INFORMATION EXTRACTION OVERVIEW

Mary Ellen Okurowski

Department of Defense.
9800 Savage Road,
Fort Meade, Md. 20755
meokuro@afterlife.ncsc.mil

1. DEFINITION OF INFORMATION EXTRACTION

The information explosion of the last decade has placed increasing demands on processing and analyzing large volumes of on-line data. In response, the Advanced Research Projects Agency (ARPA) has been supporting research to develop a new technology called information extraction. Information extraction is a type of document processing which captures and outputs factual information contained within a document. Similar to an information retrieval (IR) system, an information extraction system responds to a user's information need. Whereas an IR system identifies a subset of *documents* in a large text database or in a library scenario a subset of resources in a library, an information extraction system identifies a subset of *information* within a document. This subset of information is not necessarily a summary or gist of the contents of the document. Rather it corresponds to pre-defined generic types of information of interest and represents specific instances found in the text. For example, a user of a system may be interested in identifying and databasing information on all companies named within a set of documents, including companies not previously known to the user. An information extraction system can extract and output all of the occurrences of company names within a text with an accuracy of 75%. Moreover, it is possible to specify that the system only extract those companies of a certain type, such as Japanese companies or companies in the textile industry.

Information extraction is also related to but distinct from another type of document processing, machine translation. Both technologies process texts in multiple languages. A machine translation system converts the entire text of a source document into a different target language; an information extraction system identifies and extracts relevant information within a document in a particular language. There is no conversion from one language to another. An information extraction system in Japanese outputs information in Japanese.

Under ARPA sponsorship, research and development on information extraction systems has been oriented toward evaluation of systems engaged in a series of specific application tasks [7][8]. The task has been template filling for populating a database. For each task, a *domain* of interest (i.e., topic, for example joint ventures or microelectronics chip fabrication) is selected. Then this domain scope is narrowed by delineating the particular types of factual information of interest, specifically, the generic type of information to be extracted and the form of the output of that information. This information need definition is called the *template*. The design of a template, which corresponds to the design of the database to be populated, resembles any form that you may fill out. For example, certain fields in the template require normalizing information format (e.g. dates) and others require selecting from a set list of choices (e.g hair color). The template definition is supplemented by a set of template *fill rules* which document the conditions for extracting information and formally serve as the extraction guidelines. The fill rules may evolve as more and more data is examined and the analysts gain more understanding and control of the intricacies of an application.

To date, information extraction has been performed almost entirely manually. Even with careful template or database design and explicit fill rules, manual information extraction is not at all error-free. In carefully controlled experiments, Will found that analysts had an error rate of about 30% even after substantial training and several months of practice [10]. There was also little improvement after the initial few months. Some of these errors resulted from lapses of attention caused by the tedium of performing a repetitive task. A substantial cause of this unexpected lack of consistency lies in the cognitive demands that information extraction places on analysts. A brief example demonstrates this point. Table 1 below identifies five types of information to be extracted from one of the TIPSTER texts and correlates each type with a cognitive skill an analyst must

apply in extracting that information.Within a document, an analyst first locates a candidate entity by identifying and distinguishing a single entity from other entities, generally on the basis of the entity name. This candidate entity must be kept distinct from other entities, but, in addition, other references to the same entity using variations of the name or aliases must be merged so that there is only a single entity in the extraction template. Characteristics of the entity can also be assigned; the analyst can characterize the entity by nationality or classify the entity by type as one of a set of choices. In an application where relationships among entities are important, the analyst may need to link one entity to another. All of these activities make for a complex set of cognitive demands placed upon the analyst that often require subtle judgements to be made.

Type of Information	Cognitive Skill	Example
Name	Identify	*Toyobo Co.*
Alias	Merge	*Toyobo*
Nationality	Characterize	*Japanese*
Type	Classify	*company*
Entity-Relationship	Link	*Toyobo Co., Kanematsu Corp.*

Table 1: Cognitive Skills in Manual Extraction

1.1. Information Extraction System Architecture

How then does an information extraction system perform the kinds of complex processes required to identify and extract information? In general terms, an information extraction system is composed of a series of modules (or components) that process text by applying rules [2]. Since information extraction involves selected pieces of data, an extraction system processes a text by creating computer data structures for relevant sections of a text while at the same time eliminating irrelevant sections from the processing. Although there will be variations among systems, generally the functions for the following set of modules will be performed somewhere in the processing.

