The Chronological Information Extraction System (CHESS)

Paul O'Neil
Air Force Research Laboratory
Rome, NY 13441-4114
oneilp@rl.af.mil

Woojin Paik
TextWise, LLC
Syracuse, NY 13244
woojin@textwise.com

ABSTRACT

The Chronological information Extraction SyStem (CHESS) performs two main functions to provide the user with rich contextual, time-stamped information about relevant people, organizations, countries, or any other named entities. First, it extracts information about named entities and compiles a chronological outline. Second, it acts as a question answering system, allowing the user to find information of interest. In this capacity, the knowledge automatically extracted by CHESS can be examined in different information access modes:

1) as a browser, for information seekers who do not have specific queries;
2) as a question-answering system, for users with specific queries.

Information is extracted from free text utilizing natural language processing techniques. CHESS takes advantage of the common practice of writers placing information-rich linguistic constructions in close proximity to named entities. First, the proper name is recognized, and then the surrounding information is used to develop relationships to other names entities. In this manner, concept-relation-concept (CRC) triples are formed and stored in an object database. Complex histories of events in relation to a given named entity are constructed from these CRC triples, utilizing techniques from conceptual graph theory.

1. INTRODUCTION

Ever-increasing volumes of electronic information are becoming available to analysts and users of information systems. The ability to effectively make use of this information is becoming hampered by the sheer volume of information available. This explosion has highlighted the need for more sophisticated and efficient extraction and retrieval tools to exploit the available information.

Extraction

Information extraction is concerned with identifying predefined types of information from text [1]. Most systems incorporate some form of natural language processing, since the information to be extracted can only be found by recognizing semantic relationships defined by the role an entity performs in context. For example, to identify the name of a company in a body of text, it would be impossible to maintain a list of all known companies and simply perform a comparison, since this list would necessarily be incomplete and inordinately large. However, by searching for contextual clues, company names can be identified without a predefined list. In addition, context is needed in order to identify the role the company is playing in whatever action is taking place.

Current information extraction research seems to be heading in several directions, including partial parsing and automated knowledge acquisition [2]. Full syntactic parsing has been the goal of many endeavors in information extraction, but the benefits of building complete parse trees for every sentence in a document are outweighed by the accuracy and amount of processing required to perform this function. In many cases, partial parsing can approximate the accuracy and completeness of full syntactic parsing, at a fraction of the overhead.

Automated knowledge acquisition techniques acquire domain-specific knowledge automatically, allowing information extraction systems to better identify and extract information from texts in a particular domain. Currently, many natural language systems that use conceptual structures rely on domain knowledge to help extract information, and most of this knowledge

0-7803-4778-1/98 $10.00 © 1998 IEEE

engineering has been performed manually. Thus, automatic knowledge acquisition is essential to the development of systems capable of understanding text which is not limited to particular domains.

Retrieval

When using information retrieval engines on a large document collection, information is generally presented to users in the form of a list of documents which are most relevant (from the perspective of the retrieval technique being employed) to the given query (e.g., information returned from commercial Internet search engines). Such retrieval mechanisms serve a valuable function, but the technology has not kept pace with the growing amount of information that needs to be processed. In fact, most people complain that today's search engines present too much information in response to queries, and the relevance ranking of the information does not necessary correlate with the utility of the information. Users still have to wade through more information than they should in order to find what they are looking for.

Another method of answering user queries is to respond with a direct answer. In order for a system to perform this function, it must have a well organized knowledge base so it can draw inferences from the stored information [3]. Usually systems that respond to queries in this manner deal with very specialized domains. Putting these knowledge bases together is quite often a highly intensive manual process. As noted above, an automatic, domain independent approach to constructing knowledge bases is still an open problem in artificial intelligence research.

