BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 11 2005, pages 2748–2758
doi:10.1093/bioinformatics/bti338

Data and text mining

MaSTerClass: a case-based reasoning system for the classification of biomedical terms

Irena Spasic[1,*], Sophia Ananiadou[2] and Junichi Tsujii[3]

[1]School of Chemistry, The University of Manchester, Sackville Street, PO Box 88, Manchester M60 1QD, UK, [2]School of Computing, Science and Engineering, The University of Salford, The Crescent, Salford M5 4WT, UK and [3]Faculty of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo 113-0033, Japan

Received on November 19, 2004; revised on February 13, 2005; accepted on February 17, 2005

Advance Access publication February 22, 2005

ABSTRACT

Motivation: The sheer volume of textually described biomedical knowledge exerts the need for natural language processing (NLP) applications in order to allow flexible and efficient access to relevant information. Specialized semantic networks (such as biomedical ontologies, terminologies or semantic lexicons) can significantly enhance these applications by supplying the necessary terminological information in a machine-readable form. With the explosive growth of bio-literature, new terms (representing newly identified concepts or variations of the existing terms) may not be explicitly described within the network and hence cannot be fully exploited by NLP applications. Linguistic and statistical clues can be used to extract many new terms from free text. The extracted terms still need to be correctly positioned relative to other terms in the network. Classification as a means of semantic typing represents the first step in updating a semantic network with new terms.

Results: The MaSTerClass system implements the case-based reasoning methodology for the classification of biomedical terms.

Availability: MaSTerClass is available at http://www.cbr-masterclass.org. It is distributed under an open source licence for educational and research purposes. The software requires Java, JWDSP, Ant, MySQL and X-hive to be installed and licences obtained separately where needed.

Contact: i.spasic@manchester.ac.uk

Supplementary information: Available at http://www.cbr-masterclass.org

1 INTRODUCTION

A *terminology* is a collection of terms (denoting domain-specific concepts such as genes, proteins, etc.) typically organized into a classification hierarchy. The core of such a hierarchy is based on the general–specific relation. Other relations (e.g. biochemical interactions) are used to complete the model of a specific domain. Concepts are natively assorted into groups, either *classes* (where all concepts share a common description) or *clusters* (groups of correlated concepts), and the organization of terms in a terminology needs to reflect such properties consistently. It should also be extensible so that new terms, representing newly discovered concepts, can be efficiently incorporated into the existing structures by associating them with other terms. These associations should at least include the links between the correlated terms, thus forming the clusters of semantically related terms, and the generalization of terms sharing the same set of features into appropriate classes.

Given a corpus of relevant textual documents, the techniques for automatic term recognition, clustering and classification, can help to automate the process of creating and maintaining a specific terminology. The need for automation is particularly evident in biomedicine, where manual approaches cannot cope with an enormous and ever growing number of terms and the complex structure of biomedical terminologies.[1]

In this paper we describe an approach to classification of biomedical terms, whose results can support automatic terminology update. Structured up-to-date terminological information can then be used to improve the quality of natural language processing applications (such as information extraction and retrieval, document classification and summarization, etc.), thus making it easier for biomedical experts to navigate through huge volumes of scientific documents.[2]

Automatic classification of biomedical terms is difficult due to loose naming conventions, which rarely aim to encode particular functional properties of the underlying concepts in a systematic manner.[3] For the complexity reasons caused by inconsistent and imprecise naming practice, many methods developed for classification of biomedical terms target only a limited number of specific classes through manual identification of features typical of their terms. For example, Fukuda *et al*. (1998) developed a rule-based method for the recognition of protein names exploring their orthographic and lexical features (e.g. capital letters, digits and special characters). A series of methods have been implemented following this idea. For instance, Narayanaswamy *et al*. (2003) extended

*To whom correspondence should be addressed.

[1]UMLS (http://www.nlm.nih.gov/research/umls) contains over one million concepts named by 5 million terms, organized into a hierarchy of 135 classes and interconnected by 54 different relations.

[2]Medline (http://www.ncbi.nlm.nih.gov/PubMed) refers to ∼12 million journal articles, expanding for more than 10 000 references weekly. Over 571 000 references were added in 2004.

[3]There is no exact consensus on what constitutes a biomedical term even when it is restricted to, e.g. proteins and genes (Narayanaswamy *et al*., 2003), although the naming conventions do exist for these concepts (Oliver *et al*., 2002).

Fukuda's approach to six classes of biomedical entities: gene or protein, gene or protein part, chemical, chemical part, source and others. The main problem in such approaches is that term classification rules are often obscure and imprecise due to loose naming conventions.

In order to cope efficiently with the complexity of knowledge needed to perform reliable classification, many approaches resort to machine learning (ML) techniques to detect features that characterize specific classes. Currently, the ML term classification methods exploit little or no biomedical knowledge for guided learning. Usually, general-purpose ML algorithms are applied to shallow representation of text (Nedellec, 2002). For instance, Stapley *et al*. (2002) used a support vector machine (SVM) approach with a non-structured representation of text to classify gene names (represented as vectors of contextual features, defined as single words co-occurring in the same abstract) with respect to their subcellular location. Recently, there have been a number of other applications of SVMs for classification of biomedical terms (Kazama *et al*., 2002; Lee *et al*., 2004; Collier and Takeuchi, 2004). These approaches differ from those of Stapley *et al*. (2002) with respect to the features used which largely resemble those proposed by Fukuda *et al*. (1998). Alternatively, probabilistic methods such as naive Bayes classification (Hatzivassiloglou *et al*., 2001; Nobata *et al*., 2000) and hidden Markov models (Collier *et al*., 2001) have been used.

All mentioned methods require large amounts of training data and significant training time to prevent overfitting. Namely, they are optimized to fit the training data, which may not be ideal approximation of the real data. Thus, such algorithms require large training sets and need to be periodically retrained upon the advent of new data. They also underperform for minority classes due to the data sparsity problem. Furthermore, they explicitly differentiate between the training phase (in which classification rules are learnt) and the application phase (in which the learnt rules are applied). However, satisfactory rules cannot always be produced (e.g. due to weak correlation between term features and their classes).

In this paper we suggest an alternative ML approach. Case-based reasoning (CBR) is particularly suitable for the problem of term classification in biomedicine, because it is pragmatic and robust enough to deal with the complexity of both natural language and the biomedical domain as explained in the following section which outlines the basic principles of this methodology.

2 METHODOLOGY

CBR is based on remembering specific experiences that may be useful for the problem (case) being solved. It may be viewed as a multistage cycle involving the four '*re-*' (Aamodt, 1995): (1) *retrieve* the most similar case, (2) *reuse* the case to solve the new problem, (3) *revise* the suggested solution and (4) *retain* the useful information obtained during problem solving. Therefore, new problems are solved by adapting solutions that provided satisfactory results for similar problems, thus avoiding the need for an explicit model of the problem domain (Watson and Marir, 1994). Instead, only features relevant in the context of the current problem need to be identified. Therefore, CBR makes use of *specific* (as opposed to generalized) knowledge in both problem solving and learning (Kolodner, 1993). Specific information about the past experiences is regarded as *knowledge*, unlike in rule-based or model-based systems, where it is treated as *data*. In this manner, CBR tackles the main issues in other ML systems, such as the lack of robustness and flexibility, confinement to narrow problem domains and difficult development and maintenance (Aamodt, 1995).

