An Integrated Term-Based Corpus Query System

Irena Spasic
Computer Science
University of Salford
I.Spasic@salford.ac.uk

Goran Nenadic
Dept. of Computation
UMIST
G.Nenadic@umist.ac.uk

Kostas Manios
Computer Science
University of Salford
K.Manios @salford.ac.uk

Sophia Ananiadou
Computer Science
University of Salford
S.Ananiadou@salford.ac.uk

Abstract

In this paper we describe the X-TRACT workbench, which enables efficient term-based querying against a domain-specific literature corpus. Its main aim is to aid domain specialists in locating and extracting new knowledge from scientific literature corpora. Before querying, a corpus is automatically terminologically analysed by the ATRACT system, which performs terminology recognition based on the C/NC-value method enhanced by incorporation of term variation handling. The results of terminology processing are annotated in XML, and the produced XML documents are stored in an XML-native database. All corpus retrieval operations are performed against this database using an XML query language. We illustrate the way in which the X-TRACT workbench can be utilised for knowledge discovery, literature mining and conceptual information extraction.

1 Introduction

New scientific discoveries usually result in an abundance of publications verbalising these findings in an attempt to share new knowledge with other scientists. Electronically available texts are continually being created and updated, and, thus, the knowledge represented in such texts is more up-to-date than in any other media.

The sheer amount of published papers[1] makes it difficult for a human to efficiently localise the information of interest not only in a collection of documents, but also within a single document. The growing number of electronically available knowledge sources emphasises the importance of developing flexible and efficient tools for automatic knowledge mining. Different literature mining techniques (e.g. (Pustejovsky et al., 2002)) have been developed recently in order to facilitate efficient discovery of knowledge contained in large corpora. The main goal of literature mining is to retrieve knowledge that is "buried" in a text and to present the digested knowledge to users. Its advantage, compared to "manual" knowledge discovery, is based on the ability to systematically process enormous amounts of text. For these reasons, literature and corpus mining aim at helping scientists in collecting, maintaining, interpreting and curating domain-specific information.

Apart from digesting knowledge from corpora, there is also a need to facilitate knowledge mining via suitable querying systems, which would allow scientists to locate semantically related information. In this paper we introduce X-TRACT (XML-based Terminology Recognition and Corpus Tools), an integrated literature corpora mining and querying system designed for the domain of molecular biology and biomedicine, where terminology-driven knowledge acquisition and XML-based querying are combined using tag-based information management. X-TRACT is built on top of a terminology management workbench and it incorporates a GUI to access the features of the XQuery language that allow users to formulate and execute complex queries against a collection of XML documents.

Our main assumption is that the knowledge encoded in scientific literature is organised around sets of domain-specific *terms* (e.g. names

[1] For example, the MEDLINE database (www.ncbi.nlm.nih.gov/PubMed/) currently contains over 12 million abstracts in the domains of molecular biology, biomedicine and medicine, growing by more than 40.000 abstracts each month.

of proteins, genes, acids, etc.), which are to be used as a basis for corpora querying. Still, few domain-specific corpora mining systems incorporate deep and dynamic terminology processing. Instead, they make use of static knowledge repositories (such as formal taxonomies and ontologies). For example, the queries in the TAMBIS system (Baker et al., 1998) are based on a universal model of molecular biology (represented by a terminology). Our approach relies on dynamic acquisition and integration of terminological knowledge, which is used as the basic infrastructure for further knowledge extraction.

The paper is organised as follows: in Section 2 we describe the related work. X-TRACT is overviewed in Section 3, while terminology processing and querying techniques are presented in Sections 4 and 5 respectively. Finally, Section 6 discusses the details of the applications.

2 Related work

2.1 Querying domain-specific corpora

Various types of scientific literature corpora are widely available with different levels of linguistic and domain-specific annotations. Corpus development tools still occupy much of the research interest, slowly migrating to the systems that integrate both corpus processing and annotation facilities. Up to date, there is a limited number of flexible corpus querying systems. Such systems need to incorporate several components to facilitate more sophisticated corpus mining techniques through flexible processing of annotations and the provision of appropriate query languages.