Another strategy for presenting information to the user is through the use of various browsing and visualization techniques. Significant research has been performed in the area of visualization, and many approaches have proved useful for particular types of data [4]. Cone trees provide a hierarchical view of information, with general concepts represented at the top of the tree and specific concepts represented at the branches [5]. Fish eye views bring relevant information to the users attention by highlighting or emphasizing that portion of the information set, and de-emphasizing the surrounding parts [6]. Other techniques present 3-dimensional views of documents and collections, where each dimension can represent a specific characteristic of the information set. These techniques can be combined with each other, or with other visualization methods to produce even more dramatic views of the information. However, in all cases, the data needs to be well-defined and structured. These visualization techniques generally work on only preconstructed repositories of data.

2. SYSTEM OVERVIEW

The Chronological information Extraction SyStem (CHESS) was developed to extract and store relational information from newspaper/newswire-type documents, and retrieve such information in two complementary ways [7]:

- broad search, or data grazing, enabled by a user-friendly, dynamic visual browser
- narrow search, via natural language queries for specific information

System Components

Browser: The CHESS Browser displays the contents of the CHESS database as a visually navigable semantic network. The relationships between named entities (people, organizations, places, etc.) are displayed and are also indexed to their appearances in specific documents so that interesting relationships can be traced back to the sources that assert them.

Question-Answering System: The CHESS Question-Answering System (QnA) analyzes natural language queries to determine the relational information needed to answer them. The response to a query is the collection of all sentences supplying the required information. Answer sentences can be traced back to the documents from where they came.

Document Processing System: The CHESS Document Processing System (CHESSDPS) handles the extraction of relational information from free text and its storage, for rapid searching, in a relational database. Subcomponents of CHESSDPS recast raw newsfeeds into a common SGML-tagged format, identify complete sentences, tag each sentence word with its part-of-speech, identify and interpret proper nouns and several kinds of phrases, extract relational data from the tagged text, and store this data using an ODBC-compliant database management system. The information is stored as Concept-Relation-Concept (CRC) triples (see Figure 1). The Document Server utility communicates with both the Browser and QnA for the purpose of retrieving the full text of relevant documents, and is considered a part of the CHESSDPS.

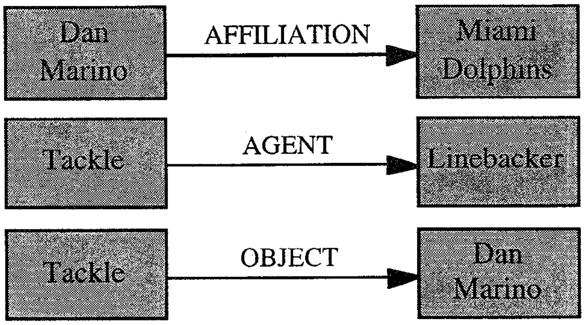


Figure 1: CRC Examples

Information Extraction

As noted above, the CHESSDPS performs the extraction of relational information from free text (using the Knowledge Extractor) as part of the CRC server. Subsystems perform SGML tagging of input text, including part-of speech identification and categorization of proper names. CRC triples are then extracted from the tagged data for subsequent storage in a relational database system. The following example illustrates how the Knowledge Extractor processes free text by semantically parsing the text and generating a Conceptual Graph knowledge representation.

The input to the Knowledge Extractor incorporates metadata information, which includes the headline, source and date of the document where the sample text resides.

INPUT:
U.S. Campaign on Sudan, Washington Post, 04/26/96
Egyptian President Hosni Mubarak was attacked by Islamic Militants in Addis Ababa.

Figure 2 shows the output of the semantic parsing. The first column shows the labels of the semantic relations which link concepts shown in the second and third columns. For example, the first line of the semantic parsing output means that the OBJECT of a concept "attack" is another concept, "Hosni Mubarak". The words following the concepts after the '|' delimiter are the semantic categories of proper name concepts (e.g., "Hosni Mubarak" is a person name).

```
OBJECT        (attack,                 Hosni Mubarak|person)
IS-A          (Hosni Mubarak|person,   President|title)
AFFILIATION   (Hosni Mubarak|person,   Egypt|country)
AGENT         (attack,                 Islamic|religion militant)
LOCATION      (attack,                 Addis Ababa|city)
```

Figure 2: Semantic Parsing Output

Figure 3 shows a graphical view of the Conceptual Graph generated from the semantic parsing output. Conceptual Graphs are the building units of the knowledge base being automatically constructed [8].