Memory forms a basis for the learning ability of CBR systems (Watson and Marir, 1994). Nevertheless, such a trivial form of learning still supports generalization and abstraction implicitly through the use of similarity (Aamodt, 1995). Therefore, a CBR system is capable of learning without explicitly generalizing specific cases into formulas, rules or other symbolic representations (Globig *et al*., 1997). Such a lazy or demand-driven approach has the following advantages (Aha, 1998; Leake, 1996): easier knowledge acquisition, reduced problem solving predisposition, incremental learning and improved user acceptance due to explanation based on precedents.

The general advantages of CBR are particularly emphasized in the family of biomedical sciences because of the homologous nature of biological systems rooted in evolution (Jurisica and Glasgow, 2004). Therefore, biomedical experts themselves often use analogical reasoning to plan and conduct experiments exploring similarities between new and known systems. Furthermore, biomedical field is overwhelmed by data but often lacks exact and complete theories that could interpret such amounts of data correctly and efficiently. For example, due to huge amounts of data, many unknowns, incomplete theories and extremely dynamic nature of molecular biology, reasoning in this domain is often based on experience as opposed to general knowledge. CBR has been successfully applied in molecular biology to solve a variety of problems, e.g. protein crystallization, genomic sequence analysis, protein structure determination, etc.

Similarly, Schmidt *et al*. (2001) emphasize the appropriateness of CBR for medical domain using an argument that the knowledge of medical experts is 'a mixture of textbook knowledge and experience'. The textbook knowledge can be represented by rules or other models, while the experience can be represented by cases. Moreover, medical cases are professionally documented resulting in an invaluable repository of information, where CBR can be used as 'an engine for intelligent text processing and retrieval, data mining and projective reasoning' in order to fully exploit available information especially in the age of electronic patient records (Macura and Macura, 1997). Furthermore, the typical decision making process of a medical practitioner involves reasoning with cases, which establishes medicine as an interaction of research and practice, where clinical practice is characterized by a collection of accumulated cases. CBR and its learning strategy mirror the learning process of a medical practitioner when faced with different cases (patients, symptoms, diseases and treatments). Hence, cognitive adequateness and explicit representation of experience make CBR a natural ML approach in medicine (Gierl *et al*., 1998). This fact has been restated by numerous medical applications including diagnosis, classification, planning, prognosis, tutoring, etc.

In view of our specific problem of classifying biomedical terms, CBR can readily utilize the large body of biomedical texts as the training data without the need to map term features to the corresponding classes, a priori. Instead, generalization (or learning) is performed on demand based on the currently available data and with respect to a particular term being classified. This helps to reduce overfitting, which in other ML approaches stems from an attempt to generalize in advance so as to fit most of the available training data. Moreover, by automatically adapting to the data available at the moment of classification and not training, the need for retraining is avoided in CBR. These properties particularly suit the dynamic nature of the biomedical domain (new data become available daily) and the difficulty in generalizing term properties into corresponding classes (due to loose naming conventions and the variability of natural languages).

Having chosen CBR as a methodology for classification of biomedical terms, the next step is to decide how to utilize it for this problem. First, note that there is a large amount of electronically available biomedical documents describing specific discoveries and a number of knowledge repositories describing general biomedical knowledge. The biomedical knowledge repositories, although typically incomplete, still contain large volumes of information in a structured form. On the other side, scientific documents contain comprehensive up-to-date information structured by the natural language rules. Our intention was to use a corpus of biomedical texts (in which known terms are classified within a biomedical ontology) as a collection of classification experiences and perform classification of new terms by making analogies on the fly. The role of an ontology in this context is to provide a classification scheme, aid semantic interpretation of domain-specific text and

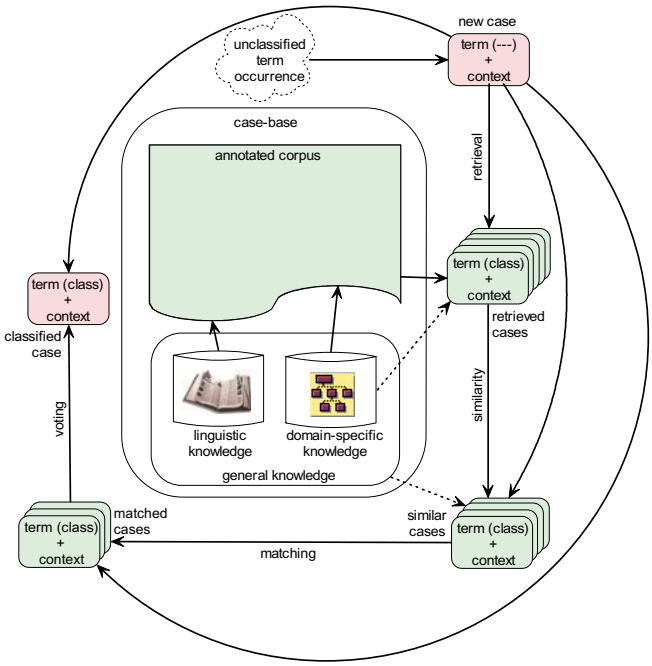

Fig. 1. The MaSTerClass system organization.

support the semantic aspect of the similarity measure used to identify term contexts similar to the one used for the classification of a new term.

3 THE MaSTerClass SYSTEM

In this section, we introduce the MaSTerClass (**ma**chine **s**upported **ter**m **class**ification) system. Although CBR served as the methodological framework, the actual techniques needed to be developed specifically for the given problem. Figure 1 depicts the organization and the workflow of MaSTerClass. Two types of general knowledge (linguistic and domain-specific) are utilized. The linguistic knowledge is used to structure textual information, i.e. to extract the underlying syntactic structure and represent it explicitly in a machine-usable form. A corpus of biomedical abstracts was automatically annotated with lexical, syntactic and terminological information. The rules for recognizing syntactic structures of interest (e.g. noun and verb phrases) have been specified by the corresponding local grammars (Gross, 1997). Terms have been identified in the corpus by looking up the UMLS[4] dictionary and applying the NC-value[5] method. The domain-specific knowledge adopted from UMLS consists of terms and the corresponding concepts (i.e. concept identifiers) organized into a classification hierarchy.

[4]UMLS is an ontology, which merges over 100 biomedical vocabularies aiming to facilitate the development of information systems for text processing in biomedicine by providing a formal representation of domain-specific knowledge in order to process, retrieve, integrate, and aggregate biomedical data and information contained in the relevant literature.

[5]The NC-value (Frantzi and Ananiadou, 1999) method extracts multiword terms [>85% of terms are multiword (Nakagawa and Mori, 2003)] by using linguistic knowledge to propose term candidates through their formation patterns followed by frequency-based analysis used to estimate their 'termhood'.

Note that the functionality of the MaSTerClass system covers term classification only. While we currently use UMLS and the NC-value method to annotate the corpus terminologically, they are external to the system and by no means part of it. The same remark applies to the use of other linguistic tools, such as tagger and parser. In other words, any other tagger, parser or term recognition method can be used just as well without the need for reimplementation. Similarly, any other ontology could generally be converted into our internal format and stored into the database used by the system.

