determiner classes with the last 450 being primarily noun classes. The verb classes were taken from Beth Levine's verb taxonomy (Levin, 1993). The tags for the preposition class, pronoun class, punctuation class, determiner class, and other closed class categories were largely an enumeration of the words themselves, with each word having a unique tag. The noun classes are an expanding group of tags taken from various sources. Many of the noun tags incorporated are taken from Sekine's tag hierarchy of over 200 entries. In addition to these entity classes, we have added additional noun classes organized by tags corresponding to the U.S. government cabinet level positions. Upper level tags exist for words related commerce, defense, homeland security, education, agriculture, energy, labor, justice, the environment, human health, and housing and urban development. Education is the largest category with 141 different classes. Tags were largely created in

a bottom up approach. For example, we analyzed the all the words in a lexicon of 10,000 organization names. We were able to group many words into general categories such as business sector, business offering, business type, presence, practical characteristics, and prestigious characteristics. These six noun classes became important tags for naming organizational entities.

The tag hierarchy contains multiple inheritance and all the algorithms that match tag sequences to the lexical profiles in the grammar are aware of the hierarchical relationships. Figure 4.3 shows the inheritance of the tag BUSSECTOR_NNP in the noun hierarchy. The inheritance within the hierarchy allows a lexical profile to be expressed at the most general level possible. If the rule input or lexical profile of "BUSSECTOR_NNP BUSTYPE_NNP" existed in the entity grammar for output tag BUSORG_NNP, then this profile would match Optical Inc. because the word "Optical" is of type BUSSECTOR_NNP and "Inc." is of type "BUSTYPE_NNP". However, the profile would miss the entity Optics Inc, which has the profile "BUSSECTOR_NNPS BUSTYPE_NNP". However, using the more general lexical profile of "BUSSECTOR BUSTYPE_NNP" without the syntax ending on BUSSECTOR would recognize both

<NP description="noun phrase">
|----------NN (non-proper noun)
|----------NN_SG (singular non-proper noun)
|-----------COMMERCE_SG
|------------BUSSECTOR_SG
|--------NN_PL (plural non-proper noun)
|-----------COMMERCE_PL
|------------BUSSECTOR_PL
|-----------SEMNOUNS (semantic noun classes)
|----------COMMERCE
|----------**BUSSECTOR**
|----------**BUSSECTOR_NNP**
|----------BUSSECTOR_NNPS
|----------BUSSECTOR_SG
|----------BUSSECTOR_PL
|----------NNPP description="proper noun"
|----------NNP_PHS (plural proper noun)
|----------NNP_PH (singular proper noun)
|----------**BUSSECTOR_NNP**
</NP>

Figure 4.3 – Inheritance for BUSSECTOR_NNP Tag.

entities because of the "is-a" relationship. Table 4.1 shows an example of a lexical profile or input rule for International Business Machines Corp., with the name shown on the first row. The tags for each of the four words in the name are shown on the second row, "PRESENCE_NNPJJ BUSSECTOR BUSOFFERING_NNPS BUSTYPE_NNP". The third row of the table shows other words with the same tag from the lexicon. The fourth row shows the total number of words in the lexicon with that particular tag. Given that the tag profile where in the lexical memory, then not only would International Business Machines Corp. be recognized in text, but also names like "Bay Medical News Co." or "National Marine Services Inc." because those names can be generated from the lexical profile.

Table 4.1 – Lexical profile for International Business Machines Corp.

1. Name	International	Business	Machines	*Corp.*
2. Tag profile	PRESENCE_ NNPJJ	BUSSECTOR	BUSOFFERING _NNPS	*BUSTYPE_ NNP*
3. Other words with the same tag	Bay Coastal Gulf National Plains Regional States Westcoast	bioscience telecommunications Manufacturing Medical Marine Networking Optics Pharmaceutical	Instruments Investments Controls NEWS PRODUCTS Services Solutions Equities	*Inc.* *Hldg* *Co.* *Ind.* *Ltd.* *PLC* *LLC* *Trading*
4. Totals in the lexicon	*87*	*578*	*14*	*262*

4.4.2 Grammar-based Algorithm

The process of entity recognition takes place in three main steps. The first step is tokenization and tagging, the second step is transformation-based tag correction, and the third step applies the entity grammar to input sequences using a matching algorithm. The three steps are illustrated in Figure 4.4 and explained more fully below.

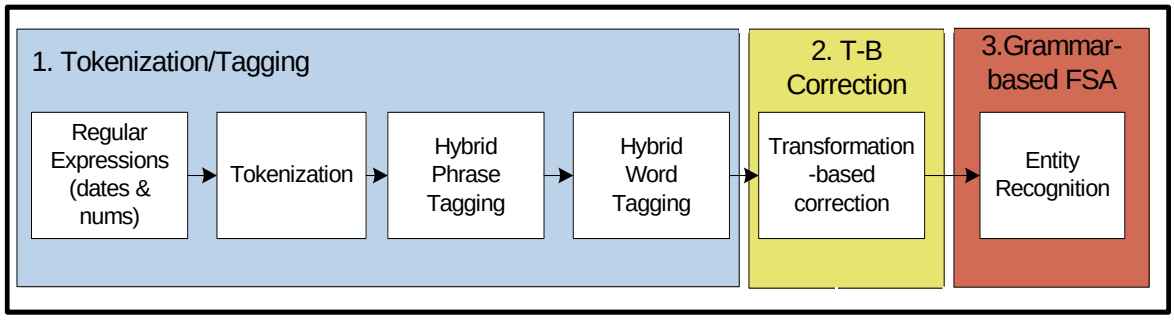

Figure 4.4 – Process of Named Entity Extraction

4.4.2.1 Tokenization and Tagging

The first step in the extraction process is to run 81 regular expressions over the input text. The regular expression identify most dates, times, money, percents, and many ticker symbols. Next, a sentence boundary detection algorithm uses simple rules and a lexicon of over 340 abbreviations to recognize full sentence stops. Phrases are tagged next from a lexicon of 30,000 phrases. The multi-word dictionary is made up primarily of entities, but also contains discourse phrases and multi-word prepositions. Next, remaining words are tagged with their most common tags. An example of the output after the tokenization and tagging process is shown in Figure 4.5. The system considers this to be the first level of tagging. Phrases grouped together can not be broken up.

```
[On the other hand/DISCOURSE] [,/,] [Billy/FNAM_NNP] [would/MD]
[use/CHARACTERIZE_VB] [the/DT] [token/JJ_JJ] [from/FROM] [the/DT][Chucky
Cheese/RESTAURANT_NNP] [machine/NN_SG] [as a means of/PREP]
[remembering/CHARACTERIZE_VBG] [the/DT] [party/EVENT_SG] [./.]
```

Figure 4.5 – Output of the Tokenization and Tagging Step.

