# Natural language processing and information extraction: qualitative analysis of financial news articles.

Marco Costantino, Richard G. Morgan, Russell J. Collingham, Roberto Garigliano
Department of Computer Science, University of Durham
Durham, DH1 3LE, U.K.
Tel +44 191 374 2631, Fax +44 191 374 2560
marco.costantino@durham.ac.uk

January 31, 1997

**Abstract**

Quantitative financial data are today largely analyzed by automatic computer programs based on traditional or artificial intelligence techniques. Differently, qualitative data and, in particular, articles from on-line news agencies or from financial newspapers are not yet successfully processed. As a result, financial operators suffer from qualitative data-overload. This paper addresses the issue of the use of Natural Language Processing and, in particular, information extraction, for processing qualitative financial data. The financial information extraction system under development at the University of Durham can identify specific kinds of information within a source article, producing a set of relevant templates which represent the most important information in the article and therefore reducing the operators' qualitative data-overload. The application has been designed in close contact with experts of the financial sector and can be fully customized by the user who can add new templates to the existing ones.

## 1 Introduction

The goal of *information extraction*, which belongs to the field of Natural Language Processing, is to extract specific kinds of information from a source article [Riloff and Lehnert, 1994]. In other words, the input to the system is a collection of documents, (e.g. a newspaper articles), while the output is a representation of the relevant information from the source documents, according to specific extraction criteria. Information extraction can be used to provide the financial operator with the most important information from a collection of source articles, depending on the relevance of the information on the operator's investment decision making process. For example, the following article from *The Financial Times* represents information which is relevant for the financial operator's decision making-process, since it is likely to have some influence on the share price of the companies involved in the deal:

*BELL ATLANTIC announced it will acquire Tele-Communications Inc with 18 billion dollars. The deal will radically change the US communications industry. It will also be one of the country's biggest ever takeovers. The deal - which is bound to face strong regulatory scrutiny - would be the first full merger between a US telephone company and a cable business at a time when the two industries are converging to create a single, multi-media inter-active entertainment and information business. Bell Atlantic, the telecommunications group serving the middle part of America's eastern seaboard, has been one of the most aggressive telephone companies trying to enter the video industry.*

Such articles tend to include information which is not essential for understanding the information which needs to be extracted. Using a financial information extraction system, instead, a "*template*" (summary) of the original article can be automatically created, which contains the most important information in the article and skips the rest of the information. A possible template for the article shown above could be:

```
Template: TAKEOVER
    COMPANY_TARGET: Tele-Communications Inc.
    COMPANY_PREDATOR: BELL ATLANTIC
    TYPE_TAKEOVER: FRIENDLY
    VALUE: 18 billion dollars
    ATTRIBUTION: BELL ATLANTIC
```

The *template* allows the operator to quickly understand the information in the article without having to read the whole article. This can significantly reduce the qualitative data-overload from which financial operators suffer and allow them to take the appropriate investment decision. The representation of the information in templates has the advantages of being highly structured. This allows further processing of the information by other applications such as traditional databases packages or other financial applications such as expert systems or neural networks [Costantino *et al.*, 1996b].

Information extraction is different from information retrieval. Information retrieval engines are able to locate the relevant documents within a collection, but they are unable to extract information from the relevant documents according to specific criteria. The power of an information extraction system compared to a information retrieval system is therefore in the ability of *extracting* the relevant information in the articles according to specific extraction criteria and represent them in structures (templates), which information retrieval systems are unable to produce.

The key-element of an information extraction system is the definition of the information to be extracted from the source articles. The financial information extraction system under development at the University of Durham allows the extraction of a set of predefined templates. In addition, the user can define new templates using the natural language user-definable template interface.

## 2 The financial activities approach

The financial information extraction system under development at the university is based on a set of pre-defined templates which has been defined according to the *financial activities* approach. A financial activity is defined as an event which is likely to have a direct influence on the share-price and, therefore, on the investment decision-process of the operator. Three groups of different financial activities have been identified, and a *template* is associated to each of the activities.

- The first group of activities comprises **company related events**, for example takeover, merger, market movement, dividend/profit announcement, privatisation, new issue, director's dealings, investigation, etc. These events are likely to have a direct impact on the share price of the company.

