

# Terminology-driven literature mining and knowledge acquisition in biomedicine

Goran Nenadić<sup>a</sup>, Hideki Mima<sup>b,\*</sup>, Irena Spasić<sup>a</sup>, Sophia Ananiadou<sup>a</sup>,  
Jun-ichi Tsujii<sup>c</sup>

<sup>a</sup> *Computer Science Department, University of Salford, Salford, UK*

<sup>b</sup> *Department of Engineering, University of Tokyo, 7-3-1 Hongo Bunkyo-ku, Tokyo 113 8654, Japan*

<sup>c</sup> *Department of Information Science, University of Tokyo, Tokyo, Japan*

---

## Abstract

In this paper we describe Tagged Information Management System (TIMS), an integrated knowledge management system for the domain of molecular biology and biomedicine, in which terminology-driven literature mining, knowledge acquisition (KA), knowledge integration (KI), and XML-based knowledge retrieval are combined using tag information management and ontology inference. The system integrates automatic terminology acquisition, term variation management, hierarchical term clustering, tag-based information extraction (IE), and ontology-based query expansion. TIMS supports introducing and combining different types of tags (linguistic and domain-specific, manual and automatic). Tag-based interval operations and a query language are introduced in order to facilitate KA and retrieval from XML documents. Through KA examples, we illustrate the way in which literature mining techniques can be utilised for knowledge discovery from documents.

© 2002 Elsevier Science Ireland Ltd. All rights reserved.

*Keywords:* Terminology management; Literature mining; Information extraction; Knowledge acquisition; XML annotation

---

## 1. Introduction

New discoveries within biomedicine result in an abundance of scientific papers verbalising these discoveries. These documents are

created in an attempt to share new knowledge with other scientists. They are often reproduced in electronic form and placed on the Internet or other types of shared resources in order to make the new information widely and easily available. 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 knowledge media.

---

\* Corresponding author

E-mail address: [mima@biz-model.t.u-tokyo.ac.jp](mailto:mima@biz-model.t.u-tokyo.ac.jp) (H. Mima).

The sheer amount of published papers<sup>1</sup> 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 (KSs) emphasises the importance of developing flexible and efficient tools for automatic knowledge acquisition (KA) and integration. Different text and literature mining techniques [1–6] have been developed recently in order to facilitate efficient discovery of knowledge contained in large textual collections. The main goal of literature mining is to retrieve knowledge that is ‘buried’ in a text and to present the distilled knowledge to users in a concise form. Its advantage, compared with ‘manual’ knowledge discovery, is based on the assumption that automatic methods are able to process an enormous amount of texts. It is doubtful that any researcher could process such huge amount of information, especially if the knowledge spans across domains. For these reasons, literature mining aims at helping scientists in collecting, maintaining, interpreting and curating information.

One of the main problems when processing a collection of KSs is their heterogeneity and dynamic nature. Even when confined to a single domain, the KSs are autonomously developed and maintained by independent organisations for different purposes, hence resulting in a heterogeneous set of KSs. Moreover, this set is dynamic as a result of continuous attempts to synchronise its content with up-to-date knowledge. New infor-

mation is being added and existing information is revised and often removed from the KSs. These two facts, heterogeneity and constant evolution of KSs, set a challenge to systems designed to assist users in locating and integrating knowledge relevant to their needs.

In this paper we introduce Tagged Information Management System (TIMS), an integrated literature mining system designed for the domain of molecular biology and biomedicine, where terminology-driven KA, knowledge integration (KI), and XML-based information extraction (IE) are combined using tag-based information management and ontology inference. TIMS incorporates a terminology management workbench and a query language with the corresponding procedures that allow users to formulate and execute complex queries against a collection of XML documents.

The paper is organised as follows: in [Section 2](#) we describe the related work. TIMS is overviewed in [Section 3](#). Terminology processing and KA techniques are presented in [Sections 4 and 5](#), respectively. Finally, [Section 6](#) provides details of the experiments and their results.

## 2. Related work

### 2.1. Terminology management

Knowledge encoded in textual documents is organised around sets of specialised *terms* (e.g. in biomedical domain, terms represent names of proteins, genes, acids, etc.). Hence, KA relies heavily on the recognition of terms. Obviously, a scheme to integrate terminology management as a key prerequisite for KA and KI is needed.

One of the main problems that makes automatic terminology recognition (ATR)

---

<sup>1</sup> For example, the MEDLINE database [7] currently contains over 12 million abstracts in the domains of molecular biology, biomedicine and medicine, growing by more than 40 000 abstracts each month.

difficult is the lack of clear naming conventions, although some attempts in this direction are being made. For instance, naming conventions do exist for some types of biomedical concepts, e.g. genes, alleles and proteins [8]. There are formal bodies, such as Enzyme Commission and Human Gene Nomenclature Committee, whose responsibilities involve assigning unique symbols and descriptive names to specific types of concepts. Still, these are only guidelines and as such do not impose restrictions to domain experts. In addition, they apply only to a subset of terms, while the rest of biomedical terminology remains highly non-standardised.

There are several approaches to ATR in biomedicine and molecular biology. Some of them rely mainly on linguistic information, namely on morpho-syntactic features of domain terms. For instance, LaSIE [9], an adapted newswire name recogniser, uses a case-sensitive terminology lexicon of component terms, set of morphological cues (biochemical suffixes) and hand-constructed grammar rules in order to recognise terms belonging to specific terminological classes (e.g. enzymes, proteins, etc.). Another example of a rule-based system is PROPER [10], which uses ‘core’ and ‘feature’ terms to identify strings that correspond to proteins. ‘Core’ terms are domain-characteristic words (containing capitals, numerals etc.) and ‘feature’ terms are keywords that describe function and characteristic of a term (e.g. protein, receptor, etc.). Recently, hybrid approaches combining linguistic and statistical knowledge are increasingly used [11–13]. In order to assess the relevance of extracted term candidates, such methods calculate weights (i.e. termhoods) according to specific statistical measures. Machine learning techniques can be applied as well: for example, [14] presents a

statistically based, unsupervised technique to acquire and disambiguate names of proteins, genes, and RNAs.