The annotated corpus of biomedical abstracts used in combination with the UMLS ontology forms the case-base of the MaSTerClass system. It is used for term classification by remembering specific classification contexts that can be useful for the term currently being classified. New terms are classified by adapting (or more precisely, adopting) the classes of similar terms in similar contexts. Each case in this approach consists of a term occurring in a specific context (description of the problem) and one or more classes that apply to that term occurrence (solution).

It would not be efficient (or even feasible) to compare a new term to all available terms and their contexts. For this reason, only potentially similar contexts are retrieved by using terminological information from the ontology to locate other contexts containing semantically similar terms and domain-specific verbs. The new case is compared with each retrieved case by the SOLD (**s**yntactic, **o**ntology-driven and **l**exical **d**istance) measure, which compares their syntactic and semantic properties. It is based on the concept of the edit distance (ED), which has been widely used for approximate string matching (Navarro, 2001). It compares two strings through the minimal number (or cost) of edit operations (including deletion/insertion of a character and the replacement of two characters in the two strings). The SOLD measure uses the same operations, but applies them to syntactic elements (obtained through lexical tagging and partial syntactic parsing) and terms (obtained automatically by the NC-value method or dictionary look-up). Both linguistic and terminological knowledge are used to approximately match individual context elements.

The most similar retrieved cases are selected for further processing. The selection process, thus, further reduces the search space to be processed in the matching phase, in which the new case and old cases are aligned according to the combinations of edit operations resulting in the minimal alignment cost. The purpose of alignment is to match the unclassified term to a classified term that has a similar role in a similar context (both syntactically and semantically). The successfully matched cases are used collectively to propose the class(es) for the unclassified term through a voting procedure. Each case contributes to the final classification results by delegating votes for the classes attached to the matched classified term. For example, in Figure 2 let us suppose that the unclassified term *5 alpha-dihydrotestosterone* is aligned with the term *testosterone* classified as *hormone*. Then the class suggested for *5 alpha-dihydrotestosterone* by this alignment is *hormone* as well. As multiple cases are used, it is expected that any outlying cases that got through retrieval, selection and matching would be outvoted at this stage. Finally, the classes receiving most votes are suggested for the given unclassified term. Following a validation procedure performed by a human curator, the newly learnt case (i.e. the term successfully classified based on its context) can be added to the case-base.

testosterone	***but***	***not***	***progesterone***	***inhibits***	***[3H]R1881***	***binding***	***to***	***AR***	***---***	***-------***
5 alpha-dihydrotestosterone	*----*	*----*	*----------*	*inhibited*	*[3H]R1881*	*binding*	*to*	*the androgen receptor*	*in*	*kidney*

Fig. 2. An alignment of similar contexts.

4 MODULES

In the previous section we described the general workflow of the MaSTerClass system. Here we provide more details about its modules.

4.1 Case-base

A case is a unit encapsulating knowledge relevant to a particular experience (Watson and Marir, 1994). It is typically structured into the problem and solution parts. Cases may be represented as feature vectors, frames, objects, predicates, semantic nets, rules, etc. The case representation affects the way in which the similarity between cases can be assessed and the efficiency of retrieval. In MaSTerClass, the problem is a term occurrence found in text, while the solution represents a set of classes applicable to the given term. As the context in which a term occurs is often necessary for its classification,[6] we can view the problem part as a term within a given context. The next question is how the context should be represented, e.g. bag of co-occurring words or terms, text window of a fixed length, sentence, paragraph or document containing the term, lexico-syntactic pattern matching the context, etc. In our approach, we kept as much contextual information as possible. First, each context has been annotated with lexical, syntactic and terminological information and treated as a sequence of syntactic and terminological units. Basic syntactic structures (e.g. noun and verb phrases) are recognized through partial parsing. Dictionary terms are annotated together with the ones recognized by the NC-value method. Both terms and basic syntactic structures are most often multiword units. By grouping and annotating these multiword units the context is structured, i.e. functional relations between consecutive single words are preserved. In addition, the positional information for individual context elements is retained. Second, the relation of a local context and the global discourse is preserved by deciding to use a pointer to a term occurrence in the corpus rather than a copy of its context. It is a flexible approach, because the structure and length of a context need not be prespecified.

A term is classified by mapping it to its class and linking it to the knowledge on that class represented by the ontology. Thus, the solution to the problem of classifying a term is a part of the ontology concerned with that particular term.

4.2 Similarity measure

CBR relies on the hypothesis that similar problems tend to have similar solutions. Therefore, the similarity assessment is a key issue in CBR. It depends on a problem domain and case representation. In the chosen representation, each case corresponds to a term context treated as a sequence of basic syntactic structures and we need to approximately match such sequences. ED has been widely used for approximate string matching, where the distance between identical strings equals zero and increases as the strings get more dissimilar with respect to the symbols they contain and the order in which they appear. ED is defined as the minimal cost incurred by the changes needed to transform one string into the other. These changes may include insertion or deletion of a single character, replacement of two characters in the two strings and transposition of two adjacent characters in a single string. The choice of edit operations and their costs depends on a specific application. ED has been successfully utilized in NLP to deal with alternate spellings, misspellings, the use of upper- and lower-case letters, etc. It has also been used in terminological processing for the recognition of orthographic term variants. For example, Tsuruoka and Tsujii (2004) compared protein names based on their internal properties focusing on orthographic features. Our intention, however, is primarily to explore contextual properties of terms.

In this case, it is more convenient to apply ED at the *word* level rather than the *character* level, i.e. the character-based ED does not cope well with permutations of words. For instance, judging by the 'conventional' ED, *stone in kidney* is more similar to *stone in bladder* than *kidney stone*. Alternatively, approximate string matching can be viewed as the problem of pairing up their words so as to minimize their ED (French *et al*., 1997). Recently, ED has been applied at the word level to allow different wordings and syntactic mistakes in the phrase-based text search (Navarro *et al*., 2000). In this approach, ED was simply applied to words as opposed to characters. We, however, developed the SOLD measure by enriching the basic ED approach with both linguistic [relying on part-of-speech (POS) tagging and partial parsing] and biomedical (using an ontology) knowledge (Spasic and Ananiadou, 2005).[7]

Partial parsing is applied to POS-tagged text to group subsequent words into basic syntactic structures. ED applied to blocks of words rather than individual words is 'forced' to take syntactic structure (at the phrase level) into account and prevented from artificially disassembling syntagmatic structures by applying edit operations to individual words. By choosing to replace syntactic categories with similar properties at lower costs (e.g. nouns and pronouns), ED can also be used to compare the syntactic structure at the sentence level, i.e. the sentences receiving low ED values are the ones that can be transformed into one another using a small number of low-cost edit operations, implying that their overall syntactic structure is fairly isomorphic. Furthermore, the cost of deleting (or equivalently inserting) contextual elements depends on their semantic load. For example, terms refer to domain-specific concepts and as such are the most important means of communicating knowledge in a specific domain. Therefore, their deletion is the costliest operation, indicating that important information is lost.[8]

[6]When classifying biomedical terms, it is by all means necessary to include their context into consideration since (1) terms do not necessarily encode sufficient information to infer their semantic types and (2) the meaning of a term can be modified by its context.

[7]The remainder of Section 4.2 represents a brief report on the similarity measure, which has been extensively described in Spasic and Ananiadou (2005). The reader may wish to read this open-access paper available at http://helix-web.stanford.edu/psb05 before proceeding to Section 4.3.

[8]All types of context elements used and the costs of edit operations involving them are specified in Spasic and Ananiadou (2005).