4.4.2.2 Transformation-based Correction

Because the most common tag is not always the correct tag, transformation rules are applied to the tags next. Transformation rules include all of Eric Brill's transformation-based learning rules along with additional rules we have generated that are tailored to semantic word sense tagging. Given the tagging output in Figure 4.4, we see that the word "token" has been erroneously tagged as an adjective ("JJ_JJ"). One of the transformation rules in our system is to transform an adjective to a noun if the word is surrounded by tags "DT" and "PREP". The conditions for the transformation are met in

this instance and thus the tag is changed/transformed from an adjective to a noun. Transformation-based rules are run 3 times at various levels of the parse tree. Transformation-based learning has been used previously for sense tagging in addition to part-of-speech tagging (Boufaden, Bengio, & Lapalme, 2004; Brill, 1994; Wilks & Stevenson, 1997).

4.4.2.3 Grammar-based Entity Finding

Once tags have been assigned and all correction checks have been run the tag sequences are compared to the grammar. Sequences of up to 20 tags are compared to the grammar memory base. If no match is found, the 20 tag window is reduced by one without advancing the window and the memory is checked again. After finding a match, the window advances the number of tokens equal to the match. If no match is found and the tag window size has been reduced to 1, the window advances 1 token and the next 20 tags are added to the window. The grammar rules often include the context tags around the entity in addition to the tags internal to the entity. This flexibility helps with disambiguation. Because of the lexical hierarchy inheritance, higher order tags can be used in the grammar rules to improve matching frequency. For example, a profile of "NNP_NNP BUSTYPE_NNP", which is a proper noun followed by a business type word (such as "Inc." or "LLP") should result in a business organization. At the same time however, profiles can also be specified to ignore inherited word classes.

4.4.2.4 Grammar Rule Generation

While tagged corpora are very helpful training resources, the ability to utilize other less costly resources of building up a profile memory is important. In addition to

tagged corpora we utilized lexicons of organizations, locations, and people names to build up the profile memory. The tagger assigns each word in the proper name a tag from the tag hierarchy. The sequence of the tags making up the entity is then added to the grammar memory. There are currently 6,415 input sequences of tag combinations in the profile memory. Of that total, 3,500 were added from profiles generated off lexicons.

4.5 Research Hypothesis

By including lexical semantics and syntax information in each tag and therefore parsing rule, we propose that our language representation is more generalizable than one based on the actual words themselves and more expressive than one based just on syntax tags. We have two hypotheses that we test to evaluate the performance of our combination tag approach. The intention of the tests is first to evaluate whether the combination representation is expressive enough to produce high precision and recall scores in the MUC competition (H1). Secondly, we wish to test whether it is discriminating enough to extract additional entity types as well (H2). At the same time, we want to evaluate whether the representation is generalizable by running the entity extractor on text from a different source, format, and time period without too much additional training (H2).

Hypothesis 1 (H1): The combination entity finder will extract MUC-7 entities (a total of 7) at or above a 90 percent F-measure on the MUC-7 testing data.

Hypothesis 2 (H2): The combination entity finder will extract 14 entities from finance documents at or above a 90 percent F-measure.

4.6 Experimental Design

The experiment consisted primarily of a precision and recall test on two different corpora from different time periods and in different formats.

4.6.1 MUC-7 Documents

For the MUC-7 conference, experts had marked training and testing sets of articles supplied by the New York Times News Service and distributed by the Linguistic Data Consortium (LDC) (DARPA, 1998). We obtained the pre-tagged corpus from the LDC. We trained the entity finder on the 100 training documents. Training consisted of verifying that every word in the training corpus was in our lexicon with the proper combination tag. In addition, the lexical profiles from all the tagged entities in the corpus were added to the grammar memory. Finally, transformation rules were added to insure that words and phrases with multiple tag possibilities were assigned the correct tag given the context of the word. The entity finder was then run on the testing set. The experiment was not blind, as additional types of entries such as “astronomical bodies” were added to the lexicon after seeing they were tagged as locations in the testing corpus. Final output from the algorithm was compared to the human tagged results for precision and recall. Partial credit was given when the entity had an incorrect phrase boundary if the entity type was correct. The F-measure we used contained an equal weighting of precision and recall.

4.6.2 Finance Documents

From a collection of over 50,000 finance documents that had been downloaded from Yahoo! Finance and other sites, 150 web pages were randomly selected. The HTML tags were manually removed from the pages and the text of each article was placed within XML tags, though still not entirely clean. Master's students in Finance and Business tagged 14 different entity types in all the documents using a highlighting interface developed for the purpose. Highlights were replaced with XML tags in the documents. The first 100 finance documents were used for training, similar to the training done with the MUC-7 documents. The last 50 documents were then evaluated for precision, recall and an F-measure of equal precision and recall weighting. Again partial credit was given when an error existed in the phrase boundary if the entity type assigned was correct. This experiment was also not blind, as some lexicon additions targeted areas of poor coverage, such as names from specific ethnic backgrounds.

4.7 Experimental Results

Table 4.2 shows the results from the experiment using the MUC-7 data. The overall F-measure with equal weighting for precision and recall was 90 percent. The entity extractor extracted people and locations at a 91 percent F-measure, while organizations were extracted at only 86 percent F-measure. Considering that organizations were the most common entity type in the MUC-7 corpus, this lower performance was the largest drain on the overall F-measure. Of the numerical entities,

precision was generally higher than recall, much higher in some cases. The algorithm extracted dates with a 94 percent F-measure, time with a 90 percent F-measure, percent

Table 4.2 – Results of Entities Extracted from MUC-7 Documents

	Person	Location	Orgs.	*Date*
Precision	87.4%	87.9%	88.9%	*98.2%*
Recall	94.1%	93.6%	84%	*89.6%*
F-measure	90.6%	91%	86.4%	*93.7%*
	Time	Percent	Money	***TOTAL***
Precision	99.3%	96.1%	92.4%	***90.7%***
Recall	82.8%	99%	84.2%	***89.3%***
F-measure	*90.3%*	*97.5%*	*88.1%*	***90%***

with a 98 percent F-measure and money with an 88 percent F-measure. In the money category, the algorithm had not been trained adequately on foreign currency and therefore performed poorly in that area. In addition, the difference between precision and recall in the numerical categories was larger than the differences in the proper name categories. Overall, compared to others that participated in the MUC-7 conference, the entity finder algorithm performed well.

