Mining biomedical literature using information extraction

Ronen Feldman, Yizhar Regev, Michal Finkelstein-Landau, Eyal Hurvitz & Boris Kogan
ClearForest Corp, USA & Israel

Text mining is the process of analyzing unstructured, natural language texts in order to discover information and knowledge that are difficult to retrieve directly. Information extraction is one of the most important techniques used in text mining. Natural language processing tools, augmented by lexical resources and semantic constraints can be used to build effective information extraction modules for mining biomedical literature. Visualization tools enable the user to explore, check (and correct if required) the results of the text mining process effectively.

In this information age it is easy to store large amounts of data electronically, but the proliferation of documents available on the web, on corporate intranets, on newswires, and elsewhere has become overwhelming. Search engines only exacerbate this by making more and more documents available in a matter of a few keystrokes. For example, the NCBI PubMed archive lists over 10,000 papers that mention 'epidermal growth factor receptor'.

Text mining

Text mining is a new and exciting research area that attempts to solve the information overload problem. It uses many techniques from data mining, but since it deals with unstructured data, a major part of the text mining process deals with the crucial stage of preprocessing the document collections (using techniques such as text categorization, term extraction, and information extraction). The process also involves the storage of the intermediate repesentations, techniques to analyze these intermediate representations (such as distribution analysis, clustering, trend analysis, association rules etc) and visualization of the results (see Figure 1).

Text mining pipeline

A typical text mining system begins with collections of raw documents, without any labels or tags. Documents are then automatically tagged by categories, terms or relationships extracted directly from the documents. Next, extracted categories, entities and relationships are used to support a range of data mining operations on the documents.

Information extraction

The most common tagging approach is information extraction (IE), that is, processing each document to find (extract) entities and relationships that are likely to have meaning in the given domain. By 'relationships' we refer to facts or events involving certain entities. A possible 'event' may be that a company has entered into a joint venture. A 'fact' may be that a gene causes a certain disease. The extracted information provides much more concise and precise data for the mining process than in a word-based approach such as text categorization and tends to represent concepts and relationships that are more meaningful and relate directly to the examined document's domain.

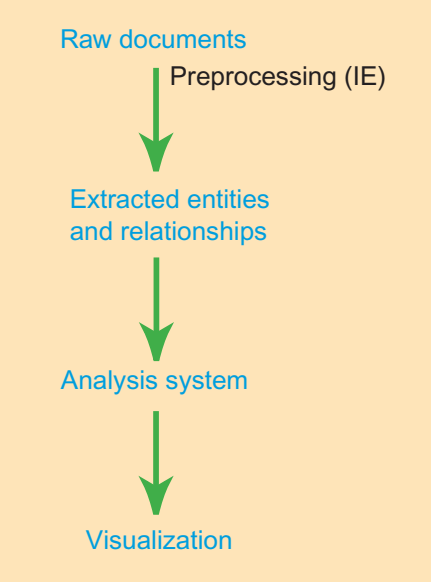

Figure 1. The text-mining pipeline.

In contrast to the document categorization approach, the IE method allows for mining of the actual information present within the text, rather than the limited set of tags associated to the documents. Using the IE process, the number of different relevant entities and relationships on which the data mining is performed is unbounded, typically thousands or even millions, far beyond the number of tags which any automated categorization system could handle.

Biomedical literature mining

In this article, we focus on using text mining techniques for mining biomedical literature. In particular, we are interested in finding relationships between genes, proteins, drugs and diseases. The input to the system is a set of biomedical articles, which are then analyzed by the IE module and all entities and relationships extracted from them. In the next section, we describe how to develop IE modules that extract biomedical entities and relationships from biomedical articles. We con-

clude with a description of an actual text mining system, called **ClearResearch**, that works on collections of MEDLINE abstracts. This tool performs IE using MEDLINE abstracts and enables visual navigation within the generated knowledge base.

Developing IE modules

In IE, key concepts (entities or relationships between entities discussed in the text) are defined in advance and then the text is searched for concrete evidence of the existence of such concepts. For example, the KDD Cup 2002 competition dealt with the curation task of the FlyBase consortium. The FlyBase curators wanted to find, within PubMed scientific papers, specific evidences for experimental results regarding products of *Drosophila* genes in wild-type flies. The papers are searched for patterns indicating a report of a gene product and when such a phrase is found, the relevant attributes are 'extracted' from that phrase (see Figure 2). Thus, structured information is created from the unstructured text.

