

# IDENTIFYING AND EXTRACTING RELATIONS IN TEXT

Roy J. Byrd and Yael Ravin <sup>1)</sup>

Much current research in the fields of text analysis and information retrieval is motivated by the growing need to browse, skim and sift through great quantities of information. Within this goal, our work focuses on extracting information from large collections of documents: salient concepts and entities mentioned in the documents as well as the relationships which hold among them. We describe here our work on the identification and extraction of relations.

## 1. Extracting Concepts and Entities

A cascade of extractors, collectively referred to as *Texttract*, identifies domain terms mentioned in the document collection; proper names as well as their type, such as people, places or organizations; abbreviations; and other special single words. When combined, the information gathered by the extractors represents the lexical labels used for concepts and named entities discussed in the text. We refer to these simply as *concepts* below. Their frequency of occurrence in the collection as a whole and in each document is used to identify the most salient ones, which are used to distill the information presented to the user: for example, highlighted salient concepts make documents easier to skim and browse, while a list of the top  $n$  ones provides an indication of what the document is about. In addition, salient concepts are one of the criteria we use for selecting important sentences to display to the user as a document summary [3].

The module that identifies concepts, or technical terms, scans documents for all sequences of words that match grammatical structures in which technical terms tend to appear. These structures can be captured as sequences of this form:

$$((A|N)^+|((A|N)^*(NP)?)(A|N)^*)N$$

where A is an adjective, N is a noun, and P is a preposition [1].

The name extractor [5] considers every sequence of capitalized words (with some exceptions) as a potential name. Names are grouped in sets, associated with a single referent, of a given type, such as a person or an organization. Sets include a canonical form and other variants. For example, *Treasury Secretary Nicholas Brady*, *Secretary Brady* and *Mr. Brady* are all variants of the canonical form *Nicholas Brady*.

---

<sup>1</sup> {byrd, yael}@watson.ibm.com, T.J. Watson Research Center, POB 704, Yorktown Heights, NY 10598, U.S.A

Duplicates and conflicts in the output of the extractors are resolved by a process called aggregation, both at the document level and at the collection level across all the documents.

## 2. Relationships among Concepts and Entities

A simple list of salient terms is very useful for gisting the content of single documents for two reasons. First, the list is relatively short and can be conveniently displayed. Second, the full text of the document can be easily made available for showing the term in context, that is, in relation to other concepts and ideas discussed. But simple lists are much less convenient for browsing the content of collections of documents: there are many terms and they are mentioned in many contexts. To provide a structure that can be navigated, we process the contexts in which salient concepts have been identified and extract both statistically significant co-occurrence relations as well as lexical relations with other concepts. These relations are triples, containing two concepts, and either a relation strength (*unnamed relations*) or a relation name (*named relations*). Out of these triples we build a network whose nodes are entities and concepts and whose arcs, connecting the nodes, are the relationships. The network can be displayed graphically, with users clicking on a node to see its relationships to other nodes, as shown in *Figure 1*. Or it can be displayed as lists of related terms, which users can navigate using standard user interface metaphors.

Unnamed relations express the fact that two concepts are more or less strongly related, although the exact nature of the relationship is unknown. In the current implementation [2], we derive unnamed relations from a document collection by calculating the mutual information for pairs of concepts. The relations extractor asserts that two concepts are related if the normalized mutual information value of their cooccurrence is sufficiently high, as in:

<Louis V. Gerstner : R95 : International Business Machines>

Named relations are stipulated when a concept occurs in a certain expected grammatical pattern in the text, such as a possessive construction or an appositive phrase. When concepts are encountered in a pattern, a triple is extracted: two concepts and a relationship that is found to hold between them. For example, the following pattern

PERSON, ... of ORGANIZATION,

is matched by this occurrence in the text:

Today, Gerstner, the CEO of IBM, announced that the company....

to yield the following relation:

<Louis V. Gerstner : CEO : International Business Machines>

assuming that *IBM* is a variant of *International Business Machines* and *Gerstner* of *Louis V. Gerstner*.