Of the original 14 participants in the MUC-7 named entity extraction task, three scored above 90 percent F-measure with equal precision and recall weighting. The Language Technology Group had an F-measure of 93.39, the IsoQuest System 1 had a 91.6 percent F-measure, and BBN had a 90.44 percent F-measure. The original annotators had an F-measure around 97 percent.

Table 4.3 shows the precision, recall and F-measure results from extracting entities from Web finance documents. Again F-measure was an equal balance of

precision and recall totals. Overall, the Entity Finder scored 87 percent F-measure, which was lower than the hypothesized 90 percent. Percentage wise, extraction performance of "Other Organizations" and "Other Locations" were significantly lower than other scores with a 55 percent and 64 percent F-measures respectively. Because of the distribution of the types of entities in the document collection, the performance of the "Business Organization" category most impacted overall performance.

Table 4.3 – Results of Entity Extraction from Finance Documents

Entity Types	**Precision**	**Recall**	***F-measure***
Person	81.8% (148/181)	89.2% (148/166)	*85.3%*
Business Organization	83.7% (441/527)	78.2% (441/564)	*80.8%*
Government Organization	81.8% (18/22)	85.7% (18/21)	*83.7%*
Other Organizations (Associations, Funds, Airports)	57.8% (26/45)	53.1% (26/49)	*55.3%*
US Cities / US Townships / US Burroughs	80% (32/40)	80% (32/40)	*80%*
US States	97.1% (33/34)	94.3% (33/35)	*95.7%*
World Cities / World States	94.1% (32/34)	82.15 (32/39)	*87.7%*
Country / Multi-Country Region	97.5% (77/79)	90.6% (77/85)	*93.9%*
Other Location (Counties, Lakes, Rivers, Mines)	56.4% (22/39)	71% (22/31)	*62.9%*
Date	95.4% (289/303)	98.6% (289/293)	*97.0%*
Time	97.8% (44/45)	100% (44/44)	*98.9%*
Percent	99.4% (83.5/84)	99.4% (83.5/84)	*99.4%*
Money	93.0% (120/129)	99.2% (120/121)	*96.1%*
Ticker	92.7% (76/82)	93.8% (76/81)	*93.3%*
TOTAL	***88%***	***87%***	***87.4%***

4.8 Discussion

Despite the Arizona Entity Finder achieving lower than the 90 percent F-measure hypothesized, there was some promising performance, especially in light of some formatting challenges of handling web documents. For example, there was greater use of capitalization throughout the pages which brought scores down. Erroneously extracting navigation text also accounted for some errors. For example, we extracted Click as an organization from the "Click Here" text. Labels for tables and figures and miscellaneous headings also caused some errors. Figure 4.6 shows a bar graph of the distribution of entity types found in the stock news text. The "Business Organization" category

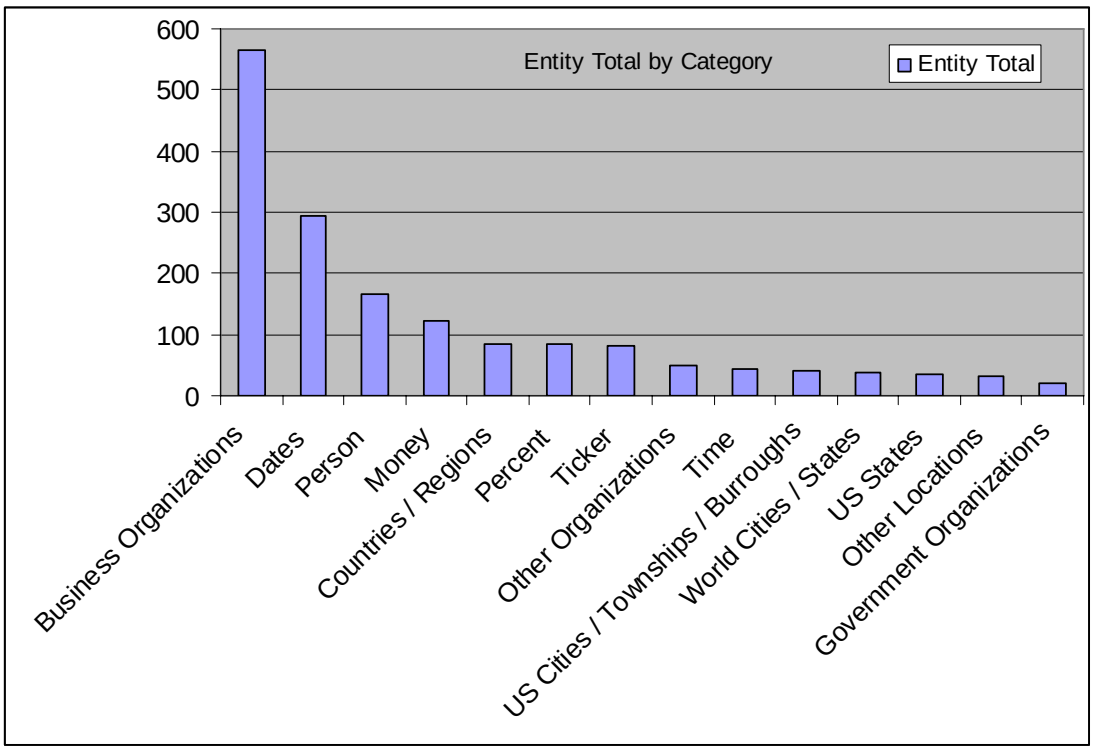

Figure 4.6 – Distribution of Entities in the Stock News Text from Yahoo

accounted for 34 percent of all entities. Scoring an 81 percent F-measure in that category hurt our overall performance. Also, we incorrectly categorized 11 "Business Organization" entities as "Other Organizations" so that error was costly in both categories. We also missed several entities that were simple not in our dictionaries and the syntax information assigned was not enough to correctly identify the entity. Some locations names missing from our dictionary included the world cities of Bishkek, Osh, and Holzkirchen. Missing business names included Groupe Finuchem, Hutchison Whampoa, Sage, Boots, KPN, and PCCW. Other difficulties involved the business Curlew Lake Resources Inc. which was referred to as Curlew Lake many other times in the document creating confusion for the algorithm. Phrases where ticker symbols overlapped with business names, such as with IBM and DELL, or when the ticker was referred to as the company itself were difficult for the algorithm.