The initial module, a Text Zoner, takes a text as input and separates the text into identifiable segments. The Preprocessor module then takes the segments that contain text (as opposed to formatted information) and, for individual words within each sentence in those segments, accesses a lexicon (i.e.,dictionary) and associates properties like part-of-speech and meaning with each word. To reduce the amount of information to be processed, the Filter module subsequently eliminates sentences that do not contain any relevant information for the application.

The following modules, including the optional Preparser and Fragment Combiner modules, are geared toward analyzing the grammatical relationships among the words to create data structures from which sentence meaning can be interpreted. Because of the difficulty of analyzing these relationships, more and more of the systems have developed a Preparser module here to identify sequences or combinations of words that form phrases. Accessing grammar rules, the next module, the Parser, analyzes the sequences of words and phrases and tries to understand the grammatical relationships among the constituents. The output is either a successfully analyzed (parsed) sentence with relationships among the sentence constituents labelled or a partially analyzed sentence with some constituent relationships labelled and others constituents left as unattached fragments. It is these unattached fragments that bring the Fragment Combiner module into play to try to turn a partially labelled sentence with fragments into a completely labelled one.

With the grammatical relationships identified, either a fully analyzed sentence or a partially analyzed sentence containing fragments is then processed by the Semantic Interpreter. This module interprets the labelled grammatical relationships and generates a representation of the sentence meaning in some form. The next module, the Lexical Disambiguation module, replaces the representation of any ambiguous words within the sentence with a specific, unambiguous representation.

The next step, the Coreference Resolution module, takes the meaning representation for the sentences within a text (from the Semantic Interpreter) and identifies which of the events or entities that occur in the data structures of the individual sentences actually refer to the same entity or event in the real world, a critical step to avoid database duplication. The final module is the Template Generator in which information output by the Semantic Interpreter and Coreference Resolution modules is turned into template fills in the desired format.

1.2. Other Tasks for Information Extraction Systems

An information extraction system may also be configured to perform tasks other than template filling. Such a configuration may involve use of some of the modules in a full system, use of modules in a different sequence than described above, or modification of the modules themselves.

Examples of such tasks include text tagging and indications and warnings. In a text tagging task, information of a particular generic type is identified, such as persons, companies, or dates. These types generally occur in a wide range of domains. The information is identified in the Preparsing module and, for some types, must be processed by the Coreference Resolution module to eliminate multiple references. The output of such a system might be used, for example, to create indexes to documents for later information retrieval applications. Another example might be the display of the original text directly to an analyst, with relevant types of information marked or highlighted in some way.

In an indications and warning task, an analyst is attempting to identify information providing evidence that a particular event or events have occurred. The system is likely to be configured the same as for a template filling task, but with the Template Generator Module modified. Such modification can allow indications and warning data to be output in whatever form is convenient for the analyst because he is using the system in some ways as a sophisticated information retrieval mechanism. He wants to identify specific events of some generic type, but does not want to database and track the information. The data extraction system is alerting the analyst to the presence of certain types of data.

2. TIPSTER TEXT PROGRAM GOALS

As information extraction technology has matured, the design of systems has responded to requirements for higher accuracy, faster performance, broader coverage, extensibility within a domain, and portability to new domains. The Phase One extraction part of the TIPSTER Text Program focused on further advancing information extraction technology by setting two goals: (1) to develop information extraction systems that include innovative algorithms that improve overall system performance and (2) to demonstrate, through a task-oriented testbed application, the portability of these systems to new languages and domains.