ED usually relies on the exact matches between symbols unless 'wild card' symbols are allowed. This is unsuitable for word comparison, because words are inflected. Also, the term variation phenomenon can cause synonymous terms not to match. We made the ED approach more flexible with respect to lexical variation. For example, two inflected word forms match if both their lexical categories and their base forms are identical. When two terms are compared, information from the ontology is utilized. All semantic classes in UMLS are organized into a hierarchy, which can be used to quantify their similarity. The tree similarity (ts) between two classes C_1 and C_2 is calculated according to the following formula:

$$\mathrm{ts}(C_1, C_2) = \frac{2 \cdot \mathrm{common}(C_1, C_2)}{\mathrm{depth}(C_1) + \mathrm{depth}(C_2)} \quad (1)$$

where $\mathrm{common}(C_1, C_2)$ denotes the number of common classes in the paths between the root and the given classes, and $\mathrm{depth}(C)$ is the number of classes in the path connecting the root and the given class. This formula is a derivative of Dice coefficient where ancestor classes are treated as term features. Since the UMLS ontology supports multiple classification of terms, we estimate the similarity between two terms as the maximal similarity between their classes. The similarity between two terms quantified in this manner is used to modify their replacement cost accordingly. The calculation of the replacement cost for two verbs described in the ontology is analogous.

The approach used in the ontology-driven component is applicable only to classified terms and verbs. Currently, biomedical ontologies are inherently incomplete due to the fast-growing number of terms. Therefore, it would be useful to use clues other than the ones explicitly stated in the ontology in order to extend the semantic comparison to unclassified terms and verbs. We exploit lexical and morphological clues as they often indicate semantic similarity. For example, *5 alpha-dihydrotestosterone* and *testosterone* are lexically similar, and this fact can be used to infer their semantic similarity. We utilized the standard ED approach applied at the character level in order to estimate lexical similarity.

Finally, the SOLD measure is computed using the standard dynamic programming approach for the calculation of ED (Wagner and Fischer, 1974).

4.3 Retrieval

Retrieval in CBR serves to improve the efficiency of the whole system by allowing for crude (and computationally less expensive) comparison of a new case against the ones stored in the case-base. The result is the search space considerably reduced in size. Finer (and costlier) comparison is then performed against the retrieved cases. Ideally, the retrieved cases should be the ones most similar to the new case. However, this is not always straightforward to achieve, so the compromise should be made between two conflicting objectives: efficiency and precision.

We now describe the retrieval approach used in MaSTerClass. Let us recall that, given a non-classified term occurring in a specific context, we would like to retrieve terms occurring in similar contexts. We previously described how the contextual similarity is assessed by applying the SOLD measure (Spasic and Ananiadou, 2005). We would like to retrieve those contexts that would most probably minimize the value of this measure. We adopted a heuristic approach exploring the notions of semantic matching and terminological load to achieve this objective.

Terms tend to co-occur with other terms and verbs denoting specific relations between them. Terms and domain-specific verbs also carry the heaviest semantic load. These facts are used to retrieve other similar context (regardless of their structure) by using the terms and verbs found in the context of the unclassified term. Contexts matching semantically are the ones that share a sufficient number of terminologically relevant elements (i.e. terms and domain-specific verbs). Semantic matching makes use of terminological information and is ontology-driven. Namely, in UMLS, both terms and verbs are hierarchically organized. These hierarchies are used to quantify the similarity between terms and verbs [Formula (1)]. When retrieving contexts through semantic matching, terms and verbs found in it are used to retrieve their classes (and their close ancestors). The resulting set of classes is then used to retrieve their instances.[9] All terms and verbs obtained in this manner form a set of semantically matching tokens. These tokens are then used to query the corpus in order to retrieve semantically similar contexts (i.e. the ones that contain sufficient number of semantically matching tokens). Let us exemplify the process of semantic matching by considering the following sentence:

> Radioinert ***testosterone*** (*T*) and *5 alphadihydrotestosterone* (*DHT*) but not *androtenedione*, *progesterone*, *estradiol-17 beta*, *estrone* or *cortisol* in a 50-fold molar excess **inhibited** [*3H*]*R1881* **binding** to the *AR* in *spinal cord*, *heart*, *kidney* and *RT*.

in which the term *testosterone* needs to be classified. Let us suppose that the italicized terms have been identified. In addition, let us assume that the verbs *inhibit* and *bind* have been identified by the tagger. These terms and verbs are used to retrieve other similar terms and verbs. For example, the term *progesterone* classified as a *hormone* is used to retrieve all other terms from this class, e.g.: *thyrotropin-releasing hormone*, *glucocorticoid*, *endorphin*, etc. Similarly, the term *AR* is used to retrieve its expanded form *androgen receptor* and all other terms from the *receptor* class, e.g.: *thyroid hormone receptor beta*, *thrombomodulin*, *nuclear receptor*, etc. For the verb *inhibit* the following similar verbs are retrieved: *prevent*, *stop*, *hinder*, *repress*, *impede*, etc. All retrieved terms and verbs form a set of semantically matching tokens. These tokens are matched against the corpus to retrieve other sentences containing them, such as:

> NFI-C does not **repress** *progesterone* induction of the MMTV promoter in HeLa cells, suggesting that *progesterone* induction of the promoter differs mechanistically from *glucocorticoid* induction.

[9] For efficiency reasons (e.g. when classes are too large measured by the number of their instances), a 'caching' approach can be used in which each classified term should be annotated in the corpus with applicable class labels. In this manner, the retrieval of class instances from the ontology is avoided as well as the subsequent complex (measured by the number of matching tokens) queries against the corpus. Instead, ontology is used only to retrieve the class labels and use them to simply query the corpus with the given values of class-label attributes. Furthermore, this attribute can be indexed to speed up the access to relevant terms in the corpus. However, in this approach the corpus should be periodically re-annotated with class information in order to synchronise the corpus with ontology content, which is a step not needed in the original 'dynamic' approach.

For two sentences to be sufficiently close with respect to the SOLD measure, it is desirable for them not only to share semantically similar terms and verbs, but also to have a similar number of them, since their deletion and insertion are the costliest edit operations. In order to take this fact into account during the retrieval process, we introduce the notion of terminological load defined as the number of terms and domain-specific verbs in a given sentence. Given an input sentence, other sentences with similar terminological load are retrieved. Obviously, the retrieval based on the terminological load does not consider the semantic types of terms and verbs, but simply their number. In order to compensate for this, terminological load is combined with previously described semantic matching, thus retrieving sentences containing a similar number of similar terms and verbs.

4.4 Classification

Once the potentially similar sentences are retrieved from the corpus and their similarity assessed by the SOLD measure, the most similar ones are retained by dynamically setting a distance threshold (t) between the minimal (m) and average (a) value of the SOLD measure for the retrieved sentences: $t = m + (a - m) \cdot d$, where d ($0 \leq d \leq 1$)[10] is a parameter determining how similar the selected sentences should be. The greater the value of d, the greater the acceptance rate. Multiple sentences are typically selected to perform classification.

Recall that the SOLD measure is a modification of ED, which is based on three types of edit operations: insertion, deletion and replacement. An optimal alignment is a sequence of these operations, whose total cost equals the value of ED. Note that there can be more than one optimal alignment. Optimal alignments can be retrieved from the cost matrix produced when calculating the ED by the dynamic programming method. Given an optimal alignment, two sentences are aligned accordingly. In such an alignment, we are interested in a syntactic element aligned with the considered unclassified term. When it is aligned with a classified term, we hypothesize that they belong to the same class(es).