4.9 Conclusion and Future Direction

Algorithms that utilize the actual words in a document have a rich representation and have achieved high results in identifying named entities in text. However, because word usage varies greatly between domains, such algorithms must be trained on large amounts of additional tagged text in order to move to a new domain. We have developed a hierarchy of combination syntax-semantic tags to represent a document's tokens as opposed to using the words themselves. Based on this new representation, lexical profiles of entities were created and added to a grammar memory. The results achieved on the MUC-7 dataset are encouraging because the performance is near the top and they were obtained with the engineering efforts of one person using the MUC-7 training corpus and

lists of lexicons available on the Internet. Increasing the amount of the training corpora should improve results.

The performance of the Arizona Entity Finder in the finance domain is also encouraging. First off, the Web documents themselves posed some unique challenges. The use of capitalization in the Web documents was more inconsistent than that in the MUC-7 data set. In addition, while the MUC documents contained primarily sentences of text, the Web finance documents contained more text intermingled with tables and lists. This variety of formatting made the entity identification task more difficult. Finally, increasing the number of categories from seven to fourteen increased the difficulty of the task. So despite the lower F-measure scores, the results show some promise. Some pruning of rules will be required to improve the precision totals while additional focused training should help improve the recall performance.

Future direction for this research is to further automate the knowledge engineering to speed training on new domains. The existing grammar rules could be used to bootstrap the training process and have user's intervene only when necessary to make corrections to tagged corpora. We would also like to focus training on word-sense disambiguation. As the grammar memory continues to grow, the number of errors from having the wrong tag assigned will be greater than those caused by missing grammar rules. We would also like to explore how entity extraction technologies could be used to improve search and clustering techniques.

5 CONTRIBUTIONS AND FUTURE DIRECTIONS

The volume of digitized unstructured text is growing rapidly. Text mining systems that successfully assist users in finding trends, finding unexpected relationships, or simply finding needed documents or events can save both time and money for organizations. Improvements in the finding and analysis of text are driven by improvements in the underlying processing of documents. This work has focused on improving document processing. Three main contributions result from this work. First, different summarization techniques have been linked to user performance in different information seeking tasks. Second, the use of combination syntax-semantic grammars has been tested in both relation extraction and entity extraction. The combination approach has shown some robustness in being able to process various document types. Third, the combination grammar has been successful in recognizing a greater number of entities.

5.1 Matching User Tasks to Information Needs

More document processing is not always better when it comes to automatic text summarization. In Chapter 2, we presented an algorithm that chooses summary sentences by balancing document structure and local lexical semantics to produce a generic summary. We pitted this summary against one that simply compared text sequences to query terms submitted by users to create a query-biased summary. In information seeking scenarios, users performed better using the generic summary when browsing and performed better using the query-biased summary when searching. This research

contributed to the understanding of user tasks by relating performance on tasks to types and content of summaries used as document proxy.

5.2 Combination Parsing to Improve Algorithm Coverage

Improvements in document processing techniques can improve algorithm performance on a domain-specific subset of documents. Improvements can also allow existing performance to be achieved in a greater range of documents. The focus of this work has been on the later. With the amount and variety of text only growing, approaches to document processing must be viable in multiple domains. A contribution of this work is the idea of combining syntax with lexical semantics into a single generative grammar in order to improve grammar coverage. Existing systems had utilized lexical semantic grammars and had achieved high performance. Evaluations of these systems had been noticeably small or focused on particular domains. Chapter 3 demonstrates how a combination syntax-semantic grammar could be implemented for relation extraction and then evaluates the system using a larger document set than typically reported. Chapter 4 demonstrates a substantial extension of the combination grammar to include over 10,000 total tags in order to extract named entities from text. The entity extractor was evaluated using a standard testing corpus, but then also in a more modern news corpus downloaded from Yahoo. The entity finder was able to achieve comparable performance on the two corpora. The demonstrated coverage of the combination parser is a contribution of this work.

5.3 Combination Parsing to Increase Number of Entities

In addition to increasing coverage, the combination grammar was utilized to increase the types of entities recognized. Chapter 4 describes the algorithm and the process whereby the types of entities extracted is increased from seven to fourteen. Classification algorithm performance typically declines as the number of output classes increases. Yet distinguishing government, business, and terrorism organizations can be quite useful in different applications, such as risk monitoring. In this work, an existing entity hierarchy was extended by hundreds of entity types. An approach to naming the entities was presented as well as a representation that was able to accurately distinguish between numerous entity types. The system design and subsequent test requiring the extraction of 14 different entity types is a contribution of this work.

5.4 Relevance to Business and Managed Organizations

It has been estimated that 80 percent of a companies knowledge stores are in unstructured text. Improved text processing to support text finding and text analysis could have a big impact on business. Most of the finding process in organizations occurs on the intranet. Unlike the pages on the Internet, intranet pages are not heavily linked, which makes retrieval more difficult. Ranking algorithms must rely more on the content of each page for weighting. Systems that can better understand document content in terms of recognizing entities and relationships can do a better job ranking. A BAE Systems study found that 25 percent of a project's cost was in searching for best practices information.

They subsequently implemented a document processing system from Autonomy to improve the finding process and have recognized a large cost savings (Hoffman, 2002).

In addition to improving the finding and retrieving of intranet content, improved processing can greatly benefit the analysis process. We review three analysis tasks where improved text processing can impact business analysis processes of interest.

5.4.1 Reputation Mining

In today's environment of corporate scandals and accounting fraud, companies are concerned with how they are perceived. Companies are also interested in knowing how their products and trademarks are perceived in the market both before and after an ad campaign or product changes. This type of analysis has been referred to as media impact analysis. Text mining has been applied to this problem of reputation mining (Fan, Wallace, Rich, & Zhang, 2006; Morinaga, Yamanishi, Tateishi, & Fukushima, 2002). This type of analysis requires in-depth processing of a large document set. Document processing techniques must be able to identify instances of products or company names in text and extract the sentiment expressed with regard to that particular product or company. A combination grammar with the ability to recognize many types of entities and other semantic categories is well-suited for these types of tasks. In this type of analysis, not only are entities important, but the sentiment expressed with regard to the entities must be captured and aggregated.

In addition to product reputation analysis, many customer service problems are received electronically. Being able to automatically process and aggregate customer complaints allows businesses to focus resources and act quickly. Text mining has also

been cited as a valuable tool to provide customer support for Frequently Asked Questions (FAQs).