Traditional, general-purpose corpus querying systems such as CWB (Christ, 1994) provide environments for managing corpora by supplying a query language that can be used to enquire both word/phrase content and the structure of a corpus. Features of such systems include incremental querying and concordancing, possibilities to combine SGML tags and attributes in order to support more sophisticated search. In addition, they have an ability to invoke external applications or resources (such as lexicons or thesauri). Still, additional features intended for domain specialist, rather than linguistically oriented users, are needed.

Few domain-specific corpora-mining systems have been developed. In an attempt to accumulate a large amount of meta-information about documents, such systems usually incorporate several types of tags, which are attached to text in different steps of document processing. The same document may have multiple, possibly interlaced tags, including POS, syntactic and domains-specific (i.e. semantic, e.g. protein, DNA, etc.) tags. Usually, a tagging scheme includes additional structural complexities such as nesting and possible combinations of syntactic and semantic structures (e.g. a noun phrase which contains a DNA name), which may cause difficulties during document processing.

Multi-layered and interlaced annotations have been addressed by several systems, usually by following the TIPSTER architecture (Grishman, 1995), i.e. by manipulating tags via an external relational database (RDB). For example, the TIMS system (Nenadic et al., 2002) addresses terminology-driven literature mining via a RDB, which stores XML-tag information separately from the original documents. The main reasons behind this choice are easy import and integration of different tags for the same document and efficient manipulation of these tags. However, in this paper we will discuss possible advantages of using an XML-native database (DB) to facilitate corpus-mining. The main reasons for this are portability and self-description of XML documents and natural association between them and XML-native databases (see Section 6 for comparison between XML-native DBs and RDBs).

2.2 Terminology extraction and structuring

Corpus mining systems may benefit from the use of a well-formed domain model, which reflects main concepts (linguistically represented by domain-specific *terms*) and relations between them. Such models can be represented by static terminologies or ontologies, which are usually constructed manually. However, documents frequently contain unknown terms that represent

newly identified or created concepts. Automatic term recognition (ATR) tools thus become indispensable for efficient processing of literature corpora, because pre-defined terminological resources could hardly keep up the pace with the needs of specialists looking for information on new scientific discoveries.

There are numerous ATR approaches, some of which rely purely on linguistic information, namely morpho-syntactic features of terms. Recently, hybrid approaches combining linguistic and statistical knowledge (e.g. (Frantzi et al., 2000)) are steadily taking primacy. In general, ATR in specialised domains (e.g. biomedicine) is in line with the state-of-the-art IE results in the named entity recognition: in average, the precision is between 80% and 90%, while the recall typically ranges from 50% to 60%.

One of the main problems that makes ATR difficult is the lack of clear naming conventions in some domains, although some attempts (in the form of conventions and guidelines) in this direction are being made. However, they do not impose restrictions to domain experts. In addition, they apply only to a well-defined, limited subset of terms, while the rest of the terminology usually remains highly non-standardised.

In theory, terms should be mono-referential (one-to-one correspondence between terms and concepts), but in practice we have to deal with *ambiguities* (i.e. homography - the same term corresponds to many concepts) and *variants* (i.e. synonymy - many terms leading to the same concept). If we aim at supporting systematic acquisition and structuring of domain-specific knowledge, then handling term variation has to be treated as an essential part of terminology mining.

Few methods for term variation handling have been developed (e.g. the BLAST system (Krauthammer et al., 2000) and FASTR (Jacquemin, 2001)). In particular, a very common term variation phenomenon in some domains is the usage of acronyms. However, there are no strict rules for defining acronyms, and few methods for acronym acquisition have been developed only recently attracting much of the attention especially in the biomedical domain (e.g. (Pustejovsky et al., 2002; Nenadic et al., 2002; Chang et al., 2002)).

In order to make full use of automatically extracted terms, they need to be related to existing knowledge and/or to each other. This means that semantic roles of terms need to be discovered, and terms should at least be organised into clusters or classes. The automatisation of this process is still an open research issue.