However, ATR is not the ultimate goal itself. The large number of new terms calls for a systematic way of accessing and retrieving the knowledge represented by them. Accordingly, the extracted terms need to be placed in an appropriate knowledge framework by discovering relations between them, and by establishing links between the terms and different factual databases.

In order to implement terminology-based knowledge structuring, several ontologies have been developed (e.g. MeSH terms, Gene Ontology, Genia ontology, etc.). Each of them provides a top-down controlled framework, which aims to organise and describe the terminology in the domain. Ontologies implement a pre-defined classification system for terms and their relationships, as well as inference rules that are used to derive knowledge represented by them. However, ontology construction and maintenance are time-consuming activities, as terms are usually manually integrated into an ontology. This is one of the reasons why ontologies typically contain just a subset of existing terminology. In addition, no solution to the well-known difficulties in manual ontology development, such as ontology conflicts/mismatches [15], is provided. Therefore, techniques for automated ontology management [16] are required for efficient and consistent KA and KI.

## 2.2. *Integration of knowledge sources*

Different approaches to linking, integrating and interpreting relevant resources have also been suggested. For example, the Semantic Web framework [17] strives to link relevant XML-based resources in a bottom-up man-

ner using the Resource Description Framework (RDF) and ontology information. Since XML allows introduction of new domain- and/or application-specific tags, RDF [18] is used to define their ‘meanings’ and relationships to one another, while the corresponding ontology is used to combine and derive additional information (e.g. synonyms, hyponyms, etc.). In this sense, ontologies are used as a key domain knowledge repository. However, though the Semantic Web framework is powerful when it comes to expressing the content of resources to be semantically retrieved, manual description is needed when defining RDF descriptions and ontologies. If we, however, endeavour to process huge collections of new documents (which cover new knowledge), we need systems that do not rely solely on manual descriptions.

Additional efforts have been made to design an appropriate interface for accessing multiple KSs. For example, the TAMBIS system [19] tried to provide integrated and transparent access to different resources using the classic mediator/wrapper architecture and homogenising layer on top of different sources. Supported by a knowledge-driven user interface, queries about protein structure and nucleic acid coding signals are expressed in a description logic language. The queries are based on a universal biological conceptual model (represented by a terminology), which is encoded in the same description language, thus facilitating query execution.

Several other systems have been proposed for integrating knowledge acquisition and management. For example, GENIES [20] uses a semantic grammar and syntactic constraints in order to extract comprehensive information about signal-transduction pathways from journal texts.

In this paper, we present our approach to terminology management and integration of KSs.

### 3. An overview of TIMS

The TIMS system has been developed with the intention to address the problems of the terminology-driven literature mining and KA. Similarly to the Semantic Web framework, TIMS deals with XML documents by using domain-specific RDF descriptions and ontology-based inference. However, TIMS facilitates KA and KI tasks not only by using manually defined resource descriptions, but also by exploiting natural language processing techniques such as ATR and automatic term clustering (ATC), which are used for automatic population of the underlying ontology. Additionally, TIMS implements a query language and the corresponding procedures that allow users to formulate and execute complex queries against a collection of XML documents.

TIMS acts as an IE engine, which is based on managing XML tag information obtained from its subfunctional components. Typically, IE-based KA process within TIMS has the following course: first, a collection of documents is linguistically processed (part-of-speech (POS) tagging, shallow parsing, etc.). Further, the collection is terminologically analysed, i.e. relevant domain-specific terms are automatically recognised and structured (classified or incorporated into an ontology). In the next step, a user (typically an expert in biomedicine) formulates a query, which is executed against the collection. The results are presented in a table (containing resulting entities) and the relevant text is highlighted. Fig. 1 illustrates this process.

The TIMS architecture is modular, and it integrates the following components (Fig. 2):

- Document retriever—retrieves documents from KSs and performs basic linguistic processing (i.e. POS tagging).

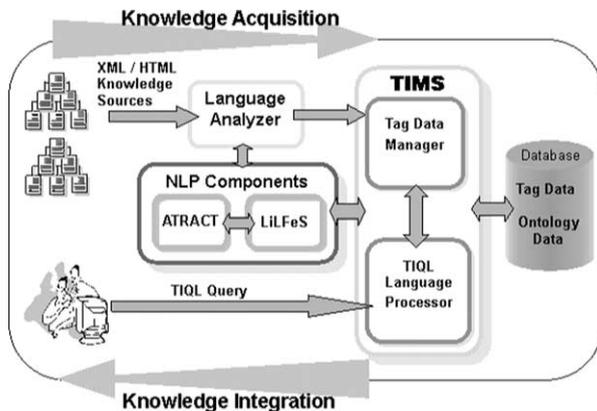


Fig. 1. IE-based KA process within TIMS.

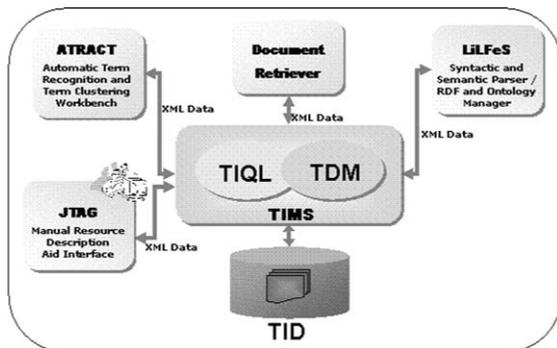


Fig. 2. The TIMS system architecture.

- LiLFeS—an abstract machine that performs shallow parsing and ontology inference.
- Automatic Term Recognition and Clustering of Terms (ATRACT)—a workbench that carries out the recognition and structuring of domain terminology.
- JTAG—facilitates manual resource description, namely definition, modification and adjustment of tags.
- Tag Data Manager (TDM)—stores tag information in a tag information database (TID) and provides the corresponding interface.
- Tag Information Query Language (TIQL) query processor—executes user/system

queries in order to extract XML-tagged information by using TID and a terminology.