Different approaches to IE

There are several algorithms and methods to perform IE. Generally, the existing methods can be divided into unsupervised (often called statistical) methods and rule-based methods. The Hidden Markov model is the best known unsupervised method; here the system 'learns' a statistical/probabilistic model that explains how the relevant entities or relationships are created. Rule-based methods, on the other hand, are based on rules (written by human developers) that capture syntactical, lexical and semantic knowledge required for identifying the entities and the relationships. For tasks that require only the extraction of independent entities (genes/proteins, drugs etc), both approaches proved relatively successful, and therefore it is often argued that unsupervised methods are preferable for these tasks because they are less (human) labor-intensive. However, for tasks requiring extraction of relationships (expression of a specific product of a certain gene in a certain tissue, specific function of a protein etc), rule-based methods are far superior. The rule-based approach is the main one we use to implement IE modules, such as the one used for the KDD Cup 2002 competition. We elaborate on this approach below.

A sample of relevant free text in the original document:

"Fig 4: Expression of dGATAc transcripts in the developing midgut"

The template with a corresponding instance:

Gene	Product type	Tissue
dGATAc	Transcript	developing midgut

Figure 2. IE template for the KDD Cup 2002 competition.

Rule-based IE system layout

Layer 0 - part of speech (POS) tagger. Assigning POS tags (noun, proper noun, verb, adjective, adverb, preposition, and so on) to each word. For example, in 'expression of dGATAc transcripts', 'expression' and 'transcripts' are nouns and 'dGATAc' is a proper noun.

Layer 1 - noun phrase and verb phrase grouper. Grouping together the head noun with its left modifiers (for example: 'the developing midgut') and, for verbs, chunking a main verb with its auxiliaries (as in 'does not antagonize').

Layer 2 - verb and noun pattern extractor. Extracting larger verb and noun phrases, based on semantic requirements. For example, 'Dac does not antagonize Dll expression'. In general, this extractor matches verbs and nouns with their complements, as specified in their sub-categorization properties. This level is semantically-oriented: It keeps track of the semantic features of a pattern as expressed by various elements such as adverbs, tense and voice of the verb group and certain syntactic structures.

Layer 3 - named entity recognizer. This is the recognition of the entities relevant to the domain. In the biomedical domain, these entities would be genes, proteins, diseases and so on. Other domains will naturally have different entities, for example, the typical entities in the financial news domain are companies, persons, products, and so forth.

Layer 4 - template ('relationship') extractor. Rule-based extraction of patterns at a full sentence or phrase level, using the components found at previous layers. For example, in the KDD competition, it was required to separate evidences of gene expression in the wild-type fly from evidences for induced/dependent expression achieved due to an artificial intervention of the researcher, for example, 'Dac does not antagonize either Dll or hth expression in the antenna'. In order to identify such template, we look for subject-verb-object structure, requiring the subject and object to be names of genes and the verb to be tensed where its head belongs to a lexicon including various verbs indicating an induction/influence between genes (eg, antagonize, inhibit, repress).

Structure-driven rule-based strategy

This strategy is based on identification of natural language elements (noun phrase and verb phrase) augmented by linguistic and semantic constraints. The extraction of the predefined semantic relationships is performed by means of deep syntactic and semantic analysis of the sentences. Implementation of the structure-driven processing is based on a general multi-level natural language processing system and, as detailed in the box, the fact that the system is layered, enables easy adaptation for new entities and relationships or even a new domain. Only the lexical and semantic sources that are unique to the specific entities or relationships should be replaced.

Using generic, syntax-based templates

Writing patterns for all possible lexical and semantic combinations of a certain relationship (Layer 4 and above) can be quite time-consuming. Sometimes, as in the KDD Cup competition, we are interested only in a certain semantic relationship (such as the natural expression of a gene product). Often, however, we are not interested in a particular relationship, or the number of possible relationships between every pair of entities is too big. In this case, we can extract a generic template based on entities within a simple syntax-based template, and then explore the results using a text-mining tool such as ClearResearch.

VerbalRelation template

Consider a collection of MEDLINE abstracts. While lexicons for the relevant entities (gene, tissue, disease etc) are readily available, modeling all the possible relationships between them is quite difficult. Instead, we use a generic template – **VerbalRelation**, which extracts two noun phrases (NP1 and NP2) connected by a verb (and possibly a suitable preposition). Then, we can classify the noun phrases according to the lexicons to which their terms belong (eg, genes or diseases). For example, 'MC3-R is potently activated by gamma-MSH peptides'. 'MC3-R' will be extracted as NP1, and 'gamma-MSH peptides' will be extracted as NP2. The verb is 'activated'. Additional examples can be seen in Figure 3.