The classification results obtained separately for each sentence are combined through a voting procedure. The classes with the majority of votes are suggested as potential classes for the given term. To be precise, a dynamic vote threshold is set as the product of the maximal votes received by a class and the parameter p ($0 \leq p \leq 1$),[11] that determines what percentage of the maximal number of votes received is regarded acceptable. If $p = 0$, then all classes which received a positive number of votes are suggested. On the other extreme, if $p = 1$, then only the class(es) (>1 if there is a tie) receiving the maximal number of votes are suggested. By using the parameter p we provided support for multiple classification. It supports the fact that biomedical concepts often belong to multiple classes depending on the classification aspect used. For example, genes can be classified with respect to their function, subcellular location or phenotype [Gene Ontology (http://www.geneontology.org)]. The ontology used in this work includes two major branches in the hierarchy depending on the point of view at a chemical, which can be structural or functional. Many terms are classified in both of these subhierarchies (e.g. many *hormones* are also classified as *pharmacologic substances*).

[10]We have used $d = 1.0$ in the experiments reported later.

[11]We have used $p = 0.9$ in the experiments reported later.

Table 1. The classification scheme

Chemical viewed functionally
Pharmacologic Substance
Antibiotic
Biomedical or dental material
Biologically active substance
Neuroreactive substance or biogenic amine
Hormone
Enzyme
Vitamin
Immunologic factor
Receptor
Indicator, reagent, or diagnostic aid
Hazardous or poisonous substance

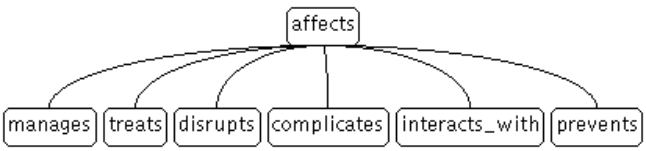

Fig. 3. A portion of the UMLS Semantic Network: 'affects' hierarchy

A run-through example summarizing the whole classification process is given as Supplementary material.

5 EVALUATION

5.1 Resources

The corpus used as part of the case-base consists of 2072 abstracts on nuclear receptors retrieved from Medline (2004). Each abstract consists of a single title and a number of sentences. The total number of sentences in the corpus (not counting the titles) is 19 449. The initially POS-tagged corpus has been terminologically processed. We have chosen UMLS as the classification scheme focusing on a subtree of 13 classes, in which chemicals are classified according to their functional characteristics (Table 1). All occurrences of 1643 terms contained in the UMLS dictionary have been annotated, resulting in a total of 24 963 annotated term occurrences. In addition, the NC-value method (Frantzi and Ananiadou, 1999) has been applied to recognize terms not listed in the dictionary. A total of 2757 terms have been recognized and annotated in the corpus, resulting in additional 28 935 annotated term occurrences. Each sentence has been annotated with its terminological load, the average value being 3.21. In addition, each sentence has been partially parsed in order to recognize syntactic structures of interest. As a result of partial parsing, each sentence is represented as a sequence of blocks, the average number of blocks per sentence being 22.34.

In addition to terms, an ontology of verbs was constructed using the part of UMLS Semantic Network that organizes domain-specific relationships into a hierarchy (Fig. 3). We used the fact that these relationships are expressed by verbs to convert this part of UMLS into an 'ontology of verbs'. We used the initial hierarchy and the given verbs as a starting point into which we manually placed additional domain-specific verbs, hand picked from the list of high frequency

Table 2. The evaluation setup

Terms	Number	Percentage (of total)	Percentage (of classified)
Training	18 236	33.83	77.67
Validation	2405	4.46	10.24
Testing	2838	5.27	12.09
Unclassified	30 419	56.44	—
Total	53 898	100.00	—

verbs extracted automatically from the corpus. The resulting ontology covers 99 infinitive forms of verbs distributed across 23 classes. A total of 55 581 verb occurrences have been recognized out of which 14 560 have been treated as domain specific.

5.2 Evaluation setup

The classification experiments were performed only for terms from the corpus that are classified in the ontology in order to automatically evaluate the classification results. The corresponding concepts (not terms) were divided randomly into three sets (using the approximate ratio 15:15:70) and mapped into the sets of terms to be used for validation, testing and training respectively. Table 2 summarizes the distribution of term occurrences across the training, validation, testing and non-classified sets of terms.

5.3 Evaluation measures

We used a standard set of evaluation measures to quantify the results of the classification experiments. These measures include precision, recall and F-measure. The precision and recall are calculated for each class separately according to the following two formulas:

$$P = A/(A + B) \quad (2)$$

$$R = A/(A + C) \quad (3)$$

where A is the number of true positives (the number of times the class was correctly predicted), B is the number of false positives (the number of times the class was incorrectly predicted) and C is the number of false negatives (the number of times the class was incorrectly not predicted). The precision and recall for all classes collectively can be calculated through macro-averaging or micro-averaging. The macro-averaged precision and recall are calculated by averaging the precision and recall obtained for each class separately by Formulas (2) and (3). Alternatively, when calculating the micro-averaged values, the numbers of true positives, false positives and false negatives obtained for each class separately are summed up to obtain the corresponding overall numbers. The micro-averaged precision and recall then combine these values as before [Formulas (2) and (3)]. In all cases, the F-measure is calculated as the harmonic mean of precision and recall: $F = 2 \cdot P \cdot R/(P + R)$.

The problem of judging the classification performance based on the described measures is that they only consider whether the predicted class is correct or not, and conversely whether the actual class is predicted or not. This may be too crude for extensive classification schemes, because the probability of a correct prediction decreases as the number of classes increases. In such cases, a simple method that maps every instance to the largest class could easily 'outperform' other, more subtle classification methods, since the number of correctly classified instances would be large as well. However, the usability of such 'better' results is seriously reduced, since they offer no information gain. However, a method that often fails to make correct predictions, but consistently makes predictions 'close' to the correct classes, can be more useful because it focuses on the correct class neighbourhood (as opposed to a single correct class).

Therefore, we introduce a concept of graded precision and recall, where we measure the distance (or similarity) between the predicted and actual classes rather than their equality. Since the classification scheme used is hierarchically organized, we used the tree similarity measure [Formula (1)] to modify the numerators in Formulas (2) and (3). Previously, the numerator A was used to count true positives for each class C in the following manner: $A = \sum a_t$, where t enumerates the testing term occurrences: a_t is 1 if $C \in \text{predicted}(t) \cap \text{actual}(t)$ and 0 otherwise, where predicted(t) and actual(t) denote the sets of predicted and actual classes, respectively for the given term t. In our evaluation approach, we want to measure the distance between the predicted and actual classes rather then comparing them binary. For example, given a class C, for each testing term occurrence t, we compare the given class with the term's actual classes looking for the minimal distance: a_t^P is calculated as the maximal value of $\text{ts}(C, C_A)$ for all classes $C_A \in \text{actual}(t)$ if $C \in \text{predicted}(t)$ and 0 otherwise. In this manner we measure the degree of incorrectness. Similarly, we compare the given class with all predicted classes for each term t looking for the minimal distance in order to estimate the recall: a_t^R is assigned the maximal value of $\text{ts}(C, C_P)$ for all classes $C_P \in \text{predicted}(t)$ if $C \in \text{actual}(t)$ and 0 otherwise. We measure by how much the system missed a correct class. Finally, the numerators in Formulas (2) and (3) are modified as follows: $A^P = \sum a_t^p$ and $A^R = \sum a_t^R$, giving the formulas for the graded precision and recall: $\text{GP} = A^P/(A + B)$ and $\text{GR} = A^R/(A + C)$. Finally, the values obtained for individual classes are combined as before to calculate macro- and micro-averaged values.