- The second group of activities comprises **company restructuring events**, for example, new product, new line of products, joint venture, staff changes, new factory and, in general, events related to changes in the productive structure of the company.

- The third group of activities comprises **general macro-economic events**, which are likely to have a direct impact on the share price of most of the shares on the market, for example, interest rates changes, unemployment, inflation changes, currency movements etc.

A template, which represents the essential information for the event, is associated to each of the activities. For example, an article about a takeover will be summarized using the takeover template shown in section 1.

The system can be used for processing articles from newspapers (e.g. *The Financial Times* or *The Wall Street Journal*) or from real-time news providers (e.g. *Bloomberg* or *Dow Jones*).

## 3 User-defined templates

The system allows the financial operator to add additional templates which are not directly available in the pre-defined templates collection. Adding new templates allows the system to capture additional information in other kinds of articles which the user might be interested in knowing, such as specific kind of companies, markets, etc. This allows the maximum degree of flexibility for the user.

The user-definable interface allows the user to add new templates using sentences in natural language text and specific *formal-elements*, designed to reduce the amount of possible ambiguities in the templates definitions but does not reduce the user's expressive power. The definition of these *formal-elements* has been done analyzing the results of an experiment carried out by potential users of the system. The test required the potential users to describe a generic takeover template using sentences in natural language. More specifically, the users were asked to describe the *condition* under which the template should have been filled (the template *main-condition*) and the specific slot-rules. The target of the experiment was to identify two key-points:

- how easy is for the user to define the templates using unconstrained input natural language text;
- how easy would it be for the system to understand such unrestricted input definition.

The analysis of the results suggested that allowing complete freedom to the user can lead to a difficult situation for both the user and the system:

1. the user can find it difficult to express the template definitions using unrestricted natural language text without the support of any formal element. This is because a template (for example the takeover template shown in figure 1) is rather structured and implies a relevant number of coreferences between the various elements. Differently, free natural language is rather unrestricted and can therefore be difficult to define a structured object such as a template without any other support. For example, the definition of the takeover template using free natural language text could be as follows:

   **Template Condition:** A company acquired another company

   **Slot: Predator:** name of the company acquiring

   **Slot: Target:** name of the company which is being taken over

   As it is possible to notice, the definition of the template is rather fuzzy and imprecise, and the user is forced to repeat information in the definition of the different slots;

2. the unrestricted natural language input can be rather difficult to process for the system and a relevant number of ambiguities can be found in the template definitions. For example, in the definition above the system would have had to identify the relations between the *companies* cited in the *template condition* and those stated in the *slots* which can be rather difficult.

Following the results of the experiment, three different kinds of variables, *formal elements*, have been introduced in the user-definable template interface. The formal elements have been designed to reduce the amount of possible ambiguities in the templates definitions but do not reduce the user's expression power and are:

```
Template-name:         T=TAKEOVER
Variables:             V=COMPANY1 is an organization
                       V=COMPANY2 is an organization.
                       V=VALUE is money.
Template main-event:   V=COMPANY1 acquired V=COMPANY2.
                       V=COMPANY1 acquired V=COMPANY2 with V=VALUE.
                       The acquisition of V=COMPANY2 by V=COMPANY1.
                       The V=VALUE acquisition of V=COMPANY2 by V=COMPANY1.
                       V=COMPANY1 paid V=VALUE for V=COMPANY2.
                       V=COMPANY1 acquired a majority stake in V=COMPANY2.
                       V=COMPANY1 took full control of V=COMPANY2.

Definition of slots:

S=COMPANY-PREDATOR:    V=COMPANY1
S=COMPANY-TARGET:      V=COMPANY2
S=TYPE-OF-TAKEOVER:
  String-fill: HOSTILE   T=TAKEOVER is hostile.
  String-fill: FRIENDLY  T=TAKEOVER is not hostile.
S=VALUE-OF-TAKEOVER:   The cost of T=TAKEOVER
                       V=VALUE
S=BANK-ADVISER-PRED:   The adviser of V=COMPANY1.
S=BANK-ADVISER-TARG:   The adviser of V=COMPANY2.
S=EXPIRY-DATE:         The date of expiry of T=TAKEOVER.
S=ATTRIBUTION:         The person who announced T=TAKEOVER
                       The organization who announced T=TAKEOVER
```

Figure 1: The takeover template defined using the user-definable template interface.