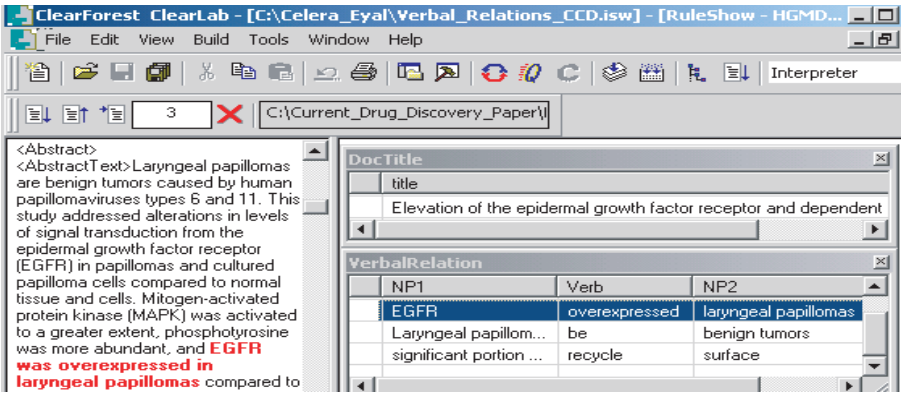

Figure 3. The ClearLab application demonstrating the VerbalRelation template.

Implementation of DIAL

The framework we use for implementing our IE system is a rule-based, general IE language, developed at ClearForest called **DIAL** (Declarative Information Analysis Language). The 'building blocks' of DIAL are rules. Rules are sequences of pattern matching elements, augmented by a set of constraints that the matched patterns must obey and by a set of assignments of the rule's parameters and/or actions concerning external variables/data structures.

The pattern matching elements themselves can be either literal strings found in the text (for example, the word 'expression'), a lexicon (known in DIAL as wordclass), or another rule. The complete syntax of DIAL is beyond the scope of this article: A sample DIAL rule is presented in Figure 4. DIAL enables the user to implement separately the different operations required for performing IE: tokenization, sectioning (recognizing paragraph and sentence boundaries), and morphological and lexical processing, parsing and domain semantics. DIAL has built-in modules that perform the general tasks of tokenization and part-of-speech tagging. In addition, we have developed a general library of rules that perform noun phrase and verb phrase grouping and separate libraries for recognizing common entities, such as companies or persons. An IE module incorporating the infrastructure libraries and specific customized rules for a specific domain or task is called a rulebook.

FURTHER READING

Collier N, Nobata C & Tsujii J (2000) **Extracting the names of genes and gene products with a hidden Markov model.** *COLING* 201-207.

Feldman R *et al* (2002) **A comparative study of information extraction strategies.** *CICLing* 349-359.

Feldman R *et al* (2000) **A framework for specifying explicit bias for revision of approximate information extraction rules.** *KDD 2000* 189-199.

Humphreys K, Demetriou G & Gaizauskas R (2000) **Two applications of information extraction to biological science journal articles: Enzyme interactions and protein structures.** *Pacific Symposium on Biocomputing 2000* 505-516.

DIAL rule example

Induced expression - The gene expression is induced or induces another activity (and is NOT observed on its own), as in: 'Fig. 4. Dac does not antagonize hth expression in the antenna.'

```
//lexicon for relevant nouns similar to 'expression':
wordclass wcExpressionNoun = expression transcription localization detection;

//lexicon for verbs indicating induction/interaction between genes as 'antagonize';
wordclass wcInducedVerbs = reduce inhibit activate induce repress alter antagonize;

//extract Noun Phrase (NG-Noun Group) incorporating a gene
GeneExpressionNG():-
ExtractedGene(Gene,Product)                    //'Dac' (The gene)
NounGroup(Article,Head,Stem)                   //'expression'
verify(InWC(Head,@wcExpressionNoun));          //verify that the Head is relevant

//Rule for the induced Expression itself
Induced_Expression() :-
ExtractedGene(Gene,Product,mutant)             //'Dac'
VerbGroup(Stem,Tense,Aspect,Voice,Polarity)    //verb group - 'does not antagonize'
GeneExpressionNG                               //'hth expression'
 verify(InWC(Stem,@wcInducedVerbs));           //verify that the Stem is indeed a //relevant verb
```

Figure 4. Sample DIAL rule from the KDD Cup competition.