5.4 Baseline methods

We compare the results of our experiments with those obtained by six baseline methods. We relied on methods commonly used to evaluate classification results, such as random classifier and the method that assigns the largest class to all objects of classification. In addition, we implemented a naive Bayes classifier and a rule-based classification method.

The first baseline method (B1) assigns a random class to each term occurrence. The following three methods (B2–B4) map each term occurrence to the largest class measured by the number of its concepts, terms and term occurrences respectively.

A naive Bayes classifier (whose goal is to maximize the conditional probability of a given term being assigned to a specific class based on the features used to represent the term) was used as the fifth baseline method (B5). Each term is represented as a bag of co-occurring words, i.e. all single words occurring with the given term within a sentence. The aforementioned conditional probability is then estimated as the product of the class probability (estimated as the ratio between the number of all terms labelled with the given class and the total number of terms) and the conditional probabilities of features given the class (estimated as the ratio between the number of times a given single word co-occurs with terms from the given class and the number of all single words co-occurring with terms from the given class). Finally, we used rule-based classification

Table 3. A sample of term classification rules

if	Term contains a word starting with prefix '*immun-*' or '*anti-*'
then	class is '*Immunologic Factor*'
if	Term contains any of the words '*toxin*', '*insecticid*' or '*pesticid*' or a word starting with prefix '*carcin-*', '*cancer-*' or '*radioactiv-*'
then	class is '*Hazardous or Poisonous Substance*'
if	Term contains a word ending with suffix '*-cyclin*' or '*-mycin*'
then	class is '*Antibiotic*'

(Table 3) similar to that of Fukuda *et al*. (1998) as the sixth baseline method (B6).

5.5 Experiments

We conducted a series of three experiments: E1, the CBR classification method used in MaSTerClass; E2, the same method supplied with more extensive biomedical knowledge; E3, the CBR method combined with classification rules exploiting the internal term characteristics. The hypothesis behind the experiment E2 is that the knowledge contained in the ontology may not be equally discriminative for all classes. In other words, precision and recall for individual classes may depend on the completeness of the ontology used. Broader and more fine-grained ontologies should have higher discriminative power. For example, in the classification scheme used (Table 1), *receptors* are expected to co-occur with *hormones* and *vitamins* (both classes being present in the classification scheme), whereas *hazardous* or *poisonous substances* are expected to co-occur with terms denoting *diseases*, *syndromes*, *poisoning*, etc. (not covered by the classification scheme). To test this hypothesis, we expanded the classification scheme with other classes found in UMLS. The unclassified term occurrences were matched against the whole UMLS ontology and the retrieved information incorporated into the smaller ontology. The reason we did not retag the corpus with all terms found in the UMLS, but instead we only classified already annotated unclassified terms, is that we wanted to examine the effects of the classification information being attached to terms against the absence of this information. A total of 2757 originally unclassified terms were identified in the corpus, out of which 547 were found in the UMLS. These terms resulted in 186 concepts, 1329 term variants and 53 classes being added to the core ontology used for the experiments. Based on the newly available classification information, 6774 term occurrences in the corpus were additionally annotated as classified.

With the lack of strict naming conventions in biomedicine reflecting particular functional properties of terms, context may often be the only clue to their meaning. Although there are no general terminological standards which would help discriminate between specific classes of terms in biomedicine, there are naming conventions for some types of concepts in the domain, e.g. *genes*, *alleles* and *proteins* (Oliver *et al*., 2002). These conventions are only guidelines and as such do not impose restrictions to experts. Still, when a concept is named by a term that is in accordance with the provided conventions, these clues should be exploited rather than be neglected in favour of contextual clues. For example, the suffix *-ase* can be used to identify terms denoting *enzymes* with high precision. In an attempt to investigate the effects of internal term characteristics, we analysed the features of terms contained in the ontology and tried to generalize some of them into classification rules (Table 3). Table 4 summarizes the experiments performed.

Table 4. The summary of experiments performed

Experiment	Description
E1	Core (case-based reasoning)
E2	Core + extended general knowledge
E3	Core + rules
B1	Random
B2	Majority by the number of concepts
B3	Majority by the number of terms
B4	Majority by the number of term occurrences
B5	Naive Bayes
B6	Rule-based

5.6 Results

The experimental results are shown in Table 5, in which P, R and F denote precision, recall and F-measure, while GP, GR and GF stand for the corresponding graded measures. Let us first discuss the hypothesis about the knowledge contained in the ontology not being equally discriminative to all classes by comparing the results given for experiments E1 and E2. In experiment E2, an improvement has been noticed in the majority of the evaluation measures used. The most significant improvement is that of macro-averaged precision due to the precision evening out across the classes. The distribution of true positives changed because the expanded ontology helped to improve the results for certain classes, whereas they were downgraded for others. The reason for deterioration is that the classes from the old ontology were moved lower down in the new ontology with respect to the root, thus automatically appearing more similar [Formula (1)]. The new tree similarity values consequently influenced the changes in the results of approximate context matching. However, the positive impact of using the expanded ontology outbalanced the negative impact, thus resulting in a better overall performance. The expanded ontology contained 66 classes, which is $<$50% of 135 classes supported in UMLS. In addition, we did not use all terms from the 66 classes mentioned, but only those already annotated in the corpus. We conclude that the best results would be achieved with an ontology that covers all aspects of the domain. In the experiment E3, the core method has been combined with a rule-based approach exploiting internal term features. A significant improvement has been noticed in all evaluation measures used, suggesting that internal features can contribute significantly to better classification performance.

Let us now compare the results of our experiments with those achieved by the baseline methods. In the majority of cases, our method outperformed the baseline methods. The significant improvement in comparison to the random classifier suggests that our method represents a reasonably strong classification method. Similarly, our core method outperforms the 'majority' classification methods on all micro-averaged evaluation measures. As for the macro-averaged evaluation measures, the baseline methods appear to have 'better' precision. However, a classification method that assigns a fixed class to all objects of classification would always have a high macro-averaged precision when applied against a classification scheme with