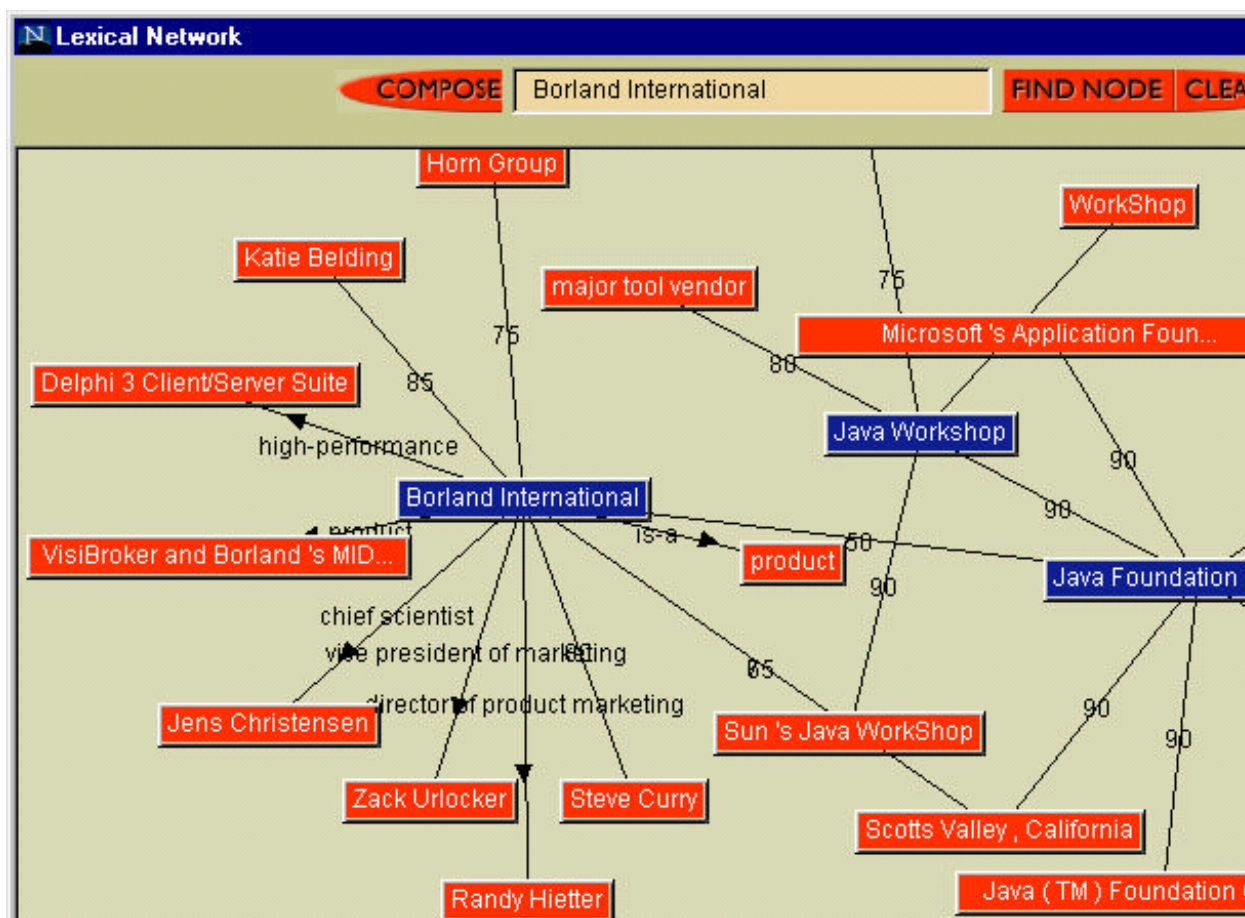


Figure 1: Network of Concepts and Relations

### 3. Techniques for Extracting Named Relations

Textract's concept extractors operate on a linked list of tokens by annotating it with concepts. Named relation processing applies to the annotated wordlist by looking for patterns with specially-built finite state automata. It is the use of finite-state automata, rather than full syntactic parsing, that gives this process the speed required to meet Textract's performance goal.