- the **name of the template**, which must begin with the letters “T=”. Tee name of the template can be used in the definitions of the slots to refer to an event which is represented by the template as a whole. For example, the slot *S=VALUE-TAKEOVER* in the takeover template shown in figure 1 is defined as “*the cost of the T=TAKEOVER*;

- the **variables** which can be defined by the user (beginning with the letters “V=”) to identify the elements of the main-events which will be later used in the definition of the slot-rules. For example, in the definition of the takeover template shown in figure 1, the user can define the variable “*V=COMPANY1 is a company*” which is used to identify the company predator and, therefore, appears in both the *main-event* (“*V=COMPANY1 acquired V=COMPANY2*”) and the slot-definitions (“*S=COMPANY-PREDATOR: V=COMPANY1*”);

- the **slot-names** which can be used in the definition of other slot rules to refer to the information contained in the previous slots. For example, the slot *S=ATTRIBUTION* of the takeover template shown in figure 1 refers to the other slots.

In figure 1 the takeover template shown in section 1 has been defined using the user-definable template interface.

## 4 Architecture of the system

The system is based on the natural language processing system under development at the university of Durham. The system has been under development for the past nine years and is based on deep natural language understanding. It has been under development for the past nine years and has recently successfully participated in the MUC-6 competition, the most important

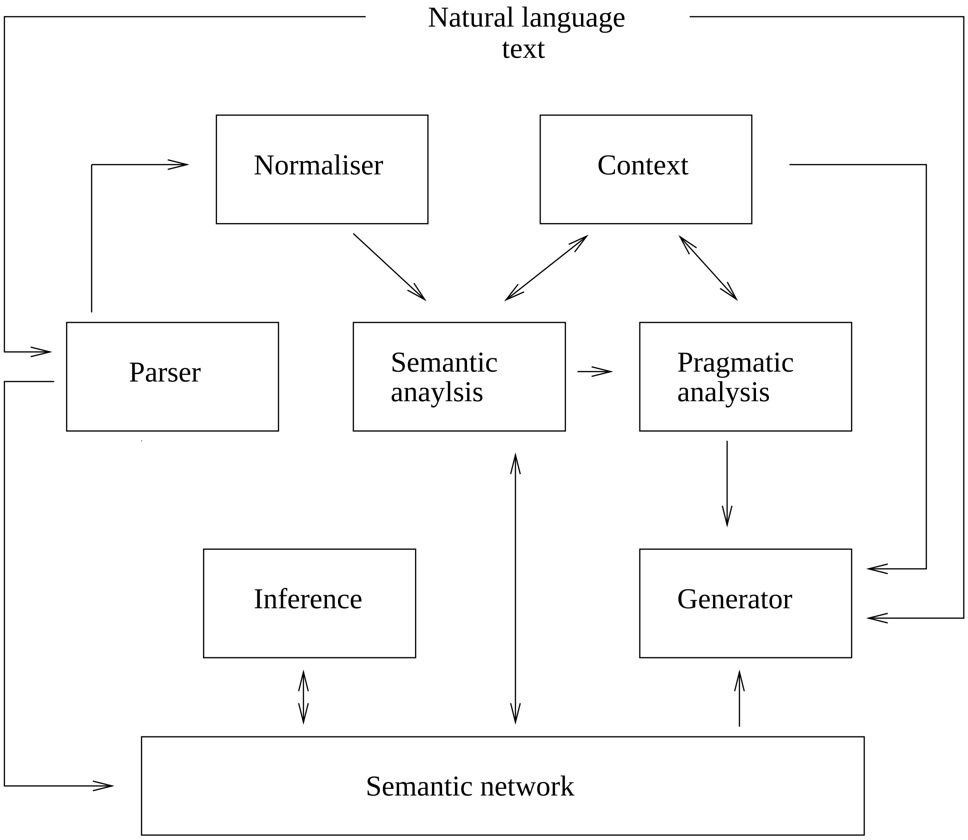

Figure 2: The Durham Natural Language Processing System's core.

competition for information extraction systems [Morgan *et al.*, 1995]. Various kinds of applications can be built using the system's general-purpose natural language functionalities. The financial information extraction system is an example of use of the natural language processing capabilities of the system for a specific domain.