3 An Overview of X-TRACT

The X-TRACT system has been developed with the objective of addressing the problems of terminology-based corpus mining in the domain of biomedicine. X-TRACT can be viewed as both a core engine and a GUI for a conceptual IE system.

Corpus querying in X-TRACT is mainly based on terminological processing performed by ATRACT (Mima et al., 2001). The role of ATRACT is to identify and organise terms from a plain-text corpus and to tag them together with their syntactic and semantic attributes. These terms are further used as a basis for corpus mining. The results produced by ATRACT are encoded in XML and then managed by X-TRACT by storing all XML-tags in an XML DB.

Additionally, X-TRACT implements a GUI allowing users (typically experts in biomedicine) easy formulation of queries. The format of XML documents and the corresponding GUI-driven query formulation offer a flexible way of querying a terminologically processed corpus.

The corpus mining process is performed in the following steps:

- A literature corpus is POS tagged, and basic syntactic chunks are marked (the EngCG tagger is used).
- Terms (including variants and acronyms) are automatically recognised and annotated in the corpus.
- Term similarities are calculated for the extracted terms, and they are clustered accordingly. Clustering information is stored within the documents.
- XML-tag information is imported into an XML-native DB (the X-Hive DB 3.0).

- Query composer is used to formulate queries against the XML DB and to translate them into XQuery.
- After running a query, users are offered a possibility to update the existing knowledge-bases (e.g. ontologies and/or terminologies), or to save the query for further use.

The GUI interface layer utilises dynamic recognition of terms and their clusters, as well as an unrestricted set of tags that can be used for querying. On the other hand, other systems that use GUI-driven query formulation, such as TAMBIS (Baker et al., 1998), usually use a pre-defined ontology impose restrictions on query definition. X-TRACT, however, rather than being limited to a static knowledge repository, uses dynamic organisation of domain knowledge and adjusts itself to a given corpus.

In the following sections we provide an overview of the X-TRACT components.

4 Terminological processing

Terminological processing in X-TRACT is performed by ATRACT in two steps. In the first step, domain-specific terms are automatically recognised in a corpus. In addition, term variants (including acronyms) are linked to their normalised representatives. In the second step, extracted terms are automatically structured in a set of domain-specific clusters grouping functionally similar terms together.

4.1 Automatic term recognition

Our approach to ATR is based on the C- and NC-value methods (Frantzi et al., 2000), which extract multi-word terms. The *C-value* method recognises terms by combining linguistic knowledge and statistical analysis. It is implemented as a two-step procedure. In the first step, term candidates are extracted using a set of linguistic filters, which describe general term formation patterns. In the second step, the term candidates are assigned termhoods (referred to as C-values) according to a statistical measure. The measure amalgamates four numerical corpus-based characteristic of a candidate term, namely the frequency of occurrence, the frequency of occurrence as a substring of other candidate terms, the number of candidate terms containing the given candidate term as a substring, and the number of words contained in the candidate term.

The *NC-method* further improves the C-value results by taking into account the context of candidate terms. The relevant context words are extracted and assigned weights based on how frequently they co-occur with top-ranked term candidates extracted by the C-value method. Subsequently, context factors are assigned to candidate terms according to their co-occurrence with top-ranked context words. Finally, new termhood estimations (referred to as NC-values) are calculated as a linear combination of the C-values and context factors for the respective terms. Evaluation of the C/NC-methods has shown that contextual information improves term distribution in the extracted list by placing the actual terms closer to the top of the list.

4.2 Term normalisation

We have incorporated term variation handling into the ATR process by enhancing the original C-value method with term normalisation. All occurrences of term variants are matched to their normalised form and considered jointly for the calculation of termhoods.

A variety of sources (see Table 1) from which term variation problems originate are considered. Each term variant is normalised, and term variants having the same normalised form are then grouped into classes in order to link each term candidate to all of its variants. A list of term variant classes, rather than a list of single terms is statistically processed, and the termhood is calculated for a whole class of term variants, not for each term variant separately.

