Learning to Classify Biomedical Terms through Literature Mining and Genetic Algorithms

Irena Spasić[1], Goran Nenadić[2] and Sophia Ananiadou[3]

[1]Department of Chemistry, [2]Department of Computation
University of Manchester Institute of Science and Technology
PO Box 88, Manchester, M60 1QD, UK
{I.Spasic, G.Nenadic}@umist.ac.uk

[3]School of Computing, Science and Engineering
University of Salford, M5 4WT, UK
S.Ananiadou@salford.ac.uk

Abstract. We present an approach to classification of biomedical terms based on the information acquired automatically from the corpus of relevant literature. The learning phase consists of two stages: acquisition of terminologically relevant contextual patterns (CPs) and selection of classes that apply to terms used with these patterns. CPs represent a generalisation of similar term contexts in the form of regular expressions containing lexical, syntactic and terminological information. The most probable classes for the training terms co-occurring with the statistically relevant CP are learned by a genetic algorithm. Term classification is based on the learnt results. First, each term is associated with the most frequently co-occurring CP. Classes attached to such CP are initially suggested as the term's potential classes. Then, the term is finally mapped to the most similar suggested class.

1 Introduction

The biomedical literature has been rapidly expanding due to the new discoveries reported almost on a daily basis [1]. The biomedical articles are swamped by newly coined terms denoting newly identified compounds, genes, drugs, reactions, etc. The knowledge repositories need to efficiently adapt to the advent of new terms by assorting them into appropriate classes in order to allow biomedical experts to easily acquire, analyse and visualise information of interest. Due to an enormous number of terms and the complex structure of the terminology,[1] manual update approaches are inevitably inflicted by inefficiency and inconsistencies. Thus, reliable term recognition and classification methods are absolutely essential as means of support for automatic maintenance of large knowledge repositories.

Still, term classification in the biomedical domain is by no means straightforward to implement, as the naming conventions do not necessarily systematically

[1] For example, UMLS (www.nlm.nih.gov/research/umls/) contains more than 2.8 million terms assorted into 135 classes.

reflect terms' functional properties. Hence, the contexts in which terms are used need to be analysed. In most of the approaches exploiting contextual features, contexts have been typically represented using "bag-of-words" approach (e.g. [4]) or pre-defined sets of patterns (e.g. [6]). Contextual features can be used to classify terms by methods such as nearest neighbour, maximum entropy modelling, naive Bayes classification, decision trees, support vector machines, etc.

In our approach, important contextual features are learnt in the form of generalised regular expressions which describe the morpho-syntactic structure and lexico-semantic content of term contexts. The classes that are compatible with specific patterns are learnt by a genetic algorithm. The nearest neighbour algorithm (based on the similarity measure that compares lexical, syntactic and contextual features) is applied to these classes in order to perform classification.

The remainder of the paper is organised as follows. In Section 2 we describe the acquisition of contextual patterns, while Section 3 gives details on the genetic algorithm used to learn the most probable classes for terms used with these patterns. The procedure for classification of terms based on the acquired patterns and the corresponding classes is given in Section 4. Finally, in Section 5 we describe the evaluation strategy and provide the results, after which we conclude the paper.

2 Mining the Literature for Contextual Features

Our approach to the extraction of contextual features is based on automatic pattern mining, whose aim is to automatically identify, normalise and harvest the contextual patterns providing the most relevant information on the terms they surround. A contextual pattern (CP) is defined as a generalised regular expression describing the structure of a term's context. We considered two types of context constituents: morpho-syntactic (e.g. noun phrases, prepositions, etc.) and terminological (i.e. term occurrences). Term contexts are generalised by mapping the constituents to their categories. In addition, lemmatised lexical forms can be used to instantiate specific constituents in order to specify their semantic content, e.g.:

V	PREP	TERM	NP	PREP
V	PREP	TERM:*nuclear_receptor*	NP	PREP:*of*
V:*belong*	PREP:*to*	TERM:*nuclear_receptor*	NP:*superfamily*	PREP:*of*

The main challenge is to "optimise" CPs so as to provide a suitable balance between their generality and partiality towards specific classes of terms. With this in mind, the categories that are not particularly significant in providing useful contextual information (e.g. determiners, linking words, etc.) [6] can be safely removed from the CPs. On the other hand, categories with high information content (e.g. terms, verbs, etc.) need to be instantiated, because they provide good discriminatory features for term comparison. The generality of a CP is also affected by its length. While the decisions about the categories and instantiation are manually encoded based on the existing observations, the

problem of variable pattern lengths is addressed automatically. The *CP-value* measure is used to determine the statistical relevance of CPs and indirectly the appropriate length of individual CPs.[2]

First, for each term occurrence the maximal left context is extracted without crossing the sentence boundaries. The results of morpho-syntactic and terminological processing encoded in the XML-tagged corpus are used to automatically discard some categories, remove their lexical content or to keep the lemmatised lexical form (as discusses above). The remainder represents an initially selected left CP. Iterative removal of the left-most constituent until the minimal CP length[3] is reached results in a number of shorter left CPs.