Linguistic pre-processing within TIMS is performed in two-steps. In the first step, POS tagging<sup>2</sup>, i.e. the assignment of basic parts of speech (e.g. noun, verb, etc.) to words, is performed. In the second step, LiLFeS [22] is used to perform parsing, i.e. the recognition of basic syntactic structures (e.g. noun phrases, verb phrases, etc.). The parser is based on a head-driven phrase structure grammar for English (LinGO grammar [23]), which is implemented as a definite clause program with typed feature structures.

ATRACT and LiLFeS play a central role in term recognition and ontology inference, while TDM and the TIQL processor are TIMS kernel modules which implement the IE engine. The following sections elaborate the functionality of the main TIMS modules.

#### 4. Terminological processing in TIMS

The lack of clear naming standards in biomedicine makes ATR a non-trivial problem [24]. Also, it typically gives rise to many-to-many relationships between terms and concepts. In practice, two problems stem from this fact: the same term may denote a number of concepts, and, conversely, the same concept may be denoted by more than one term. In other words, there are terms that have multiple meanings (*term ambiguity*), and, conversely, there are terms that refer to the same concept (*term variation*). Generally, term ambiguity has negative effects on IE precision, while term variation decreases IE recall.

<sup>2</sup> TIMS uses the EngCG tagger [21].

These problems point out the impropriety of using simple keyword-based IE techniques. Obviously, more sophisticated techniques are needed. Such techniques should identify groups of different terms referring to the same (or similar) concept(s), and, therefore, could benefit from relying on efficient and consistent ATR/ATC and term variation management methods. These methods are also important for organising domain-specific knowledge, as terms should not be treated isolated from other terms. They should rather be related to one another so that the relations existing between the corresponding concepts are at least partly reflected in a terminology.

Terminological processing in TIMS is carried out by ATRACT [25], a terminology management workbench that integrates ATR and ATC (Fig. 3). Its main purpose is to help biologists in gathering and managing domain-specific terminology. It is used to automatically retrieve and cluster terms on the fly and pass the XML-tagged results.

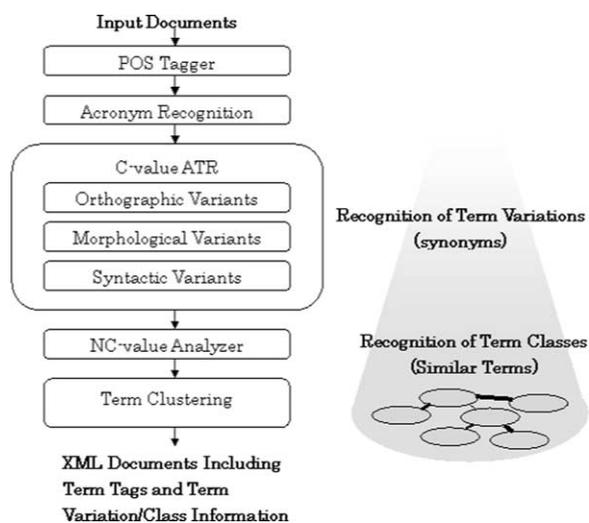


Fig. 3. Terminology processing in TIMS.

#### 4.1. Term recognition

The ATR method used in ATRACT is based on the *C*- and *NC*-value methods [11]. The *C*-value method recognises terms by combining linguistic knowledge and statistical analysis. The method extracts multi-word terms<sup>3</sup> and is not limited to a specific class of concepts. It is implemented as a two-step procedure. In the first step, term candidates are extracted by 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 characteristics 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 appear 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 (see Section 6) has shown that contextual information improves term distribution in the ex-

<sup>3</sup> More than 85% of domain-specific terms are multi-word terms [12].

tracted list by placing real terms closer to the top of the list.

#### 4.2. Term variation management

Term variation and ambiguity are causing problems not only for ATR but for human experts as well. Several methods for term variation management have been developed. For example, the BLAST system [26] used approximate text string matching techniques and dictionaries to recognise spelling variations in gene and protein names. FASTR [27] handles morphological and syntactic variations by means of meta-rules used to describe term normalisation, while semantic variants are handled via WordNet.

The basic *C*-value method has been enhanced by term variation management [24]. We consider a variety of sources from which term variation problems originate. In particular, we deal with orthographical, morphological, syntactic, lexico-semantic and pragmatic phenomena. Our approach to term variation management is based on term normalisation as an integral part of the ATR process. Term variants (i.e. synonymous terms) are dealt with in the initial phase of ATR when term candidates are singled out, as opposed to other approaches (e.g. FASTR handles variants subsequently by applying transformation rules to extracted terms). Each term variant is normalised (see Table 1

for an example), 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. This way, a list of normalised term candidate classes, rather than a list of single terms is statistically processed. The termhood is then calculated for a whole class of term variants, not for each term variant separately.

Further, we have paid special attention to acronyms as a common way of introducing term variants in the domain. A method for the automatic acquisition of newly introduced acronyms and the mapping to their expanded forms has been developed [24]. The acronym acquisition is a part of the ATR process: acronyms are acquired in the first step, and each acronym occurrence is replaced with the corresponding expanded form prior to the *C*-value statistical analysis. This way, all term occurrences (including acronyms) are considered for calculation of termhoods.

#### 4.3. Term clustering

Beside term recognition, term clustering is an indispensable component of the literature mining process. Since terminological opacity and polysemy are very common in molecular biology and biomedicine, term clustering is essential for the semantic integration of terms, the construction of domain ontologies and semantic tagging.

ATC in ATRACT is performed using a hierarchical clustering method in which clusters are merged based on average mutual information measuring how strongly terms are related to one another [28]. Terms automatically recognised by the *NC*-value method and their co-occurrences are used as input, and a dendrogram of terms is produced as output. Parallel symmetric processing is used for high-speed clustering. The calculated term

Table 1  
Term normalisation example

Term variants	Normalised term
Human cancers	Human cancer
Cancer in humans	Human cancer
Human's cancer	Human cancer
Human carcinoma	Human cancer

```

<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>

```

Fig. 4. XML document produced by ATRACT.

cluster information is encoded using the LiLFeS syntax (see Section 5.3).

#### 4.4. Manual terminology tuning

The ATR/ATC results automatically obtained by ATRACT are encoded in XML (Fig. 4). However, the results can be manually adjusted by the JTAG module, which provides a GUI for managing terminological information. A user can alter, delete or add new tags in the text, and amend the ATR/ATC results by hand (Fig. 5).