5.4.2 Environmental Scanning

Business markets are more dynamic than ever with the increasing global nature of the economy. Businesses must evolve to deal with the changing global economic conditions and also be aware of new competitive threats coming from anywhere in the world. Textual information is increasingly available online that documents large business transactions, business developments, emerging markets, and mergers and acquisitions. Such information once largely available through subscription services is now more freely distributed thanks to simple syndication formats and Web portals looking to increase site traffic. Text mining has been proposed as a solution to automatically extract and analyze environmental and competitive information (Fan, Wallace, Rich, & Zhang, 2006; S. S. L. Tan, Teo, Tan, & Wei, 1998). Central to the environmental scanning process is identifying the companies that are involved in an environment or market and the relationships that exist between different companies, products, and markets. Thus it is very important to have as accurate and complete extraction of entities and relationships as possible from the text before proper analysis can take place.

Various types of information might be relevant to one's business environment depending on the industry. In a study conducted on a cereal products supply chain, Sohal and Perry created a model of relevant environmental information which included globalization and demand trends, industry complexity and realignment, power relationships, delivery requirements, competitive supply chain requirements, the

information economy, financial reporting requirements, freight difficulties, supply chain labor challenges, and climatic conditions impacting crop yields (Sohal & Perry, 2006). Information related to these relevant environmental conditions is readily available on the Web and could be targeted by text mining software to facilitate analysis.

Tools for environmental scanning and processing of unstructured data are already being seen in the marketplace. IBM for one has unified over 20 years of related research into a project with a $100 million budget named Web Fountain (Olsen, 2004). Web Fountain indexes about 250 million web pages weekly and focuses primarily on harvesting emerging trends from the data. Their focus is not on high-ranking pages as determined by the page rank algorithm, but rather those that are ranked low. They are interested in grassroots pages that are not necessarily well linked but contain interesting commentary and sentiments that have not yet reached the mass market. Web Fountain includes techniques for entity extraction and methods for extracting relationships between the entities. Customers of Web Fountain have included Citibank, British Petroleum, and Factiva. The need to intelligently process unstructured text in order to stay competitive is only increasing in the business world. IDC has estimated that the market providing services for unstructured text management will be $9.72 billion by 2006, up from $6.46 billion in 2004.

5.4.3 Monitoring Systems

Monitoring system research is being done that attempts to detect when employees may be an insider threat to a company (Symonenko et al., 2004). Insider threats may be exposing company patents or trade secrets or leaking company strategic plans through

written communication. These predictions are made based on the employee communication patterns and e-mail content. Such techniques require robust text processing tools for which a combination grammar would be useful.

5.5 Future Directions

The main future direction for this research is to investigate how the information extraction tools presented in this work can benefit the finding and analysis stages of the text mining process. For example, in the finding stage, I want to use entity and relation extraction techniques to perform multi-document summarization on FASB accounting standards documents, SEC document filings, and tax laws. These three collections of documents are not heavily hyperlinked and thus rely on their content for accurate retrieval. In the analysis stage, I am interested in supporting the research work attorneys must do when receiving gigabytes of e-mail communication to investigate for improper employee communication. Utilizing entity and relation extraction techniques, I am interested in clustering content by relationships and by e-mail sender. I am interested in uncovering patterns of relationship content in e-mail communication

REFERENCES

Aberdeen, J., Burger, J., Day, D., Hirschman, L., Robinson, P., & Vilain, M. (1995). *Description of the Alembic system used for MUC-6.* Paper presented at the Sixth Message Understanding Conference.

Abney, S. (1991). Parsing by chunks. In *Principle-Based Parsing*. Dordrecht: Kluwer Academic Publishers.

Ackoff, R. L. (1989). From Data to Wisdom. *Journal of Applied System Analysis, 16*, 3-9.

Aone, C., Halverson, L., Hampton, T., & Ramos-Santacruz, M. (1998). *SRA: Description of the IE2 System Used for MUC-7.* Paper presented at the Message Understanding Conference - 7.

Aone, C., Okurowski, M. E., Gorlinsky, J., & Larsen, B. (1999). A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization* (pp. 71-80). Cambridge: The MIT Press.

Appelt, D., Hobbs, J. R., Bear, J., Isreal, D., Kameyama, M., Kehler, A., et al. (1995). *SRI International FASTUS system: MUC-6 test results and analysis.* Paper presented at the Sixth Message Understanding Conference, Columbia, Maryland.

Barzilay, R., & Elhadad, M. (1999). Using lexical chains for text summarization. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization*. Cambridge: The MIT Press.

Bikel, D., Miller, S., Schwartz, R., & Weischedel, R. (1997). Nymble: a high-performance learning name-finder. (http://citeseer.nj.nec.com/bikel97nymble.html).

Black, E., Jelinek, F., Lafferty, J., Magerman, D. M., Mercer, R., & Roukos, S. (1992). *Towards history-based grammars: Using richer models for probabilisitic parsing.* Paper presented at the DARPA Speech and Natural Language Workshop.

Black, W., Rinaldi, F., & Mowatt, D. (1998). *FACILE: Description of the NE System Used for MUC-7.* Paper presented at the Seventh Message Understanding Conference.

Blagosklonny, M. V., & Pardee, A. B. (2002). Conceptual biology: Unearthing the gems. *Nature, 416*(6879), 373.

Boguraev, B., & Kennedy, C. (1997). *Salience-based Content Characterization of Text Documents.* Paper presented at the Proceedings of the Workshop on Intelligent Scalable Text Summarization at the ACL/EACL Conference, Madrid, Spain.

Börner, K., & Chen, C. (2002). *Visual interfaces to digital libraries.* Paper presented at the Second ACM/IEEE-CS Joint Conference on Digital Libraries, Portland, OR.

Borthwick, A., Sterling, J., Agichtein, E., & Grishman, R. (1998). *Description of the MENE Named Entity System Used in MUC-7.* Paper presented at the Seventh Message Understanding Conference, Fairfax, VA.

Boufaden, N., Bengio, Y., & Lapalme, G. (2004). *Extended semantic tagging for entity extraction.* Paper presented at the LREC Workshop, Lisbon, Portugal.

Brandow, R., Mitze, K., & Rau, L. (1994). Automatic Condensation of Electronic Publications by Sentence Selection. *Information Processing and Management, 31*(5).

Brill, E. (1993). *A Corpus-Based Approach to Language Learning.* Unpublished PhD, University of Pennsylvania, Philadelphia.

Brill, E. (1994). *Some Advances in Transformation-Based Part of Speech Tagging.* Paper presented at the National Conference on Artificial Intelligence.

Brin, S., & Page, L. (1998, April 1998). *The Anatomy of a Large-Scale Hypertext Web Search Engine.* Paper presented at the 7th international world wide web conference, Brisbane, Australia.