The basic task of the natural language processing system is to process the input text and produce a representation of its meaning. This representation is then stored in an appropriate knowledge-base and can then be used for various different tasks and to generate natural language text. The core of the system is a large (over 100,000 nodes) semantic network, which consists of a hierarchy of nodes connected with arcs. The nodes represent entities (*a company*) or events (e.g. *The company made a takeover*). Each node is associated to specific control variables which are used to specify the type and properties of each node. Some of the control variables are[1].

- **Rank.** This control gives the nodes quantification, i.e. *individual*, it named individual *universal*, *existential*, *bounded existential* etc. For example, the node "*Roberto*" in the sentence "it Roberto owns a motorbike" shown in figure 3 is a *Named-Individual*

- **Type.** This control is very similar to grammatical qualification and comprises: entity, relation, typeless, event, fact, greeting etc. For example, the sentence "*Roberto owns a motorbike*" shown in figure 3 is a *fact.*

- **Family.** This control groups the nodes into semantic "families" which share specific properties, e.g.: living, animal, human, man-made, abstract, location, organization, human-organization etc. For example, the node "*Roberto*" shown in figure 3 belongs to the family "*human*".

[1] A more complete description of the control variables and, in general, of the Semantic Network can be found in [Morgan *et al.*, 1995]

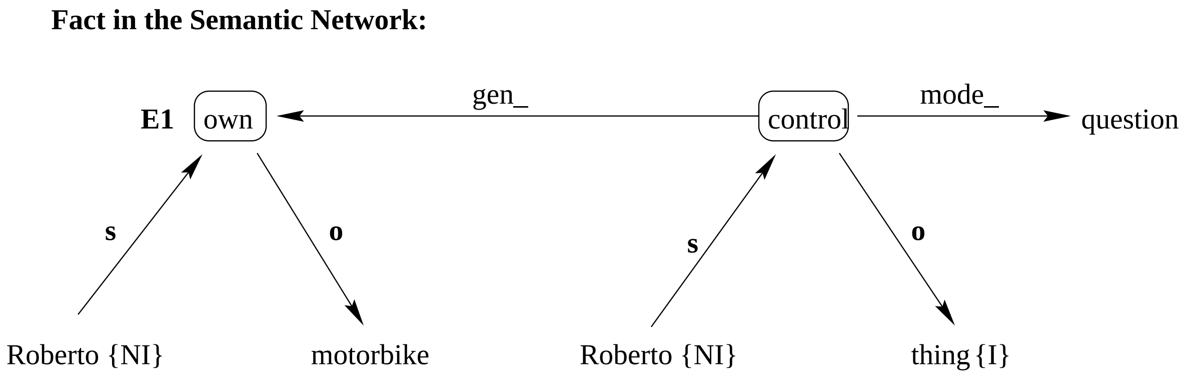

Figure 3: Representation in the semantic network of the sentence "Roberto owns a motorbike"

The source articles are processed by the system by various hierarchical modules: morphology, parsing, semantics and pragmatics (figure 2).

- The *morphology* module splits the input text into words and smaller units and produces for each word a list of possible meanings for the word together with their syntactic and semantic categories. The input is then supplied to the parser.
- The *parser* module performs a full grammatical analysis of the source sentence recognizing the role of each of the words in the sentence, for example subject, object, verb, adjective etc. At this stage, the meaning of each of the words in the sentence is not yet determined, and will be resolved by the subsequent modules of the analysis.
- The *semantic analysis* module associates each of the words with their appropriate meanings and maps them onto the system's internal representation in a format compatible with the semantic network.
- Finally, the *pragmatic analysis* module performs the disambiguation of the meanings introduced by the semantic analysis module and type checking.