### 5. Knowledge acquisition and information extraction in TIMS

Literature mining can be regarded as a broader approach to IE. Classic IE tasks relate to identification of entities and relations among them in a collection of documents, as well as extraction of domain-specific events. The goal is to fill in the pre-defined templates, which describe certain event of interest [29]. However, in literature mining the templates

can be either discovered by a system or defined by a user.

KA and IE in TIMS are implemented through the integration of XML-tag management (Section 5.1) and tag- and ontology-based IE (Section 5.4). In order to extract information, a user defines a TIQL query that describes a template or pre-defined database schema, specifying the output format. In the following subsection we describe the TIMS modules related to tag management and IE.

#### 5.1. XML-tag management

In an attempt to accumulate a large amount of meta-information about documents, several types of tags are attached to text in different steps of document processing. The same document in TIMS may have multiple tag information including POS (e.g. noun, adjective, etc.), syntactic (e.g. noun phrase, verb phrase, etc.) and semantic (e.g. protein, DNA, term, etc.) tags. These tags are typically interlaced (e.g. a noun phrase that is also a DNA), which may reduce the visual transparency of a document. Furthermore, in the case where the tagging scheme includes structural complexity such as nesting and possible combinations of syntactic and semantic structures, the difficulty of processing the tags reduces the efficiency of tag-based KA.

Our approach to interlaced and multi-layered tag information follows the TIPSTER architecture [30]. TDM module parses XML documents and stores each type of tag information in TID separately from the original documents. Tag information (namely tag type and the corresponding text positions) are stored and managed by an external database software<sup>4</sup> (Fig. 6). This approach facilitates

<sup>4</sup> The current implementation uses either JDBC/ODBC or Berkeley DB.

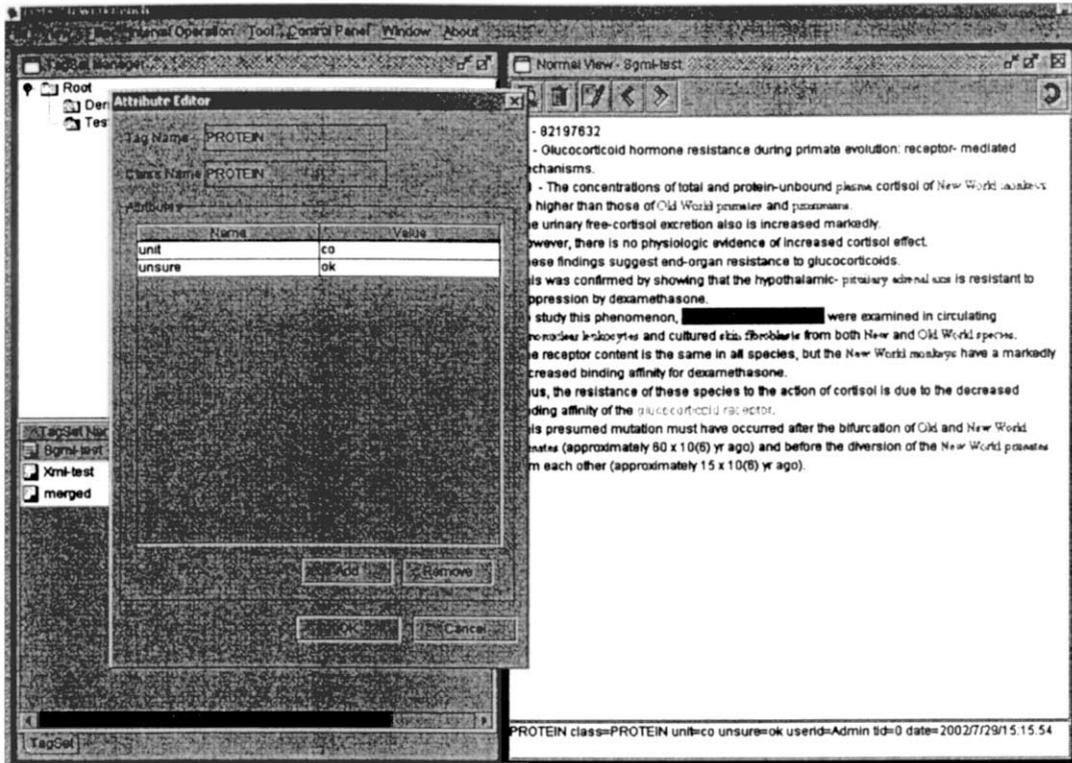


Fig. 5. Manual terminology tuning using JTAG.

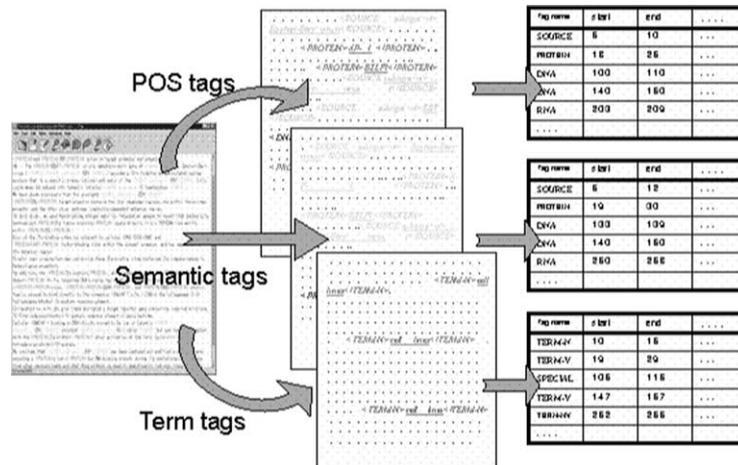


Fig. 6. Storing multi-layered tag information in TID.

efficient combination of different tags, as well as easy import and integration of different tags for the same document.