Variation type	Examples	
	Term variants	**Normalised term**
orthographical	all-trans-retinoic acid all trans retinoic acid	all trans retinoic acid
morphological	Down syndrome Down's syndrome	Down syndrome
syntactic	clones of humans human clones	human clone
lexico-semantic	cancer carcinoma	cancer
pragmatic	all-trans-retinoic acid ATRA atRA	all trans retinoic acid

Table 1: Term variation

Variation recognition also incorporates the mapping of acronyms to their expanded forms. Our method for acronym acquisition is based on

both morphological and syntactic features of acronym definitions (see (Nenadic et al., 2002) for details). We rely on syntactic patterns that are predominantly used to introduce acronyms in scientific papers in order to locate potential acronym definitions. Once a word sequence matching such a pattern is retrieved, it is morphologically analysed with the aim of discovering the link between potential acronym and its expanded form. Both acronyms and their expanded forms are normalised with respect to their orthographic, morphological, syntactic and lexico-semantic features. The acronym acquisition has been embedded into the ATR process as the first step, in which each acronym occurrence in a text is mapped to the corresponding expanded form prior to the C-value statistical analysis.

Terms (and term variants)	Termhood
retinoic acid receptor retinoic acid receptor retinoic acid receptors RAR, RARs	6.33
nuclear receptor nuclear receptor nuclear receptors NR, NRs	6.00
all-trans retionic acid all trans retionic acid all-trans-retinoic acids ATRA, at-RA, atRA	4.75
9-cis-retinoic acid 9-cis retinoic acid 9cRA, 9-c-RA	4.25

Table 2: Sample of recognised term and variants

A sample of recognised terms and their variants is provided in Table 2. The precision of the acronym acquisition is around 98% at 74% recall, and the ATR precision improved in average by 2% (resulting in 98% for the top ranked terms) by adding term variation recognition.

4.3 Term clustering

A *cluster* of terms is a group of related terms such that the degree of similarity within an individual cluster is higher then similarity between terms belonging to different clusters. The heart of the clustering problem is the criterion used to measure the coherence of clusters, i.e. similarity between terms, which is to be maximised within an individual cluster.

We used a term similarity measure named the CSL (contextual, syntactical and lexical) similarity (Spasic et al., 2002). The definition of *lexical similarity* is based on having a common head and/or modifier(s). It is useful for comparing multi-word terms, but it is rather limited when it comes to ad-hoc names.

For this reason, we introduce *syntactical similarity*, which is calculated automatically from a corpus. It is based on specific lexico-syntactical patterns indicating *parallel* usage of terms. Several types of parallel patterns are considered: enumeration expressions, coordination, apposition, and anaphora. The main idea is that all terms within a parallel structure have the same *syntactical* features within the sentence (e.g. object or subject). They are used in combination with the same verb, preposition, etc., and, thus, we hypothesise that they exhibit similar functional characteristics. This measure has high precision, but low recall.

We further introduce *contextual similarity*, where frequently used context patterns in which terms appear are used for comparison. These patterns are domain-specific, but are learnt automatically from a corpus by pattern mining. Context patterns consist of the syntactical categories and additional lexical information, and are used to identify functionally similar terms.

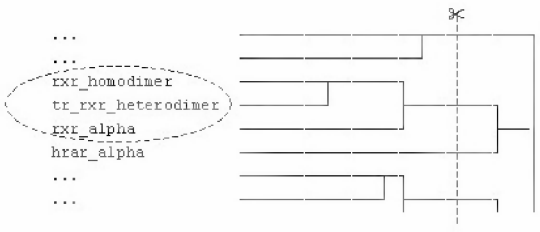

Figure 1: Producing clusters by cutting off the subtrees

The CLS similarity combines the three similarity measures, where the parameters of such combination are learnt automatically by training this measure on an ontology by using distances between terms as an indicator of their similarity (Spasic et al., 2002). This measure is fed into a hierarchical clustering algorithm. It produces a hierarchy of nested clusters, and the

final set of clusters is produced by cutting off the hierarchy at a certain level (see Figure 1). The approach achieves around 71% precision, where the precision has been calculated as the number of correctly clustered terms.