Buchholz, S. N. (2002). *Memory-Based Grammatical Relation Finding.* Unpublished PhD, University of Tilburg, Tilburg.

Carbonell, J., & Goldstein, J. (1998). *The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries.* Paper presented at the SIGIR, Melbourne, Austrailia.

Carmel, E., Crawford, S., & Chen, H. (1992). Browsing in Hypertext: A Cognitive Study. *Systems, Man and Cybernetics, IEEE Transactions on, 22*(5), 865-884.

Chen, H. (2001). *Knowledge Management Systems - A Text Mining Perspective*. Tucson: University of Arizona.

Chen, H., Houston, A. L., Sewell, R. R., & Schatz, B. R. (1998). Internet Browsing and Searching: User Evaluations of Category Map and Concept Space Techniques. *Journal of the American Society for Information Science, 49*(7), 582-603.

Chen, H., Schufels, C., & Orwig, R. (1996). Internet Categorization and Search: A Self-Organizing Approach. *Journal of Visual Communication and Image Representation, 7*(1), 88-102.

Childs, L., Brady, D., Guthrie, L., Franco, J., Valdes-Dapena, D., Reid, W., et al. (1995). *LOUELLA Parsing, An NLToolset System for MUC-6.* Paper presented at the Sixth Message Understanding Conference, Columbia, Maryland.

Choi, F. Y. Y. (2000). *Advances in domain independent linear text segmentation.* Paper presented at the NAACL '00, Seatle, USA.

Chomsky, N. (1957). *Syntactic Structures*: The Hague: Mouton & Co.

Church, K. w. (1988). *A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text.* Paper presented at the 2nd Conference on Applied Nat- ural Language Processing.

Ciravegna, F., & Lavelli, A. (1999). *Full Text Parsing using Cascades of Rules: an Information Extraction Perspective.* Paper presented at the EACL.

Cove, J. F., & Walsh, B. C. (1988). Online text retrieval via browsing". *Information Processing and Management, 24*(1), 31-37.

Cowie, J. (1995). *Description of the CRL/NMSU Sytems Used for MUC-6.* Paper presented at the Sixth Message Understanding Conference, Columbia, Maryland.

Cowie, J., & Lehnert, W. (1996). Information Extraction. *Communications of the ACM, 39*(1), 80-91.

Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A., & Mazo, I. (2004). Extracting human protein interactions from MEDLINE using a full-sentence parser. *Journal of Bioinformatics, 20*(5), 604-611.

DARPA. (1998). *Proceedings of the 7th Message Understanding Conference (MUC-7)*, Washington, D.C.

Day, D., Robinson, P., Vilain, M., & Yeh, A. S. (1998). *Description of the Alembic System as used in MUC-7.* Paper presented at the Seventh Message Understanding Conference, Fairfax, VA.

DeJong, G. F. (1979). Prediction and substantiation: A new approach to natural language processing. *Cognitive Psychology, 3*, 251-273.

DeJong, G. F. (1982). An overview of the FRUMP system. In W. G. Lehnert & M. H. Ringle (Eds.), *Strategies for Natural Language Processing* (pp. 149-176). Hillsdale, NJ: Erlbaum.

Edmundson, H. P. (1969). New Methods in Automatic Extracting. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization* (pp. 23-42). Cambridge: The MIT Press.

Etzioni, O. (1996). The World Wide Web: Quagmire or Goldmine? *Communications of the ACM, 39*(11), 65-68.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the Power of Text Mining. *Communications of the ACM, 49*(9), 77-82.

Feldman, R., & Dagan, I. (1995, August 20-21). *Knowledge discovery in textual databases (KDT).* Paper presented at the First International Conference on Knowledge Discovery and Data Mining, Montreal, Canada.

Firmin, T., & Chrzanowski, M. J. (1999). An Evaluation of Automatic Text Summarization Systems. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization* (Vol. 325-336). Cambridge: The MIT Press.

Friedman, C., Kra, P., Yu, H., Krauthammer, M., & Rzhetsky, A. (2001). GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Journal of Bioinformatics, 17*(1), S74-S82.

Gaizauskas, R., Demetriou, G., Artymiuk, P., & Willett, P. (2003). Protein Structures and Information Extraction from Biological Texts: The PASTA System. *Journal of Bioinformatics, 19*(1), 135-143.

Goldstein, J., Kantrowitz, M., Mittal, V., & Carbonell, J. (1999). *Summarizing Text Documents: Sentence Selection and Evaluation Metrics.* Paper presented at the 22nd International Conference on Research and Development in Information Retrieval.

Greene, S., Marchionini, G., Plaisant, C., & Shneiderman, B. (2000). Previews and Overviews in Digital Libraries: Designing Surrogates to Support Visual Information Seeking. *Journal of the American Society for Information Science, 51*(4), 380-393.

Hafner, C. D., Baclawski, K., Futrelle, R. P., Fridman, N., & Sampath, S. (1994). Creating a knowledge base of biological research papers. *ISMB, 2*, 147-155.

Harter, S. P. (1992). Psychological Relevance and Information Science. *Journal of the American Society for Information Science, 43*(9), 602-615.

Hearst, M. (1999). User interfaces and visualization. In R. Baeza-Yates & B. Ribeiro-Neto (Eds.), *Modern information retrieval* (pp. 257-339). New York: ACM Press.

Hearst, M. A. (1997). Segmenting Text into Multi-Paragraph Subtopic Passages. *Computational Linguistics*(23(1)), 33-64.

Hearst, M. A. (1999). User Interfaces and Visualization. In *Modern Information Retrieval*. Harlow, UK: Addison Wesley.

Hearst, M. A., Elliot, A., English, J., Sinha, R., Swearingen, K., & Yee, K.-P. (2002). Finding the Flow in Web Site Search. *Communications of the ACM, 45*(9), 42-49.

Hersh, W., Pentecost, J., & Hickam, D. (1996). A Task-Oriented Approach to Information Retrieval Evaluation. *Journal of the American Society for Information Science, 47*(1), 50-56.

Hobbs, J., Appelt, D., Bear, J., Israel, D., Kameyama, M., Stickel, M., et al. (1996). FASTUS: Extracting Information from Natural Language Texts. In E. Roche & Y. Schabes (Eds.), *Finite State Devices for Natural Language Processing*: MIT Press.

Hoffman, T. (2002, October 14, 2002). In The Know. *Computerworld, October,* 42.

Houston, A. L., Chen, H., Schatz, B. R., Hubbard, S. M., Sewell, R. R., & Ng, T. D. (2000). Exploring the use of concept spaces to improve medical information retrieval. *Decision Support Systems, 30*, 171-186.