## 5.2. Tag interval operations

The key feature of KA and KI within TIMS is an ability to logically retrieve and combine data that is annotated by different types of tags. This is done by manipulating intervals of text. The main assumption is that XML tags indicate certain *intervals* in text. A text interval is specified as a text window surrounding an XML-tagged lexical item used as an anchor<sup>5</sup>. In the simplest case, a text interval refers to a chunk of text between a pair of corresponding XML tags. An interval is denoted by specifying a tag name and, optionally, additional features. For example,  $\langle \text{TERM nf} = \text{'COUP-TF II'} \rangle$  refers to text intervals inside the  $\langle \text{TERM} \rangle$  tags, which have the value of nf-feature set to 'COUP-TF II'<sup>6</sup>.

Interval operations are XML specific text/data retrieval operations, which are used to manipulate such text intervals. Interval operations are binary operations whose operands are sets of text intervals. These sets are mapped to a set of intervals according to a specified operation. Currently, four types of interval operations are defined.

Let  $I$  be a set of all text intervals in a specific document, and let  $A$  and  $B$  be its subsets. Then the four operations may be defined as follows.

<sup>5</sup> Note that text intervals defined this way are always non-empty.

<sup>6</sup> In this paper we assume that  $\langle \text{TERM} \rangle$  tag has nf (normalised form) and cat (semantic category) attributes. Similarly, tags corresponding to syntactic categories may have a lemma attribute, which describes the canonical form of a word.

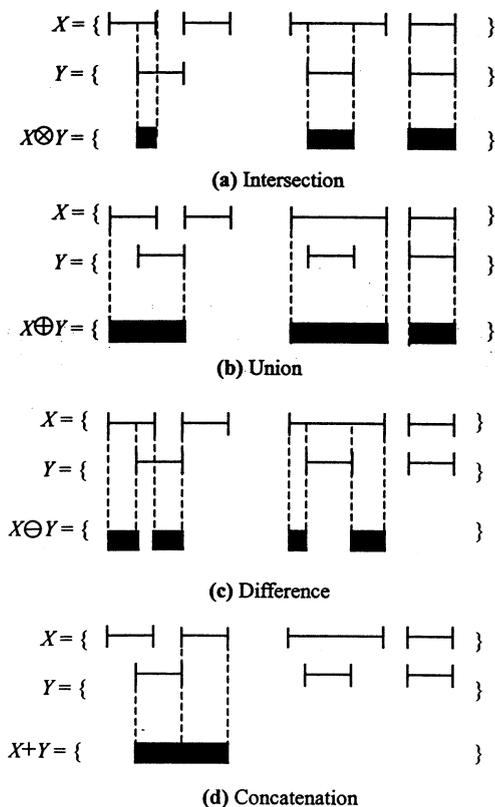


Fig. 7. Interval operations.

*Intersection* returns a set of intervals corresponding to non-empty intersections of all intervals from the two respective sets (Fig. 7a):

$$A \otimes B = \{i \in I \mid (\exists i_a \in A)(\exists i_b \in B) i = i_a \cap i_b\}$$

*Union* merges those intervals from the two respective sets that have non-empty intersection (Fig. 7b):

$$A \oplus B = \{i \in I \mid (\exists i_a \in A)(\exists i_b \in B) i = i_a \cup i_b\}$$

*Difference* returns disagreements between the intervals from the two respective sets that have non-empty intersection (Fig. 7c):

$$A \ominus B = \{i \in I \mid (\exists i_a \in A)(\exists i_b \in B) (i \subseteq i_a \Delta i_b \wedge \neg (\exists j \in I)(j \subseteq i_a \Delta i_b \wedge i \subset j))\}$$

*Concatenation* merges<sup>7</sup> the neighbouring intervals from the two respective sets that have empty intersection (Fig. 7d):

$$A + B = \{i \in I | (\exists i_a \in A)(\exists i_b \in B) \\ (i = i_a \cup i_b \wedge i_a \cap i_b = \emptyset)\}$$

We provide a couple of examples to explain the usage of these operations. For instance, the interval expression  $\langle \text{VP} \rangle \otimes (\langle \text{V} \rangle + \langle \text{TERM} \rangle)$  describes all verb-term pairs (regardless of the order) within a verb phrase<sup>8</sup>. Similarly,  $(\langle \text{SENTENCE} \rangle \oplus \langle \text{TERM nf} = \text{'COUP TF II'} \rangle) \oplus \langle \text{V lemma} = \text{'inhibit'} \rangle$  refers to all sentences that contain information on inhibition processes possibly involving transcription factor COUP-TF II. Here we combine linguistic ( $\langle \text{SENTENCE} \rangle$  and  $\langle \text{V} \rangle$ ) with domain-specific tags ( $\langle \text{TERM} \rangle$ ) in order to identify more specialised information.

### 5.3. Ontology-based inference

Knowledge management involves KA, its organisation, structuring, refinement, and distribution to domain specialists [31], which can be addressed by means of ontologies. An ontology is an explicit conceptualisation realised through a set of concepts, their definitions and relations between them [32]. Ontologies are mainly used to facilitate knowledge distribution, that is—to provide effective means of communication within a domain (between both humans and computer systems).

Ontology-based techniques for accessing information of interest allow users to perform a sophisticated search, which enables access to implicitly stated relevant information by a

hierarchical query expansion. Queries are expanded by using inference rules, so that a search term matches all hierarchically subordinated terms.

Ontology inference used in TIMS is based either on an existing ontology (e.g. the Genia ontology [33]) or on a dendrogram automatically constructed by ATTRACT<sup>9</sup>. In either case, the ontology is encoded by using the LiLFeS syntax, while LiLFeS deductive capabilities [22] are used to implement hierarchical and deductive matching. Such matching enables the specification of a whole hierarchy of terms instead of a single term, and this hierarchy is used to expand queries. For example, the expression  $\langle \text{TERM cat} = \text{nucleic\_acid} \rangle$ <sup>10</sup> refers to all terms belonging to the class of nucleic acids. This ability further enhances the functionality of interval operations. For instance, expression  $\langle \text{VP} \rangle \otimes \langle \text{nucleic\_acid*} \rangle$  describes all terms from the class of nucleic acids (including its subclasses) that appear within a verb phrase.