4.4 Encoding terminology results

The results of the terminological processing are encoded in XML together with the text itself. Namely, ATRACT marks all occurrences of terms in the body of a text and links term variants. It then stores terminological information in a separate section at the end of a document, which provides information on all normalised terms and specifies term clusters.

```
<TITLE>Glucocorticoid hormone resistance during
primate evolution: receptor-mediated mechanisms.
</TITLE>
<ABSTRACT> ...
This was confirmed by showing that the hypothalamic-
<TERM id=3 sem=010010>pituitary adrenal axis </TERM>
is resistant to suppression by dexamethasone. To study this
phenomenon, <TERM id=1 sem=10010> glucocorticoid
receptors </TERM> were examined in circulating
<TERM id=4 sem=101010> mononuclear leukocytes</TERM>
and cultured <TERM id=5 sem=101011>skin fibroblasts
</TERM> . . .
</ABSTRACT>
<TERMINOLOGY>
 . . .
<TERM id=1 sem=10010  nf="glucocorticoid receptor"/>
 . . .
<TERM id=4 sem=101010 nf="mononuclear leukocyte"/>
<TERM id=5 sem=101011 nf="skin fibroblast"/>
. . .
</TERMINOLOGY>
```

Figure 2: XML document produced by ATRACT

Figure 2 depicts the results of the terminology processing. Each *TERM* tag in the body of a text has an *id* attribute, which refers to a normalised term associated with that specific occurrence. Variants of the same term are, thus, linked via the *id* attribute. The list of all terms that are recognised is stored at the end of a document, together with all terminological information that has been collected. In this list, the *sem* attribute indicates term clusters, while *nf* refers to a normalised form of a term.

5 Querying literature corpus

Knowledge mining and conceptual information extraction in X-TRACT are supported by XML-tag management. In order to extract information, users define queries that describe relationships between terms and their contexts. Query are defined via GUI, and are translated into the XQuery language.

XQuery,[2] an XML query language, is used as an underlying query language for the GUI implemented as a part of X-TRACT. The main reason for defining a specific GUI is that the syntax of XQuery might be too complex for domain experts. There are two possible approaches to this problem. One approach is to create a scripting language on top of XQuery simplifying the most common queries. Since it is still not suitable for end users of such applications, we adopted another approach in which an interface GUI layer is used for the formulation of queries.

XQuery is a functional language and is strongly typed, i.e. all the operands used in expressions and functions must conform to their designated static types. The main building blocks of XQuery are expressions. An expression may consist of a value, function or another expression. There are several built-in operators to help build queries (logical, type casting, arithmetic, set operations, and the FLWR (for, let, where, return) expression).

An X-TRACT query is an XQuery expression that combines any linguistic (namely, POS and syntactic) and domain-specific (namely, *TERM* tags) XML-tags. Attributes of XML-tags can also be used to make queries more restricted by referring to either values of attributes (e.g. *nf="receptor"*) or their characteristics (e.g. value of the *nf* attribute starting with *'nuclear'*). Also, in the case of the *TERM* tag, all term variants are considered by default while generating query's output.

In order to define tag operations that are available via GUI, domain experts have been interviewed in order to identify the most important query types they are interested in.

[2] More information on XQuery is available at www.w3.org/TR/xquery/.