At the end of the analysis process the new knowledge is stored in the semantic network. To produce the templates, the new knowledge obtained from the analysis of the sources articles is matched against the templates definitions. Two different approaches are taken depending on the kind of template being analyzed. In the case of the pre-defined templates described in section 2, the system will try to match the new knowledge against the templates definitions coded in the system [Costantino *et al.*, 1996a] using specific *inference rules*. Differently, user-defined templates such as the takeover template described in section 3 are filled using the "*inference engine*", which will match the user natural language template definition against the source article [Costantino *et al.*, 1997].

In both cases, the system will try to identify an event which is equivalent to the event defined for each of the templates definitions. For example, for a *takeover* template the system will try to identify an event such as "A company bought another company" and the information associated to the event, such as the companies involved, the cost of the takeover etc. The system will therefore try to identify an event that has a *takeover-action* (e.g. *to buy*, *to take-over*, *to purchase*, etc.) or any other action that can be generalized to these, and the object is a company. The event shown in figure 4 will therefore be selected as relevant event for the takeover template. Finally, the templates are filled using the English generator which produces natural language English text using the knowledge stored in the semantic network.

The system is written in the functional programming language Haskell and runs on a Sun SparcStation with 80MB of Ram. However, it can easily be adapted for use within other Unix environments.

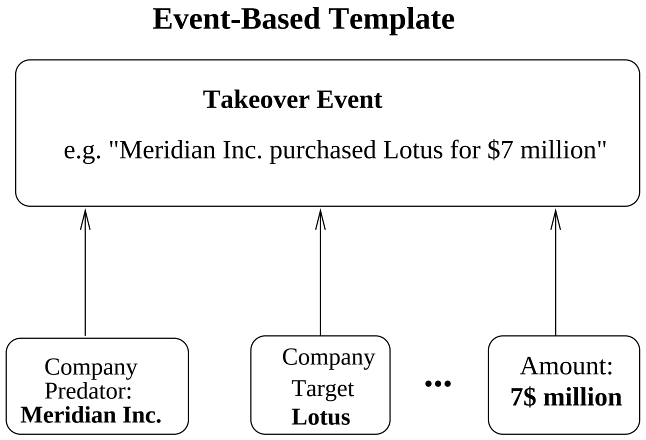
Event-Based Template
Takeover Event
e.g. "Meridian Inc. purchased Lotus for $7 million"
Company Predator: Meridian Inc.
Company Target Lotus
...
Amount: 7$ million

Figure 4: A candidate event for the takeover template.

## 5 Conclusions

In this paper we have shown how natural language processing and, more specifically, information extraction can be used in finance. The use of information extraction can be particularly useful for the financial operators who have to deal with the increasing quantity of information available today. By processing the source articles using an information extraction system, the operators can obtain a summary of the most important information without having to read the whole article. The information extracted from the source texts can be used for an analysis of the effects of news on price behavior.

## References

[Costantino *et al.*, 1996a] M. Costantino, R. J. Collingham, and R. G. Morgan, "Information Extraction in the LOLITA System using Templates from Financial News Articles", in *Information Technology Interfaces '96*, June 1996.

[Costantino *et al.*, 1996b] M. Costantino, R. J. Collingham, and R. G. Morgan, "Qualitative Information in Finance: Natural Language Processing and Information Extraction", *NeuroVe$t Journal*, 4 No.6, November 1996.

[Costantino *et al.*, 1997] M. Costantino, R. G. Morgan, and R. J. Collingham, "Financial Information Extraction using pre-defined and user-definable Templates in the LOLITA System", *CIT - Journal of Computing and Information Technology*, February 1997.

[Morgan *et al.*, 1995] R. Morgan, R. Garigliano, P. Callaghan, S. Poria, M. Smith, A. Urbanowicz, R. Collingham, M. Costantino, C. Cooper, and The LOLITA Group, "University of Durham: Description of the LOLITA System as used in MUC-6", in *Sixth Messages Understanding Conference (MUC-6)*, Morgan Kaufmann, November 1995.

[Riloff and Lehnert, 1994] E. Riloff and W. Lehnert, "Information Extraction as a Basis for High-Precision Text Classification", *ACM Transactions on Information Systems*, 12 No.3:296–333, 1994.