Knowledge Extraction using Information Extraction

Enrique Alfonseca

Dpto de Ingeniería Informática
Universidad Autónoma de Madrid
28049 Madrid
enrique.alfonseca@ii.uam.es

Computer Science Department
University of York
York YO10 5DD
enrique@cs.york.ac.uk

1 Introduction

Information Extraction is a somewhat recent discipline, and it consists of techniques and methods for extracting structured information from free text, i.e. from text as it is written or spoken by us. The scientific community began studying this problem in detail about ten years ago. At least two facts have triggered this research [Zechner, 1997]. In the first place, the amount of text data has been increasing exponentially in the last years: clinical records in hospitals, newswire articles, scientific papers... The need to classify and extract the relevant information from these large repositories of documents has given a big impulse to research in Information Extraction. Secondly, this field has been pushed forward by the ARPA funded Message Understanding Conferences (MUC). Nowadays there are some fully automated IE systems working with state-of-the-art technology, and with a performance and accuracy similar to that of a human expert.

2 The history of Information Extraction

A precursor of IE is the field of text understanding. In this approach, the goal is to produce an accurate representation of the contents of an entire text. All the sentences are evaluated in depth, and the output should be a complete world model, a pragmatic representation of the text. As expected, these systems required a great deal of hand-crafted engineering, and they only operated in reduced domains. Two of them are the PUNDIT [Hirshman et al, 1989] and TOPC [Hahn, 1989]. The system PAKTUS [Thompson, 1994] apparently used *full natural language understanding* for Information Extraction. It participated in the earlier MUCs, up to MUC-4.

The information extraction systems, in the way that we build them nowadays, appeared and evolved impulsed by the TIPSTER program and the Message Understanding Conferences. In fact, the first two MUCs were held before the creation of the TIPSTER program. Later, the conferences were organised by the TIPSTER committee, focusing their efforts on *information extraction*, as opposed to the TRECs conferences which focus on *Document Retrieval*.

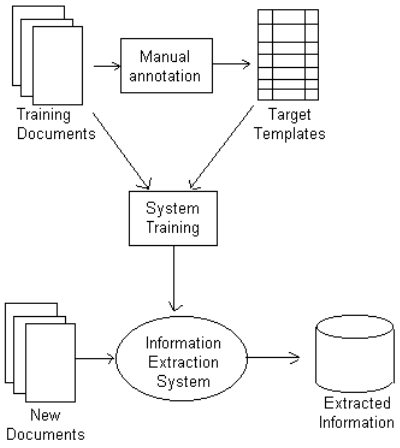

Figure 1: Procedure to build an Information Extraction System.

3 Information Extraction

Information Extraction [Hobbs, 1998] is the set of techniques and methods used to obtain structured data from natural language texts.

The user has a set of texts belonging to a domain of interest (newswire articles, medical reports, etc.), and he specifies which particular information has to be extracted. For example, if the user has a repository of medical documents, he might be interested in knowing, for each document, the patient's name, the doctor's name, the illness, the date and the prescription given. For each text in the database, the system has to decide whether it is relevant or not, and extract this information.

A typical information extraction system begins with a training phase, during which it learns the relevant words for the domain of interest and the rules for finding the information. Nowadays, this training has to be supervised by software developers and domain experts, but a big effort is being devoted to automatizing it as much as possible. This process is illustrated in Figure 1.

When the training is complete, the system is ready to work on new documents without special supervision.

4 Structure of an information extraction system

An information extraction system can be described as a cascade of different modules which receive a text and delete irrelevant information while annotating the relevant data.

Usually a complete system has the following modules, not necessarily in the same order:

1. *Text Zoner*: turns a text into a sequence of text segments. for example, it can separate the formatted from the unformatted regions. In general, a text zoner is coded *ad hoc*, without a systematic treatment.

2. *Preprocessor*: identifies the sentences in the text and the words in the sentences, and annotates each word with its part-of-speech. It identifies certain types of words, such as company names, dates, places, personal names, currencies, and so forth. It also handles unknown words, trying to guess as much as it can about them, and corrects the spelling errors.

3. *Filter*: deletes the irrelevant sentences. Some systems discard the sentences that do not contain specific triggering words or patterns, while others use statistical methods.

4. *Preparser*: tries to identify some small-scale structures, like simple noun phrases, verbs, objects, etc.

5. *Parser*: returns the parse trees for the different sentences, or possibly only fragments of them. Bottom-up chart parsers are the most widely used.

6. *Fragment Combiner*: combines fragments of parse trees into a unique parse tree for the whole sentence. So far all the methods used are *ad hoc*.

7. *Semantic Interpreter*: generates a semantic structure or logical form. To my knowledge, no current system attempts to achieve a full understanding of the text. Instead, the semantic structure produced is fragmentary, with only the information that is relevant to the user.

8. *Lexical Disambiguation*: solves ambiguities. It usually uses statistical methods taking into account the context of the ambiguous word.

9. *Coreference Resolution* or *Discourse Processing*: identifies elements from different sentences which refer to the same object, and finds the relationships among the objects referenced in the text.

10. *Template Generator*: derives the templates from the semantic structure. It usually inserts them into a relational database.

5 The Yorkie system

In the first stage of out project, I plan to develop the Yorkie (York Information Extraction) system together with José Luis Jara, under the supervision of Professor Suresh Manandhar. It will be a modular system with the following parts:

- Tokenizer
- Sentence splitter
- Part-of-speech tagger
- Morphological analyser

- Named-entity recogniser
- Parser
- Semantic analyser
- Discourse analysis

Each module will convey the generated information to the following one through XML annotations in the texts. We have already designed the XML specifications (the DTD) for our documents. Finally, the output of the system will be written according to the specifications in the Seventh Message Understanding Conference.

6 Further research

Once the system Yorkie is operative and working, we intend to use it as a starting point for new research on Information Extraction techniques, particularly in:

1. Using machine-learning techniques for training the modules and increasing the portability.
2. Using multilanguage techniques in some of the modules.
3. Possibly applying the system to the medical domain.
4. Exploring ways of representing the output with knowledge representation methods (semantic networks, frame systems, etc.)

References

[Hahn, 1989] U. Hahn, 1989. *Making understanders out of parsers: semantically driven parsing as a key concept for realistic text understanding.* International Journal of Intelligent Systems, 4.

[Hirshman et al, 1989] Lynette Hirshman, Martha Plmer, John Dowding, Deborah Dahl, Marcia Linebarger, Rebecca Passonneau, Francois-Michel Lang, Catherine Ball and Carl Weir, 1989. *The PUNDIT natural-language processing system.* Proceedings of the Annual AI Systems in Government Conference, Washington DC.

[Hobbs, 1998] Jerry R. Hobbs, 1998. *Generic Information Extraction System.* Artificial Intelligence Center SRI International.
http://www.nist.gov/itl/div894/894.02/related_projects/tipster/gen_ie.htm

[Thompson, 1994] P. Thompson, 1994. *Concept-based information extraction.* Integrated Computer-Aided Engineering, 1

[Zechner, 1997] Klaus Zechner, 1997. *A Literature Survey on Information Extraction and Text Summarization* Paper for Directed Reading.