Hovy, E., & Lin, C.-Y. (1999). Automated text summarization in SUMMARIST. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization* (pp. 81-94). Cambridge: The MIT Press.

Huber, G. (1991). Organizational Learning: The Contributing Processes and the Literatures. *Organizational Science, 2*(1), 88-115.

Huffman, S. (1995). *Learning information extraction patterns from examples.* Paper presented at the IJCAI-95 Workshop on new approaches to learning for natural language processing.

Jenssen, T.-K., Laegreid, A., Kmorowski, J., & Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expresion. *Nature Genetics, 28*, 21-28.

Jing, H., Barzilay, R., McKeown, K., & Elhadad, M. (1998). *Summarization Evaluation Methods: Experiments and Analysis.* Paper presented at the AAAI Spring Symposium on Intelligent Text Summarization.

Jurafsky, D., & Martin, J. H. (2000). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Upper Saddle River: Prentice Hall.

Kan, M.-Y., Klavans, J. L., & McKeown, K. R. (1998). *Linear segmentation and segment significance.* Paper presented at the 6th International Workshop of Very Large Corpora (WVLC-6), Montreal, Quebec, Canada.

Katz, S. M. (1996). Distribution of Content Words and Phrases in Text and Language Modelling. *Natural Language Engineering, 2*(1), 15-59.

Kim, J.-T., & Moldovan, D. I. (1993). *PALKA: A System for Lexical Knowledge Acquisition.* Paper presented at the ACM CIKM, Washington D.C.

Kleinberg, J. (1999). Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM), 46*(5), 604--632.

Krupka, G. R., & Hausman, K. (1998). *Description of NetOwl(TM) Extractor System as Used in MUC-7.* Paper presented at the Seventh Message Understanding Conference, Fairfax, VA.

Kuhlthau, C. C. (1991). Inside the Search Process: Information Seeking from the User's Perspective. *Journal of the American Society for Information Science, 42*(5), 361-371.

Kupiec, J., Pedersen, J., & Chen, F. (1995). *A Trainable Document Summarizer.* Paper presented at the Proceedings of the 18th ACM-SIGIR Conference.

Lam-Adesina, A. M., & Jones, G. J. F. (2001, September 9-12). *Applying Summarization Techniques for Term Selection in Relevance Feedback.* Paper presented at the SIGIR, new Orleans, Louisiana, USA.

Landauer, T. K., Egan, D. E., Remde, J. R., Lesk, M., Lochbaum, C. C., & Ketchum, D. (1993). Enhancing the usability of text through compuer delivery and formative evaluation: the SuperBook project. In C. McKnight, A. Dillon & J. Richardson (Eds.), *Hypertext: A Psychological Perspective* (pp. 71-136): Ellis Horwood.

Levin, B. (1993). *English Verb Classes and Alternations*. Chicago: The University of Chicago Press.

Liddy, E. D. (1998). Enhanced Text Retrieval Using Natural Language Processing. *American Society for Information Science*(April/May 1998), 14-16.

Lin, D. (1998). *Using Collocation Statistics in Informatin Extraction.* Paper presented at the Seventh Message Understanding Conference (MUC-7), Fairfax, VA.

Luhn, H. P. (1958). The Automatic Creation of Literature Abstracts. In I. Mani & M. T. Maybury (Eds.), *Advances in Automatic Text Summarization* (pp. 15-22). Cambridge: The MIT Press.

Mani, D., House, D., Klein, G., Hirschman, L., Obrst, L., Firmin, T., et al. (1998). *The tipster summac text summarization evaluation: Final report*: DARPA.

Mani, I., & Maybury, M. T. (Eds.). (1999). *Advances in Automatic Text Summarization*. Cambridge: The MIT Press.

Marchionini, G., & Shneiderman, B. (1988). Finding Facts vs. Browsing Knowledge in Hypertext Systems. *Computer, 21*(1), 70-79.

Marcu, D. (2000). *The Theory and Practice of Discourse Parsing and Summarization*. Boston, MA: MIT Press.

Marshall, B., McDonald, D., Chen, H., & Chung, W. (2004). EBizPort: Collecting and Analyzing Business Intelligence Information. *JASIST*.

Marshall, B., Su, H., McDonald, D., & Chen, H. (2006). Aggregating automatically extracted biological relations. *IEEE Transactions on Information Technology in Biomedicine, 10*(1), 100-108.

McDonald, D., & Chen, H. (2002). *Using Sentence-selection Heuristics to Rank Text Segments in TXTRACTOR.* Paper presented at the Second ACM/IEEE-CS JCDL, Portland, Oregon, USA.

McDonald, D., Chen, H., Su, H., & Marshall, B. (2004). Extracting gene pathway relations using a hybrid grammar: The Arizona relation parser. *Bioinformatics*.

Medicine, N. L. o. (2005). Fact Sheet MEDLINE. Retrieved October 24, 2006, from http://www.nlm.nih.gov/pubs/factsheets/medline.html

Mikheev, A., Grover, C., & Moens, M. (1998, April 29, 1998). *Description of the LTG system used for MUC-7.* Paper presented at the Seventh Message Understanding Conference, Fairfax, Virginia.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography, 3*(4), 235-234.

Miller, L. A. (1995). *Description of the SAIC DX System as Used for MUC-6.* Paper presented at the Sixth Message Understanding Confernece, Columbia, Maryland.

Minel, J.-L., Nugier, S., & Piat, G. (1997). *How to appreciate the quality of automatic text summarization.* Paper presented at the ACL/EACL '97 Workshop on Intellgent Scalable Text Summarization.

MIPT. (2006). Terrorism Knowledge Base.

Moldovan, D., Cha, S., Chung, M., Hendrickson, K., Kim, J., & Kowalski, S. (1992). *Description of the SNAP system used for MUC-4.* Paper presented at the Fourth Message Understanding Conference.

Morarescu, P., & Harabagiu, S. (2004). *NameNet: a Self-Improving Resource for Name Classification.* Paper presented at the 4th International Conference on Language Resources and Evaluation, Lisbon, Portugal.

Morinaga, S., Yamanishi, K., Tateishi, K., & Fukushima, T. (2002). *Mining product reputations on the web.* Paper presented at the SIGKDD 02, Edmonton, Alberta, Canada.

Morris, A., Kasper, G., & Adams, D. (1992). The Effects and Limitations of Automatic Text Condensing on Reading Comprehension Performance. *Information Systems Research, 3*(1).