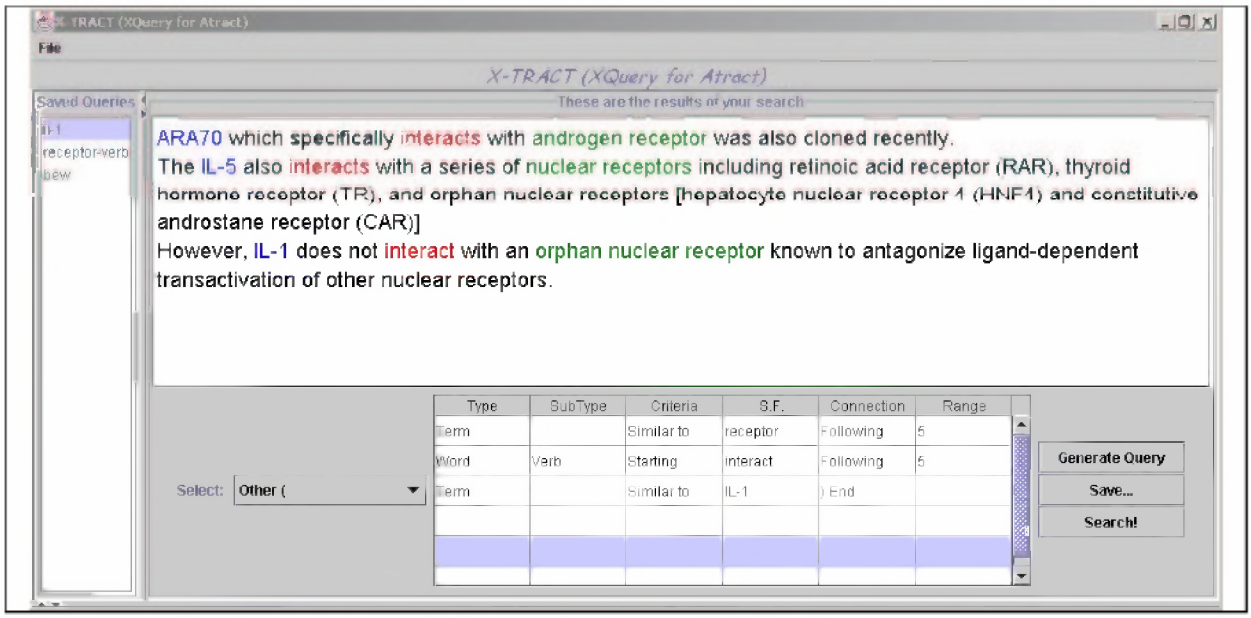

Figure 3: Querying in X-TRACT

Consequently, we defined the following unary tag operations:

- *similar(TERM)*, which denotes a set of terms belonging to the same cluster as *TERM*;
- *following(TAG)*, which denotes an entity which follows (not necessarily immediately) the given *TAG*;
- *preceding(TAG)*, which denotes an entity which preceedes (not necessarily immediately) the given *TAG*, and
- *range(TAG, m, n)*, which denotes an entity which appears in a window of *m* words left and *n* words right of the given *TAG*.

The tag operations (apart from *similar*) are applied to sentences, and the ones that match the query criteria are selected for the output.

A query is constructed via the Query Composer (QC). The QC presents a user with a table, where each row specifies a tag and its attributes. Rows are combined via Boolean or range operators. After the user completes his/her query, the QC translates it to the XQuery equivalent, which is passed on to the XML-DB management system.

Figure 3 depicts an example of the formulation of a query that approximates the following IE task: "*which entities similar to 'receptor' interact with entities similar to 'IL-1'?*". This query extracts all sentences that have terms similar to *'receptor'* followed by the verb *'interact'*, which is further followed by a term similar to *'IL-1'*. The results are presented in a window with matching elements highlighted. As we can note, the results also include 'negative' examples (see the last sentence in Figure 3: for *'not interact'*), which may be beneficial in the knowledge mining process.

6 Discussion

XML has been already widely used by the NLP community as a format suitable for data-exchange and document processing. There are many reasons behind this choice, portability and self-description being the most important ones. An XML document has a concise, well-defined, hierarchical structure, separating pieces of data into identifiable elements each having a precise meaning.

The main advantage of XML representation is that it can represent nested structures, something not easily done in RDBs. However, even when XML is used to encode documents, many applications still use RDBs for storage and manipulation. In order to store an XML document in a RDB, all tags need to be removed and stored in a separate table together with their starting and ending position in the plain text and their attributes (Nenadic et al., 2002). More importantly, the hierarchical structure of a document may be lost if all tags are stored at the

same level (i.e. in flat tables). Theoretically the structure can be retained, but in order to do so a new table has to be created for each element type that can contain other elements. However, this can dramatically increase the number of tables required. These problems are avoided if an XML-native DB is used for the storage of XML documents, as they naturally store hierarchy of tags.