Rulebook development applications

ClearForest has two development environments for DIAL rules – **ClearLab**, (see Figure 3), aimed at developers wanting to build rulebooks for new domains and applications and **ClearStudio**, aimed at end-users who want to customize existing rulebooks without the need to write DIAL code themselves. In Figure 5, we show the pattern that was developed in **ClearStudio** for extracting a causality relationship between gene-mutation and disease. For example, 'These data suggest that the T9997C mutation in mtDNA is causative of respiratory chain dysfunction when present at high levels of heteroplasmy' (an extract from a MEDLINE abstract).

We have created several small lexicons such as CausalityWC and MutuationWC and used two predefined lexicons that included Genes (GenWC) and Diseases (DiseaseWC). Both applications enable the user to test the rulebook on relevant document collections, view the extracted instances of the various templates and their location in the original text. This information allows the user to change and correct (debug) the rules as necessary.

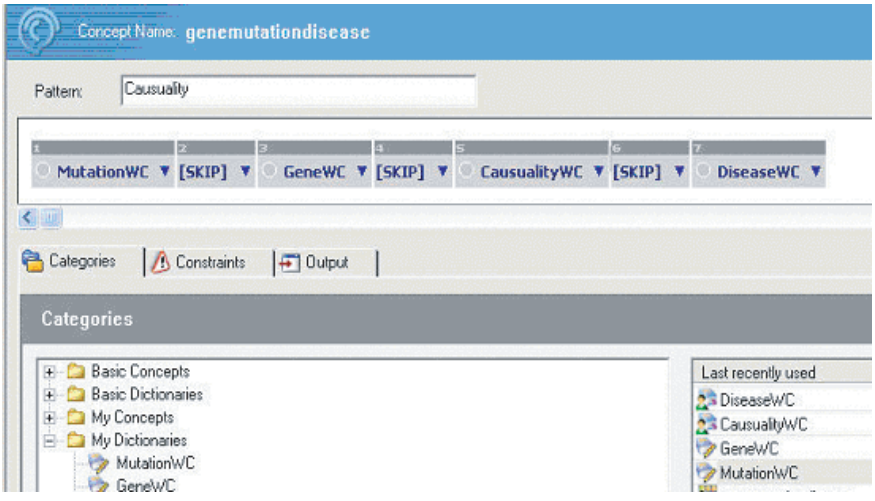

Figure 5. The ClearStudio application demonstrating the casualty relationship.

Text mining in action

When developing a text mining system one of the crucial needs is the ability to browse through the document collection and to be able to visualize the various elements within the collection. This type of interactive exploration enables one to identify new types of entities and relationships that can be extracted and better explore the results of the IE phase.

Visualization tool

By using the **ClearResearch** visualization tool, which enables the user to visualize relationships between entities that were extracted from the documents, one can view collocations between entities or a semantic map that will show entities that are related by any of a defined set (user definable) of relationships.

Relationship maps provide a visual means for concise representation of the

relationship between many terms in a given context. In order to define a relationship map, the user must define a taxonomy category (eg, 'genes') that determines the nodes of the graph. As well as an optional context node (eg, 'phosphorylation') that will determine the type of connection he/she wishes to find among the graph nodes. If no context is provided, the system will revert to using co-occurrence information between entities.

Two entities co-occur within some lexical unit (a sentence, paragraph or document), if they both are contained inside that lexical unit. The most common lexical level for co-occurrence computation is the sentence level. Entities that appear within the same sentence are said to be co-occurring in the sentence level. Figure 6 is an example of relationship map between genes; this figure depicts the co-occurrence (within the same sentence) relationships between gene phrases in the context of any type of cancer. The darker the edge between two phrases, the more frequent their co-occurrence is.