### 5.4. Tag- and ontology-based IE

Two types of IE approaches are combined in TIMS: tag-based pattern matching and ontology-based hierarchical matching. TIQL is used as an interface for both approaches. Using this language, a user is able to specify interval operations to be performed on selected documents. Such queries can be automatically expanded through ontology inference. The basic queries in TIQL have the following SQL-like format:

<sup>7</sup> Note that concatenation between sets of intervals defined this way is a commutative operation, unlike concatenation defined for strings.

<sup>8</sup>  $\langle \text{VP} \rangle$  is a tag for a verb phrase, while  $\langle \text{V} \rangle$  denotes a verb.

<sup>9</sup> Our future work will include a possibility of combining a predefined ontology with an automatically obtained dendrogram.

<sup>10</sup> An equivalent TIQL notation is  $\langle \text{nucleic\_acid*} \rangle$ , with  $**$  denoting hierarchical matching.

```

SELECT      <variables>
FROM        <documents>
WHERE       <expression>
FROM        <documents>
WHERE       <expression>
...

```

where, <variables> specifies an output table format, <documents> is a list of XML documents to be processed, and <expression> is a combination of interval operations to be performed on documents from the corresponding WHERE clause. For example, the following expression:

```

SELECT  X, Y
FROM    'paper-1.XML', 'paper-2.XML'
WHERE   <VP>⊗(X:<EVENT*>
            +Y:<nucleic_acid*>)

```

extracts all combinations of terms X and Y that are part of a verb phrase (<VP>), where X matches any term from the <EVENT> class (and all its subclasses) and Y matches any term from the <nucleic\_acid> class (and all its subclasses; Fig. 8).

Similarly, information about all nuclear receptors that possibly interact with DAX-1 may be extracted from a text by executing the following query:

```

SELECT  X
FROM    'Medline2082.XML'
WHERE   <VP>⊗((X: <nuclear_receptor*>
            +<V lemma = 'interact'>)
            +<DAX-1>)

```

As the formulation of appropriate interval operations may be cumbersome especially for novice users, TIMS allows a possibility of 'recycling' typical queries and macros.

## 6. Experiments and evaluation

In this section we briefly present the quality of the ATR/ATC results and, after that, we

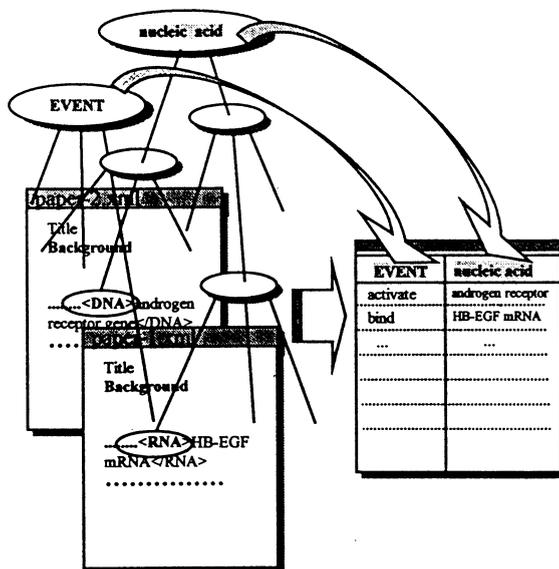


Fig. 8. Tag- and ontology-based IE.

discuss the practical performance of tag manipulation in TIMS.

The experiments in ATR with the term variation management were conducted on a corpus containing 2082 abstracts from the Medline database [7]. A sample of automatically recognised terms and their variants is presented in Table 2. Fig. 9 shows the interval precision of C- and NC-value methods compared with the frequency of occurrence. As one can see, the NC-value method increases slightly the precision compared with that of the C-value method. The recall and precision of the methods are given in Fig. 10<sup>11</sup>.

Additional experiments with acronym acquisition have been conducted on two corpora containing 2082 and 6323 Medline abstracts, and a random sample of 50 abstracts taken

<sup>11</sup> For detailed evaluation the reader is advised to see [24,34].

Table 2  
Sample of recognised terms

Terms (and term variants)	Termhood
<i>Retinoic acid receptor</i> Retinoic acid receptor Retinoic acid receptors RAR, RARs	6.33
<i>Nuclear receptor</i> Nuclear receptor Nuclear receptors NR, NRs	6.00
<i>All-trans retinoic acid</i> All-trans-retinoic acid All-trans-retinoic acids ATRA, at-RA, atRA	4.75
<i>9-cis-Retinoic acid</i> 9-cis-Retinoic acid 9cRA, 9-c-RA	4.25

Table 3  
Sample of acronyms acquired

Acronym(s)	Expanded form(s)
RAR alpha RAR-alpha RARA RARa	Retinoic acid receptor alpha
RARs RAR RT-PCR	Retinoic acid receptor Retinoic acid receptors Reverse transcription PCR
TR	Thyroid hormone receptor Thyroid hormone receptors
TRs	Thyroid receptor
9-c-RA	9-cis-retinoic acid
9cRA	9-cis retinoic acid
ES	Ewing sarcoma Ewing's sarcoma Ewings sarcoma

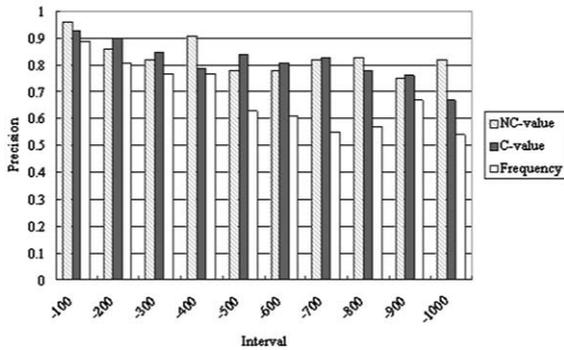


Fig. 9. Interval precision of C/NC-value methods.