RDBs are generally considered more efficient when it comes to retrieving specific types of elements. On the other hand, XML-native DBs provide extended querying facilities given by a native query language (e.g. XQuery).

Although the use of a GUI to drive a user when formulating a query has obvious benefits, it is impossible to retain complete expressiveness of a query language. For this reason, there is an option in X-TRACT to formulate queries using the syntax of XQuery directly.

7 Conclusion

In this paper we presented X-TRACT, a terminology-driven literature corpus mining system. The main aim is to aid domain specialists in systematic location and extraction of the new knowledge from scientific literature corpora. X-TRACT integrates ATR, term variant recognition, acronym acquisition and term clustering.

Before querying, a corpus is subjected to automatic terminological analysis and the results are annotated in XML. All term occurrences including their variants are linked, and XML documents are stored in an XML-native database. All corpus retrieval operations are performed against this database using an XML query language. IE within the system is terminology-driven and based on tag operations.

The preliminary experiments show that this approach offers improved user satisfaction while mining literature corpora. Important areas of future research will involve integration of a manually curated ontology with the results of automatically performed term clustering. Further, we will investigate the possibility of using an automatic term classification system as an alternative structuring model for knowledge deduction and inference (instead of clustering).

References

Baker P.G., Brass A., Bechhofer S., Goble C., Paton N. and Stevens R. 1998. *TAMBIS: transparent access to multiple bioinformatics information sources - an overview.* In Proc. of 6th International Conference on Intelligent Systems for Molecular Biology - ISMB98, Montreal, Canada, pp. 25-34.

Chang J.T., Schutze H. and Altman R.B. 2002. *Creating an online dictionary of abbrevations from Medline.* JAMIA, to appear.

Christ O. 1994. *A modular and flexible architecture for an integrated corpus query system.* In Proceedings of COMPLEX'94, Budapest, Hungary.

Grishman R. 1995. *TIPSTER phase II architecture design document.* New York University, available at http://www.tipster.org/arch.htm.

Jacquemin C. 2001. *Spotting and discovering terms through NLP.* MIT Press, Cambridge MA, 378 p.

Krauthammer M., Rzhetsky A., Morozov P. and Friedman C. 2000. *Using BLAST for identifying gene and protein names in journal articles.* Gene, 259, pp. 245-252.

Frantzi K.T., Ananiadou S. and Mima H. 2000. *Automatic recognition of multi-word terms: the C-value/NC-value method.* Int. J. on Digital Libraries, 3/2, pp. 115-130.

Mima H., Ananiadou S. and Nenadic G. 2001. *ATRACT workbench: an automatic term recognition and clustering of terms.* In "Text, Speech and Dialogue", V. Matoušek, P. Mautner, R. Mouček & K. Taušer, ed., LNAI 2166, Springer Verlag, pp. 126-133.

Nenadic G., Mima H., Spasic I., Ananiadou S. and Tsujii J. 2002. *Terminology-driven literature mining and knowledge acquisition in biomedicine.* International Journal of Medical Informatics, pp. 1-16.

Nenadic G., Spasic I. and Ananiadou S. 2002. *Automatic acronym acquisition and term variation management within domain specific texts.* In Proc. of LREC 2002, Las Palmas, Spain, pp. 2155-2162.

Pustejovsky J., Castaño J., Zhang J., Kotecki M. and Cochran B. 2002. *Robust relational parsing over biomedical literature: extracting inhibit relations.* In Proc. of PSB-2002, Hawaii, pp. 7:362-373.

Spasic I., Nenadic G., Manios K. and Ananiadou S. 2002. *Supervised learning of term similarities.* In "Intelligent Data Engineering and Automated Learning", H. Yin, N. Allinson, R. Freeman, J. Keane & S. Hubbard, ed., LNCS 2412, Springer Verlag, pp. 429-434.