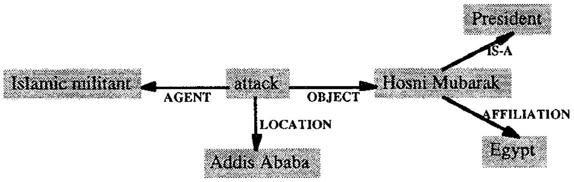


Figure 3: Conceptual Graph

Certain types of linguistic constructions commonly contain pertinent information about proper named entities located in close proximity. For example, apposition noun phrases can contain information directly relating a proper named entity to their role [9]:

Bill Clinton, President of the United States, ...

Detecting the end of an appositional phrase is a difficult task, since the noun phrase can be very complex, and it likely can be confused with other noun phrases having similar structure. Copula sentences also provide rich information for CRC extraction:

Bill Clinton is the President of the United States.

These types of clauses are easier to detect, since they have a well-defined structure, containing a "be" verb (*is, are, was, were*) and an optional determiner, and are less likely to be confused with other constructions. Precision and recall for this type is quite high. Other information-rich linguistic constructions have been identified, and algorithms for extracting CRCs from them have been developed. In addition, CRCs can be extracted from general text (not one of the identified constructions).

Queries

In general, user queries request information about particular entities, and their relationship to other entities. *"When did Disney acquire ABC?"*, *"What teams did the Miami Dolphins lose to last year?"*, and *"Which countries currently have economic sanctions against Iraq?"* are examples of such queries. The ability of CHESS to extract CRC triples from free formatted text data thus makes it an ideal tool for this type of transaction. CHESS can help to answer the following types of queries:

Scenario Analysis: CHESS analyzes and keeps track of the relatively abstract chronology of associations between entities that comprise a scenario of events. By putting together a scenario of events which lead up to a particular significant end event, it would be possible to predict when that end event may occur again at some point in time. CHESS can also take a general category of entity and produce specific instances of that category to help find answers to queries.

Trend Analysis: CHESS can show how the relationships between proper named entities changes over time. This allows users to spot significant trends in information, and to possibly predict a future course of action.

Biographical Questions: When the proper named entity is a person, CHESS can construct a detailed biography of the individual. It can incorporate everything known about the individual, or it can be tailored to only those activities of interest to the user.

Chronology: All information extracted by CHESS is time-stamped, so CHESS is particularly useful when answering questions about when an event took place, and when it occurred in relation to some other event(s).

Monitoring: Stored queries can be left to run in the background or at specified times, so that the user can be automatically notified when an event has happened concerning a particular named entity.

3. Browsing

CHESS uses several different techniques to visualize information. The main browser window is divided into several sections: the Hyperbolic Browser, the Concept Visualizer, and the Headline Viewer. Information is presented in the browser in the form of a cone of concepts and relations. The depth of the cone corresponds to the meaning relationships between concepts, and the breadth of the cone corresponds to the relationships extracted from the input text. Thus, the Hyperbolic Browser is used to drill down through the meaning relationships, and the Concept Visualizer is used to expand a concept into the relationships between it and other concepts.

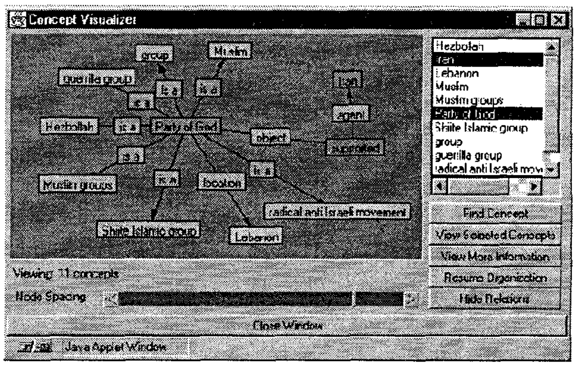


Figure 4: Concept Visualizer

The Concept Visualizer is shown in Figure 4, with information about a terrorist organization called *Party of God*. The relationship of other entities to *Party of God* can be discovered graphically, by visual browsing. For example, in Figure 4 we see that Iran is a backer of this organization. Boxes are colored differently to represent semantic relations, concepts, or more relevant concepts based on the current user view.