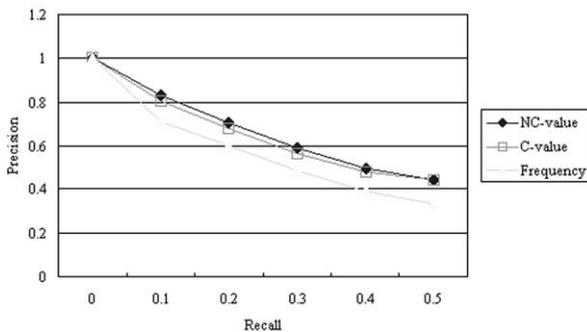


Fig. 10. Precision and recall of C/NC-value methods.

from the first corpus has been used for the calculation of recall. Table 3 shows some examples of automatically recognised acronyms, and the evaluation is presented in Table 4.

For the ATC experiment, we used the Genia resources [33], which include 1000 Medline abstracts, with overall 40 000 (16 000 distinct) semantic tags annotated for terms in the domain of nuclear receptors. As a golden standard, we used the Genia ontology. In the experiment, the test set contained 10 694 terms belonging to any of the three major Genia classes (namely, nucleic acid, amino acid, Source). These terms have been used as input for the ATC module of ATRACT and the corresponding dendrogram has been produced.

In order to calculate the quality of the dendrogram, we have adopted the average semantic similarity calculation method for measuring the similarity between terms [35]. The average similarity (AS) for two sets of terms is calculated as an AS between the corresponding terms:

Table 4  
Acronym acquisition results

	2008 abstracts	6323 abstracts	50 abstracts
Acronyms recognised	1015	2343	66
Correct acronyms recognised	992	2314	62
Acronyms introduced	–	–	85
Precision (%)	97.73	98.76	93.94
Recall (%)	–	–	72.94

$$AS(X, Y) = \frac{\sum_{x \in X, y \in Y, x \neq y} \text{sim}(x, y)}{|X| + |Y|} \quad (1)$$

The similarity between two individual terms is determined according to their position in a dendrogram: a commonality measure is defined as the number of shared ancestors between two terms in the dendrogram, and a positional measure as a sum of their distances from the root [13]. Similarity between two terms corresponds to a ratio between commonality and positional measure.

The AS values for all pairs of the three Genia classes considered in this experiment were calculated (Table 5). The AS values for elements from the same class (i.e. when  $X = Y$  in Eq. (1)) were greater than the values for elements from different classes. This means that terms belonging to the same Genia class are more closely (i.e. more consistently) placed in the resulting dendrogram. In other words, the average distances between terms belonging to different classes are greater than the average distances within a class. Therefore, we assume that the organisation of terms within the dendrogram produced by

Table 5  
AS-values for the GENIA classes

AS	Nucleic acid	Amino acid	Source	Terms
Nucleic acid	0.498	–	–	3108
Amino acid	0.396	0.492	–	4284
Source	0.390	0.388	0.480	3302

ATTRACT depicts the actual similarities between them.

In order to examine the tag manipulation performance of TIMS, we measured the processing times consumed for executing an interval operation. These times were compared with the time needed by using string-based regular expression matching (REM). We focused on testing the interval operation  $\langle \text{TITLE} \rangle \otimes \langle \text{TERM} \rangle$ , which extracts all terms within titles. In the evaluation process, we used five different samples to estimate IE performance according to their size (namely the number of tags and file size in kb).

Table 6 and Fig. 11 show the results: the processing times of TIMS were about 1.4–1.8 times shorter (depending on the number of tags and the corpus length) than those of

Table 6  
TIMS— practical performance

Sample	1	2	3	4	5
Tags	1146	2383	3730	4799	5876
Size (kb)	92	191	298	382	470
TIMS (ms)	16	28	40	44	62
REM (ms)	24	38	58	80	104

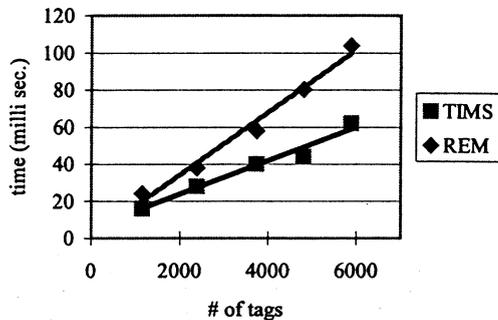


Fig. 11. IE performance (TIMS vs. REM).

REM. Therefore, we conclude that the TIMS tag information management scheme may be also considered as an efficient mechanism to facilitate KA through IE.

## 7. Conclusion

In this paper, we presented a system for literature mining over large KSSs. TIMS is an XML-based integrated KA system, in which we have integrated ATR, ATC, tagged data management and ontology-based knowledge extraction. It allows users to search and combine information from various sources. IE within the system is terminology-driven, with terminology information provided automatically in the XML format. Tag-based retrieval is implemented through interval operations, which, in combination with hierarchical, ontology-based matching, offers powerful means for KA through literature mining.

The preliminary experiments show that the TIMS tag information management scheme is an efficient methodology to facilitate KA and IE in the field of biomedicine.

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 a term

classification system as an alternative structuring model for knowledge deduction and inference (instead of an ontology).