If a CP does not occur nested inside other CPs, then its CP-value is proportional to its frequency and length. Otherwise, we take into account both the absolute frequency (positive impact) and the frequency of nested occurrences (small negative impact), thus measuring the frequency of its independent occurrences. Further, since a CP is more independent if it appears nested inside a larger number of different CPs, we reduce the negative impact by dividing it with the number of such CPs. Formally:

$$CP(p) = \begin{cases} \ln|p| \cdot f(p) & \text{, if } p \text{ is never nested} \\ \ln|p| \cdot \left(f(p) - \frac{1}{|T_p|}\sum_{q \in T_p} f(q)\right) & \text{, otherwise} \end{cases}$$

where $f(p)$ is the absolute frequency of the CP p, $|p|$ is its length, and T_p is a set of all CPs that contain p. CPs with high CP-values are usually general patterns, the ones with low CP-values typically are rare patterns, while the middle-ranked CPs represent relevant domain-specific patterns.[4]

3 A Genetic Algorithm for Class Selection

Given a CP, we define a *class selection* (CS) as a set of classes applicable to the majority of terms (from the training set) used in contexts described by the CP. Generally, a CS represents a hypothesis about the classes of terms complementing the corresponding CP. With that respect, each CS can be quantified by its *precision* and *recall* calculated as $P = A/(A+B)$ and $R = A/(A+C)$, where A, B and C denote the numbers of *true positives*, *false positives* and *false negatives* respectively, which are calculated as follows based on the set of training terms $\{t_1, ..., t_m\}$ co-occurring with the corresponding CP:

$$A = \sum_{i=1}^{m} |CS \cap C(t_i)| \quad B = \sum_{i=1}^{m} |CS \backslash C(t_i)| \quad C = \sum_{i=1}^{m} |C(t_i) \backslash CS|$$

[2] We will describe the way of processing the left contexts. The right contexts are treated analogously.

[3] The minimal and maximal CP length have been empirically set to two and ten respectivelly.

[4] We used the CP-value to discard 20% of the top-ranked CPs and 30% of the bottom-ranked CPs.

In the above formulas $C(t_i)$ $(i = 1, ..., m)$ denotes the set of actual classes for the term t_i. In general, the recall of a CS increases with its size as the probability of some of its classes applying to individual terms is higher, while its precision is decreasing as many of the classes would not apply to individual terms. The goal, thus, is to find a CS of an optimal size and content so as to provide suitable recall and precision. In our approach, we opted to use a genetic algorithm (GA) to learn the CSs automatically, since GAs are particularly suited for the optimisation problems [3].

GAs are meta-heuristics incorporating the principles of natural evolution and the idea of "survival of the fittest" [3]. An *individual* encodes a solution as a sequence of genes. In the initial phase of a GA a number of solutions is generated, usually at random. Selection, crossover, mutation, and replacement are applied in this order aiming to gradually improve the quality of the solutions and the possibility of finding a sufficiently good solution. *Selection* is usually defined probabilistically: the better the solution, the higher the probability for that solution to be selected as a parent. Selected individuals are recombined by applying the *crossover* between pairs of individuals. The offspring is expected to combine the good characteristics of their parents, possibly giving way to better solutions. The *mutation* operator introduces diversity into a population by modifying a solution, possibly introducing previously unseen good characteristics into the population. *Fitness function* quantifies the quality of individuals. The ones with the best fitness values *replace* less fit individuals. Once a suitable solution has been found or the number of iterations exceeds some threshold, the iterations of the GA are stopped.

In our approach, each individual (i.e. CS) is represented as a sequence of genes, where each gene denotes whether the corresponding class is a member of the CS. The goal is to optimise a CS so as to enhance its recall $R(CS)$ and precision $P(CS)$. The fitness f of a CS is calculated as follows: $f(CS) = w_R \cdot R(CS) + w_P \cdot P(CS)$, where w_R and w_P are the weights modelling the preferences towards precision and recall.[5] The objective is to find a solution with a (near)maximal fitness value. The initial population is formed by generating random individuals. We used uniform crossover: genes at each fixed position are exchanged with 50% probability. Individuals are mutated with 1% probability by a randomly changing a randomly chosen gene.