Figure 5 shows the Hyperbolic Browser, which is the ontology browser, allowing fast access to the knowledge base. The ontology used in this window is based on the TextWise proper name categorization scheme; i.e., all proper names which occur in the texts which are processed as a base to construct the knowledge base, are accessible by the hierarchical organization of the proper names. What is shown in Figure 5 are the names of the religions which occurred in the source texts.

The current ontology window is shown, since the concept, *Party of God*, was discovered by first exploring all religions which are related with Iran, and then by noting that *Party of God* is a Muslim group.

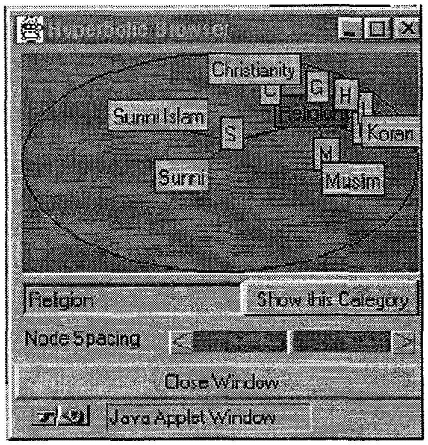
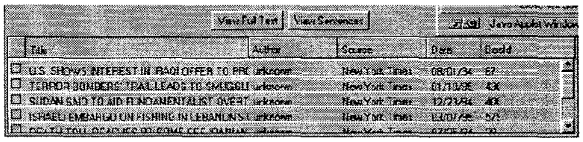


Figure 5: Hyperbolic Browser

Figure 6 shows the Headline Viewer, which displays the source texts' title, author, source, data, and ID. The graphical representation shown in the Concept Visualizer is based on the aggregation of data from multiple documents. In addition, users can view full texts of the source documents, or specific segments, (i.e., sentence, paragraph) if so desired.

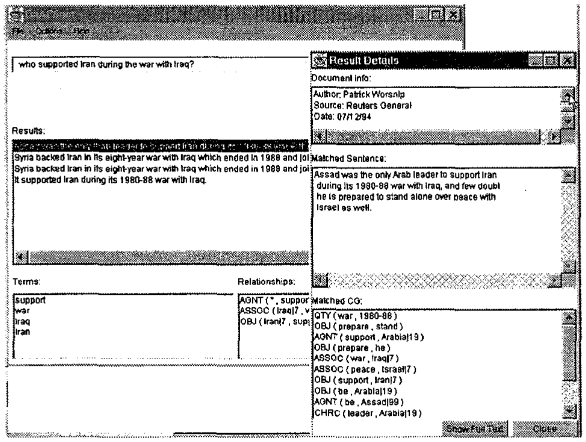


	Title	Author	Source	Date	DocId
□	U.S. SHOWS INTEREST IN IRAQI OFFER TO PRE	unknown	New York Times	08/01/94	67
□	TERROR BOMBERS' TRAIL LEADS TO SMUGGLI	unknown	New York Times	01/13/95	43
□	SUDAN SAID TO AID FUNDAMENTALIST OVERT	unknown	New York Times	12/23/94	40
□	ISRAELI EMBARGO ON FISHING IN LEBANON'S	unknown	New York Times	[illegible]	57
□	[illegible]	[illegible]	New York Times	[illegible]	[illegible]

Figure 6: Headline Viewer

4. Question-Answering

Figure 7 shows the main window in the question-answering interface to CHESS. A user enters either a yes/no question, or one which begins with "what", "who" or "which". The question should be as concise and clear as possible.



who supported Iran during the war with Iraq?