There are three broad classes of pattern that are used in the current implementation to identify named relations.

- Patterns which anchor the positions of both concepts and allow for the discovery of the relation name in the remainder of the pattern, as in the *Gerstner* example above.
- Patterns which anchor the positions of both concepts within a constant pattern and yield a fixed relation name. An example would be the *location* relations found in "-based" constructions; for example, the string "... Armonk, New York-based IBM ...", yielding

<Armonk, New York : location : International Business Machines.>

- Patterns which anchor the position of one of the concepts and of a fixed relation name and allow for the discovery of the second concept. For example, the pattern “ORG MAKE-VP ...” when applied to “...IBM, which manufactures computing equipment ...”, yields

<International Business Machines : make : computing equipment >

Here is a sample of named relations extracted from Wall Street Journal documents[4].

AAI : unit : United Industrial

Albert E. Suter : president and chief operating officer : Firestone Tire & Rubber

Alcan Aluminium : sell : aluminum

Alex Henderson : analyst : Prudential-Bache Securities

Alexia Morrison : prosecutor : Theodore Olson

Allegheny International : make : Sunbeam appliance and consumer product

Allen F. Jacobson : chairman : 3M

American Express : own : Fireman

Anthony J. Gajda : principal : Mercer Meidinger Hansen

Anthony P. St. John : head of labor relations : Chrysler

Archie R. Boe : former president : Sears , Roebuck

Finite-state processes alone are not sufficiently constrained to identify and resolve all ambiguity that occurs at the level of description represented by annotated wordlists. As an example, although many sequences of words bounded by commas are appositives, not all are. A relation extractor that depends on appositives risks making mistakes if it encounters a comma-delimited conjunct in a long conjunction, for example.

To handle such problems, and to help insure the accuracy of extracted relations, we use a number of filters to reject bad candidates. The types of filters we use are:

- *frequency filters*, to reject, for example, a relation name that occurs only once and may therefore be just an accidental string of words;
- *lexical and morphological filters*, to require, for example, that the lexical head of the verb phrase be a verb of manufacturing or selling;
- *selectional restrictions*, to require a place name, for example, as the first concept in a location relation;
- Other filters are *coordination censors*, and *length filters*.

The process of extracting named relations in itself contributes to enhancing the quality of the resulting network, in the following ways:

- More concepts are added to the original set of concepts, as when we discover what it is that companies we know about manufacture.

- Named entities may be better categorized, if they are found in patterns we expect. For example, an entity of an unknown type can be “upgraded” to become a person if it is in a *CEO-of* relationship with an organization.
- Ambiguous coordinate structures are disambiguated and split into their components.

## 4. Evaluation

As described above, unnamed relation extraction assigns a higher relation strength to relations in which it has more confidence, by virtue of the fact that the related concepts cooccur more frequently than would be expected by chance. It turns out that the proportion of unnamed relations for which there are matching named relations increases as the strength of the unnamed relations increases. In other words, named relations correlate strongly with our confidence measure for unnamed relations, a result that we would expect if our named relation extraction is doing a good job. A more precise characterization of this correlation is given in [2].

## 5. References

- [1] JUSTESON, J.S. and S. KATZ, Technical Terminology: Some Linguistic Properties and an Algorithm for Identification of Terms in Text, in *Natural Language Engineering*, 1. 9-27, 1995.
- [2] KAZI, Z., E.W. BROWN and R.J. BYRD, Discovering Unnamed Relations between Vocabulary Items in Large Text Collections, in preparation.
- [3] NEFF M.S. and J.W. COOPER, Automatic Text Summarization (this volume).
- [4] Tipster Information-Retrieval Text Research Collection, CD-ROM, NIST, Gaithersburg, Maryland.
- [5] RAVIN, Y. and N. WACHOLDER, Extracting Names from Natural-Language Text. *IBM Research Report 20338*, 1996.