Four contractor teams with different algorithmic approaches were selected: Bolt Beranek & Newman, GE Corporate Research and Development/Carnegie Mellon University/ Martin Marietta Management and Data Systems, New Mexico State University/Brandeis University, and the University of Massachusetts/Hughes. A template-filling task was defined for two domains and two languages [5].The University of Massachusetts/Hughes was tasked to work in both domains, and the other three teams were tasked to work in both domains and languages. As is the case in any large scale research program in which a number of sites share the same development data set and participate in regularly scheduled evaluations, general trends in innovative algorithm development appeared across systems. These included statistical language modeling, the automated acquisition of knowledge, generic tools and taggers, and the use of finite-state pattern matching.

These trends reflect the interrelatedness of the goals of the TIPSTER program, improvement of the technology within the context of a task. The first goal, that of improving the technology, meant overcoming two central problems with which pre-TIPSTER information extraction systems had been grappling.These were incomplete knowledge for text processing and inadequate text processing algorithms. Without adequate linguistic and domain knowledge, an information extraction system is brittle, and performance is poor. Without analysis of text processing algorithms and experimentation with innovative extraction algorithms, there can be no serious re-engineering of the technology and subsequent breakthroughs. Under TIPSTER, system developers have increasingly adapted practical approaches in algorithm development, not simply by coping with deficient information, but by overcoming the deficiency by creatively redefining (1) which knowledge is acquired, (2) how that knowledge is acquired, and (3) how that knowledge is applied. They have continued the direction of redefining information extraction algorithms through redesign of processing modules and their functions. The second goal, that of language and domain portability within the context of a task, required understanding the task application itself and the direct effect of this understanding upon the technology development. The above-mentioned trends will be discussed in terms of these two goals.

First, with the employment of statistical language modeling, we see an extension in the definition of knowledge, its acquisition, and its application. Knowledge sources are defined to include not just linguistic and domain knowledge, but also statistical models of language as well. This knowledge is acquired through training on archived texts or templates and applied to guide processing on a new task, shoring up deficient linguistic and domain knowledge. For example, the BBN PLUM system uses Markov modeling techniques in its part-of-speech tagger, POST [6]. In the preprocessing stage, POST assigns part-of-speech tags to known words using probabilities derived from large corpora and probabilities for unknown or highly ambiguous words based on word endings. Later in template filling, BBN applies a correctness probability model in order to estimate the confidence for any given PLUM response.The model can also be used to filter out hypothesized answers that fall below a given threshold or rank and select among possible (slot) fillers. That these types of knowledge were derived automatically from annotated text or templates is an example of a shift in how knowledge can be acquired for information

extraction systems and a demonstration of the greater ease in porting to new domains and languages.

The second trend, just touched upon in the discussion of statistical language modeling, is the automated acquisition of knowledge.This includes both algorithms for automatic acquisition of knowledge from corpora or templates and automated tools for acquiring knowledge from human experts. The University of Massachusetts/Hughes system, CIRCUS, for example, makes use of both forms of automated knowledge acquisition [4]. The system development team focused their effort on automating the construction of domain-specific dictionaries and other language resources and thereby minimizing human assistance in customizing CIRCUS for specific applications. The dictionary construction tool, AutoSlog, is a good illustration of this approach [9].Within CIRCUS, concept nodes indicate potential extractable concepts. AutoSlog proposes domain-specific concept node definitions to a human who selects good definitions, rejects bad ones, and thus creates the dictionary component of the CIRCUS system. An initial experiment with two analysts, who had filled templates in the English joint venture domain, demonstrated that analysts can build effective dictionaries quickly for information extraction with the assistance of Autoslog. Learning algorithms, like those developed in PLUM and CIRCUS, signal new directions in automating the acquisition of knowledge.

The two trends in new algorithms for information extraction processing reflect an evolving understanding of the information extraction task, in particular, the TIPSTER task with two different domains and languages. Information extraction can be viewed as an odd tug-of-war between two opposing demands. On the one hand, there are demands for generic systems created with general knowledge sources that are broadly reusable across domains. On the other hand, there is the reality of domain-specific requirements, tied ultimately to the domain text. Reusability makes information extraction more feasible, tailoring makes information extraction more successful