Results:

[illegible]
Syria backed Iran in its eight-year war with Iraq which ended in 1988 and joi
Syria backed Iran in its eight-year war with Iraq which ended in 1988 and joi
It supported Iran during its 1980-88 war with Iraq.

Terms:

support
war
iraq
iran

Relationships:

AGNT (* , suppor
ASSOC (Iraq|7 , v
OBJ (Iran|7 , sup

Document info:

Author: Patrick Worsnip
Source: Reuters General
Date: 07/12/94

Matched Sentence:

Assad was the only Arab leader to support Iran during its 1980-88 war with Iraq, and few doubt he is prepared to stand alone over peace with Israel as well.

Matched CG:

QTY (war , 1980-88)
OBJ (prepare , stand)
AGNT (support , Arabia|19)
OBJ (prepare , he)
ASSOC (war , Iraq|7)
ASSOC (peace , Israel|7)
OBJ (support , Iran|7)
OBJ (be , Arabia|19)
AGNT (be , Assad|99)
CHRC (leader , Arabia|19)

Figure 7: Question Answering Interface

The question answering system takes natural language questions input, and finds the answers by accessing the same knowledge base which was used in the knowledge base browser described in Section 3.

In this example (see Figure 7), the user has typed a question, *"Who supported Iran during the war with Iraq?"*, and the system found four answers to the question (shown in the "Results" box). Notice that the answers are displayed as sentences which contain the actual answer. The box labeled "Terms" shows concepts which occurred in the question. These are determined automatically by the system. The "Relationships" box shows the semantic parsing output of the question.

The system displays metadata information such as author, source, and date of publication in the "Result Detail" window. Under this metadata, the system shows a result sentence, followed by its semantic parsing output.

5. Conclusion

CHESS technology has achieved great success in advancing the state of the art in information extraction and retrieval. It automatically extracts concepts and relationships between proper named entities from free text and can then populate a relational database with this information. It can also be used as a precision-enhancing adjunct in document retrieval systems, it can supply automatically extracted content to information management and visualization systems, and it can serve as the basis for semantic content analysis and comparison systems.

6. References

[1] MUC-3, *Proceedings of the Third Message Understanding Conference (MUC-3),* Morgan Kaufman, San Mateo, CA, 1991.

[2] Riloff, Ellen, *Information Extraction as a Stepping Stone toward Story Understanding*, in "Understanding Language Understanding: Computational Models of Reading, edited by Ram and Moorman, MIT Press, 1997.

[3] Paik, Woojin, Liddy, J., Allen, E., Liu, D., *Browsing, Question-Answering, and Information Retrieval in CHESS*, in "Proceedings of the IEEE Dual-Use Technologies and Applications Conference, Syracuse, NY, 1997.

[4] Goan, Terrance, *Artificial Intelligence Support for Graphics-Based Information Access*, Air Force Research Laboratory Final Technical Report, AFRL-IF-RS-TR-1998-73, 1998.

[5] Robertson, George, Mackinlay, J., Card, S., *Cone Trees: Animated 3D Visualizations of Hierarchical Information*, in "Proceedings of UIST", Hilton Head, SC, 1991.

[6] Furnas, George, Zacks, J., *Multitrees: Enriching and Reusing Hierarchical Structure*, in "Proceedings of Human Factors and Computer Systems", Boston, MA, 1994.

[7] TextWise, LLC, *CHESS Software User Manual (SUM)*, for AFRL, 1998.

[8] Sowa, John, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, MA, 1984.

[9] Liddy, Elizabeth, Paik, W., *Information Extraction from Text Through Application of Advanced Processing Techniques*, Air Force Research Laboratory Final Technical Report, RL-TR-96-46, 1996.

7. Acknowledgments

CHESS research and development has been conducted by TextWise, LLC, and funded in part through a Small Business Innovative Research (SBIR) contract, F30602-96-C-0164, with the Air Force Research Laboratory's Information Directorate.