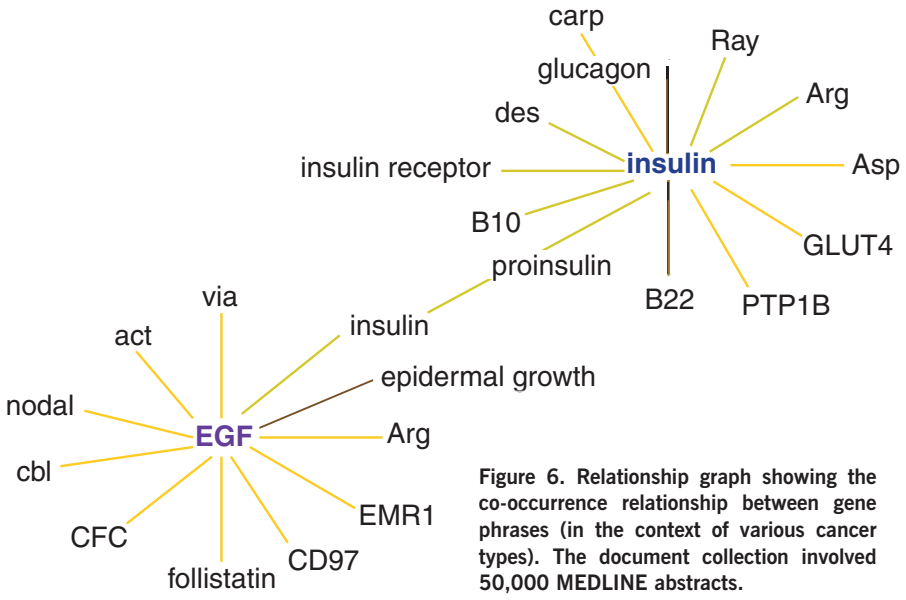

Figure 6. Relationship graph showing the co-occurrence relationship between gene phrases (in the context of various cancer types). The document collection involved 50,000 MEDLINE abstracts.

Machine-assisted indexing

No IE system is 100% correct. Whatever approach is taken, there will be always instances (of entities or relationships) that the system will miss and some incorrect (false positive) instances that will nevertheless be extracted. The reason for this is the complex nature of human language – a computerized system will never be able to trace all the possible phrasing and contexts used by humans, and of course, use all the domain expert humans have. Therefore, for many applications it is useful to give human experts the opportunity to review the results tagged or extracted by the IE system. This is particularly useful for areas where a large amount of domain expert is required, such as in the biomedical domain. In Figure 7, a machine-assisted indexing (extraction) system, suggested for the FlyBase Curation task is shown. On the left, the expert can view the location of the extracted instance within the original paper in a PDF format (with the original figures). The expert can thus decide whether the instance was extracted correctly or incorrectly. He/she doesn't have to review the whole paper, only the relevant sections suggested by the system.

See the forest and the trees

Due to the abundance of biomedical textual data, there is a growing need for efficient tools for text mining. Unlike structured data, where the data mining algorithms can be performed directly on the underlying data, textual data requires some preprocessing before the data mining algorithm can be successfully applied. IE has proved to be an efficient method for this preprocessing phase and its results are better than those of pure word-based approaches. Text mining using IE thus hits a useful middle ground on the quest for tools for understanding the information present in the large amount of data that is only available in textual form. The powerful combination of precise analysis of the biomedical documents and a set of visualization tools enable the user to easily navigate and utilize very large biomedical document collections.

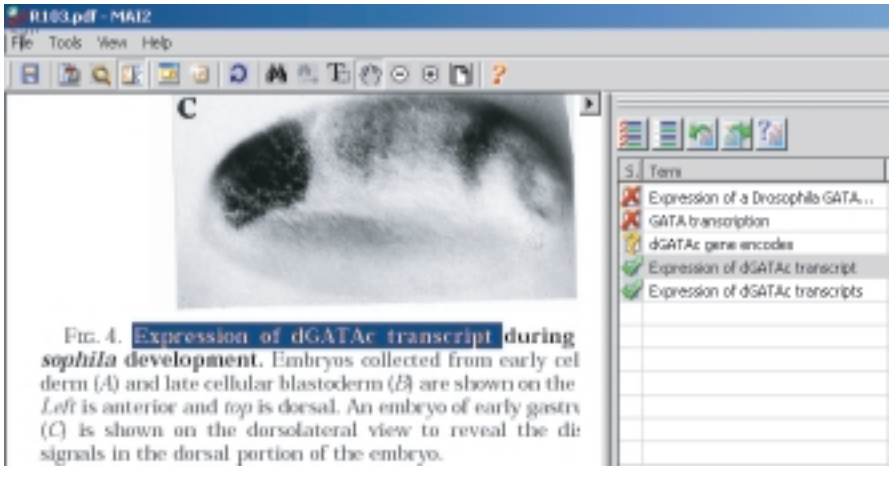

Figure 7. The machine-assisted indexing (MAI) application.

Ronen Feldman
President and Chief Scientist
ClearForest Corp
Corporate Headquarters
15 E 26th Street
Suite 1711
New York
NY 10010
USA

Email: ronen@clearforest.com

FURTHER INFORMATION

ClearForest
www.clearforest.com

KDD Cup 2002 competition
www.biostat.wisc.edu/~craven/kddcup