## References

- [1] M. Hearst, Untangling Text Data Mining, Proceedings of ACL, University of Maryland, 1999.
- [2] M. Craven, J. Kumlien, Constructing biological knowledge bases by extracting information from text sources, Proceedings of ISMB-99, Heidelberg, Germany, 1999, pp. 77–86.
- [3] C. Blaschke, M. Andrade, C. Ouzounis, A. Valencia, Automatic extraction of biological information from scientific text: protein–protein interactions, Proceedings of ISMB-99, Heidelberg, Germany, 1999, pp. 60–67.
- [4] M. Hearst, Text mining tools: instruments for scientific discovery, IMA Text Mining Workshop, Institute for Mathematics and its Applications, Minneapolis, USA, 2000.
- [5] T. Rindfleisch, J. Rajan, L. Hunter, Extracting molecular binding relationships from biomedical text, Proceedings of the ANLP-NAACL 2000, ACL, 2000, pp. 188–195.
- [6] J. Pustejovsky, J. Castaño, J. Zhang, M. Kotecki, B. Cochran, Robust relational parsing over biomedical literature: extracting inhibit relations, Proceedings of PSB-2002, vol. 7, Hawaii, 2002, pp. 362–373.
- [7] National Library of Medicine, Medline, <http://www.ncbi.nlm.nih.gov/PubMed/>, 2002.
- [8] D.E. Oliver, D.L. Rubin, J.M. Stuart, M. Hewett, T.E. Klein, R.B. Altman, Ontology development for a pharmacogenetics knowledge base, Proceedings of PSB-2002, vol. 7, Hawaii, 2002, pp. 65–76.
- [9] R. Gaizauskas, G. Demetriou, K. Humphreys, Term recognition and classification in biological science journal articles, Proceedings of Workshop on Computational Terminology for Medical and Biological Applications, NLP-2000, Patras, Greece, 2000, pp. 37–44.
- [10] K. Fukuda, T. Tsunoda, A. Tamura, T. Takagi, Toward information extraction: identifying protein names from biological papers, Proceedings of PSB-98, vol. 3, Hawaii, 1998, pp. 705–716.
- [11] K.T. Frantzi, S. Ananiadou, H. Mima, Automatic recognition of multi-word terms: the *C*-value/*NC*-value method, International Journal on Digital Libraries 3/2 (2000) 115–130.
- [12] H. Nakagawa, T. Mori, Nested collocation and compound noun for term recognition, Proceedings of CompuTerm 98, Canada, 1998, pp. 64–70.
- [13] D. Maynard, S. Ananiadou, Term extraction using a similarity-based approach, in: D. Bourigault, et al. (Eds.), Recent Advances in Computational Terminology, John Benjamins Publishing Company, 2001, pp. 261–278.

- [14] V. Hatzivassiloglou, P. Duboue, A. Rzhetsky, Disambiguating proteins, genes, and RNA in text: a machine learning approach, *BioInformatics* 17/1 (2001) S97–S106.
- [15] P.R.S. Visser, D.M. Jones, T.J.M. Bench-Capon, M.J.R. Shave, An analysis of ontology mismatches—Heterogeneity versus Interoperability, *Proceedings of AAAI Spring Symposium on Ontological Engineering*, Stanford University, California, USA, 1997, pp. 164–172.
- [16] J. Gamper, W. Nejd, M. Wolpers, Combining ontologies and terminologies in information systems, *Proceedings of the Fifth International Congress on Terminology and Knowledge Engineering*, Innsbruck, Austria, 1999, pp. 152–168.
- [17] T. Berners-Lee, The semantic web as a language of logic, available at: [www.w3.org/DesignIssues/Logic.html](http://www.w3.org/DesignIssues/Logic.html), 1998.
- [18] D. Brickley, R. Guha, Resource description framework (RDF) schema specification 1.0, W3C Candidate Recommendation, available at <http://www.w3.org/TR/rdf-schema>, 2000.
- [19] P.G. Baker, A. Brass, S. Bechhofer, C. Goble, N. Paton, R. Stevens, TAMBIS: transparent access to multiple bioinformatics information sources—an overview, *Proceedings of Sixth International Conference on Intelligent Systems for Molecular Biology-ISMB98*, Montreal, 1998, pp. 25–34.
- [20] C. Friedman, P. Kra, H. Yu, M. Krauthammer, A. Rzhetsky, GENIES: a natural-language processing systems for the extraction of molecular pathways from journal articles, *BioInformatics* 17/1 (2001) S74–S82.
- [21] A. Voutilainen, J. Heikkilä, An English constraint grammar (ENGCG) a surface-syntactic parser of English, in: U. Fries et al. (Eds.), *Creating and Using English language corpora*, Rodopi, Amsterdam, Atlanta, 1993, pp. 189–199.
- [22] T. Makino, K. Torisawa, J. Tsujii, LiLFeS—practical programming language for typed feature structures, *Proceedings of Natural Language Pacific Rim Symposium'97*, 1997.
- [23] Y. Miyao, T. Makino, K. Torisawa, J. Tsujii, The LiLFeS abstract machine and its evaluation with the LinGO grammar, *Journal of Natural Language Engineering* 6/1 (2000) 47–62.
- [24] G. Nenadić, I. Spasić, S. Ananiadou, Automatic acronym acquisition and term variation management within domain specific texts, *Proceedings of LREC 2002*, Las Palmas, Spain, 2002, pp. 2155–2162.
- [25] H. Mima, S. Ananiadou, G. Nenadić, ATRACT workbench: an automatic term recognition and clustering of terms, in: V. Matoušek, P. Mautner, R. Moušek, K. Taušer (Eds.), *Text, Speech and Dialogue*, LNAI 2166, Springer, 2001, pp. 126–133.
- [26] M. Krauthammer, A. Rzhetsky, P. Morozov, C. Friedman, Using BLAST for identifying gene and protein names in journal articles, *Gene* 259 (2000) 245–252.
- [27] C. Jacquemin, *Spotting and Discovering Terms Through NLP*, MIT Press, Cambridge, MA, 2001, p. 378.
- [28] A. Ushioda, Hierarchical clustering of words, *Proceedings of COLING '96*, Copenhagen, Denmark, 1996, pp. 1159–1162.
- [29] C. Manning, H. Schütze, *Foundations of Statistical Natural Language Processing*, The MIT Press, 2000, p. 680.
- [30] R. Grishman, TIPSTER phase II architecture design document, New York University, available at <http://www.tipster.org/arch.htm>, 1995.
- [31] V.R. Benjamins, D. Fensel, A.G. Pérez, Knowledge management through ontologies, *Proceedings of the Second International Conference on Practical Aspects of Knowledge Management-PAKM 98*, Basel, Switzerland, 1998.
- [32] M. Uschold, Building ontologies: towards a unified methodology, 16th Annual Conference of the British Computer Society Specialist Group on Expert Systems, Cambridge, UK, 1996.
- [33] Genia project, Genia project home page, <http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/>, 2002.
- [34] H. Mima, S. Ananiadou, An application and evaluation of the *C/NC*-value approach for the automatic term recognition of multi-word units in Japanese, *International Journal on Terminology* 6/2 (2001) 175–194.
- [35] K. Oi, E. Sumita, H. Iida, Document retrieval method using semantic similarity and word sense disambiguation, *Journal of Natural Language Processing* 4/3 (1997) 51–70 (in Japanese).