4 Term Classification

Let $CP = \{cp_1, ..., cp_n\}$ be a set of automatically extracted CPs. During the phase of learning the CSs, each cp_i $(i = 1, ..., n)$ is associated with a class selection it may be complemented with, $CS_i = \{c_{i,1}, ..., c_{i,m_i}\}$. Each CS typically contains multiple classes. In order to link a term to a specific class, we score each class by a hybrid term similarity measure, called the *CLS measure*, which combines contextual, lexical and syntactic properties of terms [7]. This measure, however, applies to terms, while we need to compare *terms* to *classes*. We,

[5] In our experiments we used equal weights for precision and recall.

therefore, set the similarity between a term t and a class $c_{i,j}$ $(j = 1, ..., m_i)$ to be proportional to the average similarity between the term and the terms from the given class. More formally, if $e_1, ..., e_k$ are randomly chosen terms[6] from that class that occur in the corpus, then term-to-class similarity is calculated in the following way:

$$S(t, c_{i,j}) = \frac{\frac{1}{k}\sum_{i=1}^{k} CLS(t, e_i)}{\sqrt{\sum_{i=1}^{k} CLS^2(t, e_i)}}$$

Note that the described method implicitly incorporates the class probability factors. The more frequently a certain class complements the given CP, the more likely it will be present in the corresponding CS.

5 Experiments and Evaluation

We performed the classification experiments on a corpus of 2072 abstracts retrieved from MEDLINE (www.ncbi.nlm.nih.gov/PubMed/). The corpus was terminologically processed by using the C/NC-value method for term recognition [2]. Terms were tagged with the classification information obtained from the UMLS ontology. We focused on a subtree describing chemical substances (13 classes). Terms from this hierarchy were used as part of the training (1618 terms) and testing (138 terms) sets. A total of 1250 CPs have been automatically extracted from the corpus. We conducted experiments with 631 most relevant CPs (the ones most frequently co-occurring with terms). Based on the training set, CPs were associated with the CSs, e.g. the pattern V:*activate* PREP:*by* TERM was associated with the following CS: {*immunologic factor, receptor, enzyme, hormone, pharmacologic substance*}.

Each testing term was associated with the CP it most frequently co-occurred with. The CS learnt for that CP was used to classify the term in question. For example, the term *ciprofibrate*, most frequently occurring with the above CP, was correctly classified as a *pharmacologic substance*. Table 1 summarises the classification results achieved by our method and compares them to three baseline methods, which include random, majority and naive Bayes classification. The baseline methods map a term respectively to (1) a random class, (2) a class with the highest number of term occurrences in the corpus, and (3) the most probable class based on the words found in its context.

Table 1: Evaluation of the classification results.[7]

Method	Precision	Recall	F-measure
CP/CS	61%	38%	47%
random	11%	8%	9%
majority	35%	25%	29%
naive Bayes	42%	18%	25%

[6] We preselected ten terms for each class.

[7] F-measure is calculated according to the following formula: $F = 2 \cdot P \cdot R/(P + R)$, where precision P and recall R are calculated as before.

A comparison to other methods reported in the literature would require the results obtained on the same set. Most of these methods were either unavailable or designed for specific classes (e.g. [5]). Nonetheless, most of the results were reported for fewer number of classes, while the probability of missing a correct class increases with the higher number of classes available. It is then natural for the performance measures to provide "poorer" values when tested on broader classification schemes. For example, our method achieved 61% precision for 13 classes, while Hatzivassiloglou et al. [5] achieved 67% for three classes.

6 Conclusion

We presented a term classification method, which makes use of the structured contextual information acquired automatically through contextual pattern mining. The presented term classification approach revolves around these patterns. Namely, they are used to collect unclassified terms and to suggest their potential classes based on the classes automatically predicted by a genetic algorithm, which optimises precision and recall estimated on the set of training terms. The suggested classes are compared through their terms to the given term by a similarity measure, which takes into account lexical, syntactic and contextual information.

Note that terms can be compared directly to all classes in the classification scheme in order to perform the nearest neighbour classification directly. However, the class pruning approach has been adopted in order to enhance the computational efficiency of the classification process itself. In this approach, the contextual and classification information learned offline needs to be updated periodically in order to reflect the changes in the corpus of literature and the information available in the ontology.

The precision of our method could be improved by analysing and exploiting orthographic and lexical term features characteristic of specific classes (e.g. suffix *-ase* for the class of enzymes). On the other side, the recall could be improved by taking into account more context patterns and by using larger corpora.

References

1. Blaschke, C., Hirschman, L., Valencia, A.: Information Extraction in Molecular Biology. Briefings in Bioinformatics 3/2 (2002) 154-165
2. Frantzi, K., Ananiadou, S., Mima, H.: Automatic Recognition of Multi-Word Terms. Int. J. on Digital Libraries 3/2 (2000) 117-132
3. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley (1989) 432
4. Grefenstette, G. Exploration in Automatic Thesaurus Discovery. Kluwer Academic Publishers (1994)
5. Hatzivassiloglou, V., Duboue, P., Rzhetsky, A.: Disambiguating Proteins, Genes, and RNA in Text: A Machine Learning Approach. Bioinformatics 1/1 (2001) 1-10
6. Maynard, D., Ananiadou, S.: Identifying Terms by Their Family and Friends. Proceedings of Coling 2000, Luxembourg (2000) 530-536
7. Nenadić, G., Spasić, I., Ananiadou, S.: Mining Term Similarities from Corpora. Terminology 10/1, (2004) 55-80