Nomoto, T., & Matsumoto, Y. (2001, September 9-12, 2001). *A New Approach to Unsupervised Text Summarization.* Paper presented at the SIGIR, New Orleans, LA, USA.

Nonaka, I. (1994). A Dynamic Theory of Organizational Knowledge Creation. *Organizational Science, 5*(1), 14-37.

Novichkova, S., Egorov, S., & Daraselia, N. (2003). MedScan, a natural language processing engine for MEDLINE abstracts. *Journal of Bioinformatics, 19*(13), 1699-1706.

Ohta, T., Tateisi, Y., Hideki, M., & Jun'ichi, T. (2002). *The Genia Corpus: an annotated research abstract corpus in molecular biology domain.* Paper presented at the Human Language Technology Conference, San Diego, CA, USA.

Olsen, S. (2004). IBM sets out to make sense of the Web, *Cnet News*: CNET.

Over, P., & Yen, J. (2004, May 6-7, 2004). *An Introduction to DUC 2004: Intrinsic Evaluation of Generic News Text Summarization Systems.* Paper presented at the Document Understanding Workshop presented at HLT/NAACL, Boston, MA.

Park, J. C., Kim, H. S., & Kim, J. J. (2001). Bidirectional Incremental Parsing for Automatic Pathway Identification with Combinatory Categorical Grammar. *Pacif. Symp. Biocomp., 6*, 396-407.

Pirolli, P., Card, S. K., & Wege, M. (2001). *Visual Information Foraging in a Focus + Context Visualization.* Paper presented at the ACM Conference on Human Factors in Computing Systems.

Plaisant, D., Carr, D., & Shneiderman, B. (1995). Image-browser taxonomy and guidelines for designers. *IEEE Software, 12*(2), 21-32.

Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M., & Cochran, B. (2002). *Robust relational parsing over biomedical literature: extracting inhibit relations.* Paper presented at the Pacific Symposium on Biocomputing, Hawaii.

Radev, D. R., Jing, H., & Budzikowska, M. (2000, April). *Centroid-based summarization of mulitiple documents: sentence extraction, utility-based evaluation, and user studies.* Paper presented at the ACL/NAAL Workshop on Summarization, Seatle, WA.

Riloff, E. (1993). Automatically constructing a dictionary for information extraction tasks. *Proceedings of the 11th National Conference on Artificial Intelligence*, 811-816.

Rindflesch, T. C., Tanabe, L., Weinstein, J. N., & Hunter, L. (2000). EDGAR: extraction of drugs, genes and relations from the biomedical leterature. *Pac. Symp. Biocomput.*, 517-528.

Robb, D. (2004). Text mining tools take on unstructured data. *Computerworld*.

Salton, G., Wong, A., & Yang, C. S. (1975). A Vector Space Model for Automatic Indexing. *Communications of the ACM, 18*(11), 613-620.

Sanderson, M. (1998). *Accurate user directed summarization from existing tools.* Paper presented at the Conference on Information and Knowledge Management, Bethesda, MD, USA.

Schamber, L., Eisenberg, M., & Nilan, M. (1990). A Re-examination of Relevance: Towards a Dynamic, Situational Definition. *Information Processing & Management, 26*(6), 755-776.

Schank, R. (1972). Conceptual dependency: Theory of Natural Language Understanding. *Cognitive Psychology, 3*(4), 532-631.

Sekimisu, T., Park, H., & Tsujii, J. (1998). Identifying the interaction between genes and gene products based on frequently seen verbs in Medline abstracts. *Genome Inform*, 62-71.

Sekine, S., & Nobata, C. (2003). *Definition, dictionaries and tagger for Extended Named Entity Hierarchy.* Paper presented at the Proceedings of the LREC 2003.

Soderland, S., Fisher, D., Aseltine, J., & Lehnert, W. (1995). *Crystal: Inducing a conceptual dictionary.* Paper presented at the 14th International Joint Conference on Artificial Intelligence (IJCAI-95).

Sohal, A. S., & Perry, M. (2006). Major business-environment influences on the cereal products industry supply chain: An Australian study. *International Journal of Physical Distribution & Logistics, 36*(1), 36-50.

Srinivasan, P. (2004). Text mining: Generating hypothesis from MEDLINE. *Journal of the American Society for Information Science and Technology, 55*(5), 396.

Strzalkowski, T., Wang, T. J., & Wise, B. (1998). *A Robust Practical Text Summarization.* Paper presented at the Proceedings of the AAAI-98 Spring Symposium on Intelligent Text Summarization.

Swanson, D. R. (1988). Migraine and Magnesium: Eleven neglected connections. *Perspectives in Biology and Medicine, 31*, 526-557.

Swanson, D. R., Smalheiser, N. R., & Bookstein, A. (2001). Information discovery from complementary literatures: Categorizing viruses as potential weapons. *Journal of the American Society for Information Science and Technology, 52*(10), 797-812.

Symonenko, S., Liddy, E. D., Yilmazel, O., Del Zoppo, R., Brown, E., & Downey, M. (2004). Semantic Analysis for Monitoring Insider Threats. In *Intelligence and Security Informatics*. Berlin / Heidelberg: Springer

Tan, A.-H. (1999). *Text Mining: The state of the art and the challenges.* Paper presented at the PAKDD workshop on Knowledge Discovery from Advanced Databases, Beijing, China.

Tan, S. S. L., Teo, H.-H., Tan, B. C. Y., & Wei, w.-K. (1998). *Environmental scanning on the Internet.* Paper presented at the international conference on Information systems, Helsinki, Finland.

Teufel, S., & Moens, M. (1999). *Sentence Extraction as a Classification Task.* Paper presented at the Workshop on Intelligent Scalable Summarization ACL/EACL Conference, Madrid, Spain.

Thomas, J., Milward, D., Ouzounis, C., Pulman, S., & Carroll, M. (2000). Automatic extraction of protein interactions from scientific abstracts. *Pac. Symp. Biocomput.*, 510-552.

Tolle, K. M., & Chen, H. (2000). Comparing noun phrasing techniques for use with medical digital library tools. *Journal of the American Society for Information and Science, 51*(4), 352-370.

Welge, M. (1998). Analytical and Visual Data Mining.

Wilks, Y., & Stevenson, M. (1997). *Sense Tagging: Semantic Tagging with a Lexicon.* Paper presented at the SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What and How?, Washington D.C.

Yakushiji, A., Tateisi, Y., Miyao, Y., & Tsujii, J. (2001). Event Extraction from Biomedical Papers Using a Full Parser. *Pacif. Symp. Biocomp., 6*, 408-419.