The need for generality has resulted in the third trend: reusable tools and taggers. From the initial suite of template filling tools (that allow analysts performing manual extraction to easily organize information to fill in fields and that can detect errors for analysts) to Tabula Rasa, New Mexico has driven tool design toward general, reusable tools for any domain in any language [1]. Tabula Rasa allows a developer to create a complete template-filling tool for a new topic domain in an afternoon, once a template definition is available. In a similar way, annotated text taggers represent a new understanding of the extraction process, perhaps affected to a large extent by the movement toward object-oriented design. Taggers are based on the assumption that there are types of information, data elements, that occur across domains. The New Mexico State University/Brandeis system DIDEROT uses finite-state feature taggers to identify things such as names, organization names, place names, and data expressions.Taggers (also known as specialists, recognizers, or concept taggers) identify information early on in the processing and label it. This facilitates processing in later stages by marking larger units for processing. This ability of taggers to be reusable gives systems a headstart in a new application.

The conflicting reality that a new application requires specific domain knowledge has resulted in the fourth trend: finite-state pattern matching. Information extraction is viewed as a domain application, where content, specifically the corpus, rather than linguistic knowledge sources, drives system development. A number of different groups have adopted this approach, including the TIPSTER site GE/ CMU/MM. Their SHOGUN system design illustrates the central role of pattern matching [3].With domain/application patterns central, the team essentially redefined the knowledge to be acquired as "domain knowledge" and acquired that type of knowledge by analysis of the domain corpus. To apply these patterns, they replaced the traditional Parsing module with a Pattern Matching module. The SHOGUN team's complementary focus on corpus knowledge acquisition and the relative ease of implementing finite-state rules creates a reusable, simple approach for new domains and languages.

3. OVERVIEW OF THE INFORMATION EXTRACTION SECTION

The information extraction section of this volume is a collection of papers that provide a broad perspective of information extraction within the TIPSTER Text Program, Phase one. This overview paper is followed by a paper discussing details of the extraction tasks "Tasks, Domains, and Languages for Information Extraction", a paper describing the selection of text corpora and preparation of filled templates for the task, "Corpora and Data Preparation for Information Extraction", and a paper examining template design issues, "Template Design for Information Extraction." The next three papers help the reader interpret evaluation results. In "TIPSTER/MUC-5 Information Extraction System Evaluation", the design and overall results of the final evaluation of the TIPSTER Phase one extraction systems are discussed. The unexpected higher system performance in Japanese is examined in the next paper, "An Analysis of the Joint Venture Japanese Text Prototype and Its Effect on System Performance." The performance of human analysts for the extraction task is

compared with that for machine systems in the third paper, "Comparing Human and Machine Performance for Natural Language Information Extraction". Papers from each of the TIPSTER teams then provide a technical description of their research and system development efforts: "BBN's PLUM Probabilistic Language Understanding System", "The TIPSTER/SHOGUN Project", "CRL/Brandeis: The DIDEROT System", and "UMass/Hughes: Description of the CIRCUS System Used for TIPSTER Text" ("Dictionary Construction by Domain Experts").

REFERENCES

1. Cowie, Jim et al., "CRL/Brandeis: The DIDEROT System", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca.,1994.

2. Hobbs, Jerry, "The Generic Information Extraction System.", In *Proceedings of the Fifth Message Understanding Conference, (MUC-5)*. Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

3. Jacobs, Paul S., et al.,"The TIPSTER/SHOGUN Project", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

4. Lehnert, Wendy, et al. "UMass/Hughes: Description of the CIRCUS System Used for TIPSTER Text", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

5. Onyshkevych, Boyan, et al., "TIPSTER Tasks, Domains and Languages for Information Extraction", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

6. The Plum System Group, "BBN's PLUM Probabilistic Language Understanding System", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

7. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, Morgan Kaufmann Publishers, San Mateo, Ca., 1992.

8. *Proceedings of the Fifth Message Understanding Conference (MUC-5)*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

9. Riloff, Ellen and Wendy G. Lehnert, "Dictionary Construction by Domain Experts", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.

10. Will, Craig, "Comparing Human and Machine Performance", In *Proceedings of the TIPSTER Text Program, Phase One*, Morgan Kaufmann Publishers, San Mateo, Ca., 1994.