Table 5. Experimental results

Experiment	Macro-averaged						Micro-averaged					
	P	*R*	*F*	GP	GR	GF	*P*	*R*	*F*	GP	GR	GF
E1	45.93	13.16	20.46	82.21	62.81	71.21	42.90	32.07	36.70	80.61	67.44	73.43
E2	60.50	19.74	29.77	86.75	66.13	75.05	43.38	32.65	37.25	81.00	67.99	73.92
E3	63.89	35.48	45.62	88.09	71.00	78.63	67.96	50.90	58.21	89.94	75.82	82.28
B1	10.75	6.85	8.37	64.19	56.77	60.25	10.60	7.77	8.97	63.99	58.03	60.86
B2	94.97	7.69	14.23	97.19	52.82	68.44	34.67	25.43	29.34	63.46	57.39	60.27
B3	92.31	7.69	14.20	97.57	58.21	72.92	0.05	0.03	0.04	68.39	60.92	64.44
B4	94.97	7.69	14.23	97.19	52.82	68.44	34.67	25.43	29.34	63.46	57.39	60.27
B5	54.22	10.31	17.33	82.29	40.40	54.19	41.95	18.12	25.31	83.11	43.14	56.80
B6	93.56	23.85	38.01	96.46	27.36	42.63	98.94	29.58	45.54	99.57	31.55	47.92

multiple classes; i.e. the class precision would be 100% for all classes other than the chosen fixed class, resulting in the average class precision getting closer to 100% with the higher number of classes. In addition, the averaged precision is even higher when the fixed class is a majority class, because its class precision would be higher. In this case, the macro-averaged precision provides misleading estimation of the quality of classification, which is made obvious by low macro-averaged recall values. In general, a reliable conclusion about the classification quality cannot be reached by looking at a single evaluation measure. Instead, as many evaluation measures as possible should be taken into account in order to provide a fuller insight.

Furthermore, the micro-averaged precision of our method is similar to that of the naive Bayes classifier. Although our method did not significantly outperform the precision of this baseline method, this fact is still taken as a positive feature of our classification approach, because it is comparable with the method which maximizes the probability of a correct prediction. However, our method significantly outperforms the recall (graded recall in particular) of the naive Bayes classifier, which results in better overall performance estimated by the *F*-measure. The only measure where the naive Bayes classifier significantly outperformed our method is the macro-averaged precision. This happened because the naive Bayes classifier concentrated on the most probable classes (in general and not for specific term occurrence alone), whereas the least probable classes were rarely suggested. Therefore, the least probable classes were seldom used to produce incorrect classifications, thus having high class precision. This reflected well on the macro-averaged precision. Again, a single evaluation measure cannot be used to fairly judge a classification method. For example, in this case, other evaluation measures imply the overall poorer quality compared with our classification method.

Finally, let us compare our approach with the rule-based method. Our method outperformed the given baseline on half of the evaluation measures. Not surprisingly, the precision of rule-based classification is extremely high. This is a general characteristic of rule-based classification. However, the opposition between precision and recall is particularly apparent in such systems. Namely, more rules typically increase recall due to higher coverage, but decrease the precision at the same time. In general, the rule-based method provides better precision, while our method provides better recall. The benefits of these two complementary features are exploited in a hybrid approach (E3), which significantly improved the precision of our CBR method, while significantly improving the recall of the rule-based method. The *F*-measure (in all four forms) for the combined method significantly enhances the *F*-measure for both methods used separately.

Based on the comparison with the six baseline methods, we conclude that our method provides a strong classification model. However, there is room for further improvement. The best results have been achieved with additional knowledge used, including the expansion of the original ontology and the use of rules generalizing internal term characteristics into the corresponding classes. The results substantiate the superiority of the combined method in comparison to the given baseline methods. However, further evaluation is needed with more diverse ontologies and large-size corpora.

6 DISCUSSION AND CONCLUSIONS

We explored the use of CBR for the burning problem of term classification in biomedicine. In particular, we described MaSTerClass as a specific implementation for this problem, which classifies individual term occurrences by learning how to locate other similar cases and extract linguistic and biomedical information necessary to perform classification from these cases. We demonstrated through a set of experiments that an effective and efficient ML approach can be developed and successfully employed for the given problem. We moved away from the existing classification approaches in several aspects. First, most of the existing approaches do not utilize high degree of biomedical and linguistic knowledge. Most often, they target specific classes by exploiting surface features (such as orthographic or lexical) typical of these classes. The main problem in such approaches is the obscurity of discriminative features. In our approach, we make use of linguistic and domain-specific features, as both are necessary for reliable classification. The linguistic knowledge is applied to acquire syntactic features of term contexts. In addition, our system efficiently utilizes explicit and extensive biomedical knowledge. While other systems may explicitly encode a certain degree of biomedical knowledge, they usually do so through a set of rules. Such knowledge representation approaches are targeted at specific tasks and classes and as such have limited generality and applicability. In our approach, the knowledge is represented by an ontology which comprises information about concepts (together with terms representing them), their classes and mutual relations. Unlike rules, ontologies can be used for various applications by both human users and computers. The effort needed to utilize an existing

biomedical ontology in our system is considerably lower than that required to engineer satisfactory classification rules, which makes our system easily portable between different tasks and subdomains.

As opposed to the existing term classification systems, rather than generalizing the background knowledge (both biomedical and linguistic) into a complex set of formal rules guiding the classification process, we opted to perform generalization as part of the classification process by relating the unclassified term occurrences together with their contexts to classified terms occurring in similar contexts. A flexible distance measure has been developed as a way of relating unclassified to relevant classified terms, which combines linguistic and domain-specific features. The flexibility of the method reflects in the fact that some features can be discarded, whereas others can be changed in an *ad hoc* manner to suit specific circumstances.

However, there is room for further improvement of the similarity measure in order to tackle the problem of discrepancy between the knowledge described in the ontology and that found in the literature. Currently, lexical similarity based on ED is used as an alternative to semantic comparison of 'unknown' terms, but it is not always appropriate, e.g. when very short terms as potential acronyms (2–4 characters) are compared with longer terms, in which case the ED would result in high values that do not reflect well the semantic similarity between the terms involved. Therefore, expanding acronyms to their full forms could improve the overall similarity for some cases. However, acronym matching need to be handled with special caution, since they are known to be highly ambiguous (e.g. AR could be expanded to any of the following terms: *androgen receptor*, *amphiregulin*, *acyclic retinoid*, *agonist-receptor*, *adrenergic receptor*, etc.). Their polysemy can be tackled by quantifying the match between an acronym and the expanded form with the probability of their match estimated from the number of possible expanded forms through acronym acquisition and term variant management (Nenadic *et al*., 2002). In addition, a more general approach to the problems caused by synonymy and polysemy will be used in future versions of the system, i.e. the latent semantic analysis will be used to infer semantic properties of terms by statistically estimating the contextual usage substitutability of terms (Deerwester *et al*., 1990).

Furthermore, the presented approach is context-sensitive and as such can readily be utilized for disambiguation of biomedical terms (e.g. to distinguish between homonymous genes and proteins they encode). In that sense, our method is more general than other term classification approaches. In our approach we classify specific occurrences rather than generic terms. Nonetheless, terms in general can still be classified by collecting classification information obtained for their occurrences. Other approaches either do not exploit the context at all (i.e. rely only on the internal term features) or process them collectively rather than focusing on a specific term occurrence and its context. The former approach cannot be generally used for disambiguation, because the appropriate interpretation of an ambiguous term can be inferred only from its context. Similarly, the latter approach cannot be used for term disambiguation unless the contexts are clustered so as to reflect specific aspects of terms used in them, which requires additional processing.

Another advantage of the MaSTerClass system is the ability to learn by storing newly solved classification problems for future use, hence gradually improving its competence. The suggested term classification approach is inductive in its nature, thus bearing strong resemblance to the human acquisition of language, who are believed not to acquire their native languages through rules, but rather to learn from examples by performing analogical reasoning. Moreover, the users are expected to embrace the CBR system more readily, largely due to the fact that similar cases readily lend an explanation for a particular choice of solution by presenting a context in which a similar solution produced satisfactory results. In particular, the validation of the classification results and their incorporation into an ontology are made easier, because the human curator can be offered an explanation by presenting the new term, its context, together with other similar terms in similar contexts.

ACKNOWLEDGEMENTS

I.S. gratefully acknowledges support from the Overseas Research Students Award Scheme (ORSAS), UK. S.A. is supported by the JISC-funded National Centre for Text Mining (NaCTeM), UK. All authors express their gratitude to Daiwa Foundation for enabling their scientific collaboration under the Daiwa Adrian Prize scheme.

REFERENCES

Aamodt,A. (1995) Knowledge acquisition and learning from experience—the role of case-specific knowledge. In Tecuci,G. and Kodratoff,Y. (eds), *Machine Learning and Knowledge Acquisition; Integrated Approaches*. Academic Press, New York, pp. 197–245.

Aha,D. (1998) The omnipresence of case-based reasoning in science and application. *Knowledge-Based Systems*, **11**, 261–273.

Collier,N. *et al*. (2001) Automatic acquisition and classification of terminology using a tagged corpus in the molecular biology domain. *J. Terminol.*, **7**, 239–257.

Collier,N. and Takeuchi,K. (2004) Comparison of character-level and part of speech features for name recognition in biomedical texts. *J. Biomed. Inform.*, **37**, 423–435.

Deerwester,S. *et al*. (1990) Indexing by latent semantic analysis. *J. Soc. Inform. Sci.*, **41**, 391–407.

Frantzi,K. and Ananiadou,S. (1999) The C-value/NC-value domain independent method for multiword term extraction. *J. Nat. Lang. Process.*, **6**, 145–180.

French,J., Powell,A. and Schulman,E. (1997) Applications of approximate word matching in information retrieval. In Golshani,F. and Makki,K. (eds), *Proceedings of the 6th International Conference on Knowledge and Information Management*, Los Angeles, CA, ACM, New York, pp. 9–15.

Fukuda,K., Tsunoda,T., Tamura,A. and Takagi,T. (1998) Toward information extraction: identifying protein. In Altman,R. Keith,D.K. and Hunter,L. (eds), *Proceedings of PSB*, Hawaii, USA, World Scientific Publishing Company, Singapore, pp. 705–716.

Gierl,L., Bull,M. and Schmidt,R. (1998) CBR in medicine. In Lenz,M. Bartsch-Spörl,B., Burkhard,H.-D. and Wess,S. (eds), *Case-Based Reasoning Technology: From Foundations to Applications*. *LNCS* **1400**, Springer-Verlag, Berlin, pp. 273–298.

Globig,C. *et al*. (1997) On case-based learnability of languages. *New Generation Comput.*, **15**, 39–83.

Gross,M. (1997) The construction of local grammars. In Roche,E. and Schabes,Y. (eds), *Finite State Language Processing*. MIT Press, CA, pp. 329–352.

Hatzivassiloglou,V. *et al*. (2001) Disambiguating proteins, genes and RNA in text: a machine learning approach. *Bioinformatics*, **1**, 97–106.

Jurisica,I. and Glasgow,J. (2004) Applications of case-based reasoning in molecular biology. *AI Magazine*, **25**, 85–95.

Kazama,J., Makino,T., Ohta,Y. and Tsujii,J. (2002) Tuning support vector machines for biomedical named entity recognition. In Johnson,S. (ed.), *Proceedings of the ACL Workshop on Natural Language Processing in the Biomedical Domain*, Philadelphia, PA, Morgan Kaufmann, pp. 1–8.

Kolodner,J. (1993) *Case-Based Reasoning*. Morgan Kaufmann.

Leake,D. (1996) Case-based reasoning: the present and future. In Leake,D. (ed), *Case-Based Reasoning: Experiences, Lessons, and Future Directions*: AAAI Press/MIT Press, CA.

Lee,K., *et al*. (2004) Biomedical named entity recognition using two-phase model based on SVMs. *J. Biomed. Inform.*, **37**, 436–447.

Macura,R.T. and Macura,K. (1997) Case-based reasoning: opportunities and applications in health care. *Artif. Intell. Med.*, **9**, 1–4.

MEDLINE (2004).

Nakagawa,H. and Mori,T. (1998) Nested collocation and compound noun for term recognition. *Proceedings of the 1st Workshop on Computational Terminology*, Montreal, Canada, pp. 64–70.

Nakagawa,H. and Mori,T. (2003) Automatic term recognition based on statistics of compound nouns and their components. *Terminology*, **9**, 201–219.

Narayanaswamy,M., Ravikumar,K.E. and Vijay-Shanker,K. (2003) A biological named entity recognizer. In Altman,R. *et al*. (eds), *Proceedings of PSB*, Hawaii, USA, World Scientific Publishing Company, Singapore, pp. 427–438.

Navarro,G. (2001) A guided tour to approximate string matching. *ACM Comput. Survey.*, **33**, 31–88.

Navarro,G., *et al*. (2000) Adding compression to block addressing inverted indexes. *Information Retrieval*, **3**, 49–77.

Nedellec,C. (2002) Bibliographical information extraction in genomics. *IEEE Intelligent Syst. Trend. Controversies*, **17**, 76–80.

Nenadic,G., Spasic,I. and Ananiadou,S. (2002) Automatic acronym acquisition and management within domain-specific texts. *Proceedings of the 3rd International Conference on Language*, Resources and Evaluation, Las Palmas, Spain, pp. 2155–2162.

Nobata,C., Collier,N. and Tsujii,J. (2000) Automatic term identification and classification in biology texts. *Proceedings of the Natural Language Pacific Rim Symposium*, Beijing, China, pp. 369–374.

Oliver,D., Rubin,D., Stuart,J., Hewett,M., Klein,T. and Altman,R. (2002) Ontology development for a pharmacogenetics knowledge base. In Altman,R. Dunker,A.K., Hunter,L. and Klein,T.E. (eds), *Proceedings of PSB*, Hawaii, USA, World Scientific Publishing Company, Singapore, pp. 65–76.

Schmidt,R. *et al*. (2001) Cased-based reasoning for medical knowledge-based systems. *Int. J. Med. Inform.*, **64**, 355–367.

Spasic,I. and Ananiadou,S. (2005) A flexible measure of contextual similarity for biomedical terms. In Altman,R. Jung,T.A., Klein,T.E., Dunker,A.K. and Hunter,L. (eds), *Proceedings of PSB*, Lihue, Hawaii, USA, World Scientific Publishing Company, Singapore, pp. 197–208.

Stapley,B., Kelley,L. and Sternberg,M. (2002) Predicting the sub-cellular location of proteins from text using support vector machines. In Altman,R. Dunker,A.K., Hunter,L. and Klein,T.E. (eds), *Proceedings of PSB*, Hawaii, USA, World Scientific Publishing Company, Singapore, pp. 374–385.

Tsuruoka,Y. and Tsujii,J. (2004) Improving the performance of dictionary-based approaches in protein name recognition. *J. Biomed. Inform.*, **37**, 461–470.

Wagner,R. and Fischer,M. (1974) The string-to-string correction problem. *J. ACM*, **21**, 168–173.

Watson,I. and Marir,F. (1994) Case-based reasoning: a review. *Knowledge Eng. Rev.*, **9**, 327–354.