Selecting Text Features for Gene Name Classification: from Documents to Terms

Goran Nenadić[1,2], Simon Rice[2], Irena Spasić[3], Sophia Ananiadou[3], Benjamin Stapley[2]

[1]Dept. of Computation
UMIST
Manchester, M60 1QD

[2]Dept. of BioMolecular Sciences
UMIST
Manchester, M60 1QD

[3]Computer Science
University of Salford
Salford, M5 4WT

Abstract

In this paper we discuss the performance of a text-based classification approach by comparing different types of features. We consider the automatic classification of gene names from the molecular biology literature, by using a support-vector machine method. Classification features range from words, lemmas and stems, to automatically extracted terms. Also, simple co-occurrences of genes within documents are considered. The preliminary experiments performed on a set of 3,000 *S. cerevisiae* gene names and 53,000 Medline abstracts have shown that using domain-specific terms can improve the performance compared to the standard bag-of-words approach, in particular for genes classified with higher confidence, and for under-represented classes.

1 Introduction

Dynamic development and new discoveries in the domain of biomedicine have resulted in the huge volume of the domain literature, which is constantly expanding both in the size and thematic coverage (Blaschke et al., 2002). The literature, which is still the most relevant and the most useful knowledge source, is swamped by newly coined terms and relationships representing and linking newly identified or created compounds, genes, drugs, reactions, etc., which makes the existing terminological resources rarely up-to-date. Therefore, domain knowledge sources need to frequently adapt to the advent of such terms by assorting them into appropriate classes, in order to allow biologists to rapidly acquire, analyse and visualise entities or group of entities (Stapley et al., 2002).

Naming conventions solely cannot be used as reliable classification criteria, since they typically do not systematically reflect any particular functional property or relatedness between biological entities. On the other hand, it has proved surprisingly difficult to automatically predict classes for some types of biological entities based solely on experimental data (e.g. the prediction of protein cellular locations from sequences (Eisenhaber and Bork, 1998) or the amino acid composition of proteins (Nishikawa and Ooi, 1982)).

In order to overcome this problem, several literature-based classification methods have been developed (Collier et al. 2001; Hatzivassiloglou et al., 2001). Classification methods typically rely on supervised machine learning techniques that examine the wider context in which terms are used. For example, Raychaudhuri et al. (2002) used document-based word counts and naive Bayesian classification, maximum entropy modelling and nearest-neighbour classification to assign the GO ontology codes to a set of genes. Recently, support-vector machines (SVMs, (Vapnik, 1995)) have been widely used as fast, effective and reliable means for text-based classification, both for document classification (Joachims, 1998) and classification of specific named entities (Stapley et al., 2002; Kazama et al., 2002).

Regardless of the learning approach and target entities (documents or terms), different types of text features have been employed for the classification task. For example, a bag-of-words approach was used by Stapley et al. (2002) to classify proteins, while Collier et al. (2001) used orthographic features to classify different biological entities. On the other hand, Hatzivassiloglou et al. (2001) experimented with morphological, distributional and shallow-syntactic information to discriminate between proteins, genes and RNAs.

In this paper we analyse the impact of different types of features on the performance of an SVM-based classifier. More precisely, we discuss the multi-class SVM performance with respect to the type of features used, ranging from document identifiers, through words, lemmas and stems, to automatically extracted terms.

The paper is organised as follows. After presenting the related work on feature selection in Section 2, the methods used for engineering features in our approach are explained in Section 3. Section 4 discusses the experiments and results.

2 Related work

An SVM is a binary classification method that combines statistical learning and optimisation techniques with kernel mapping (Vapnik, 1995). The main idea of the method is to automatically learn a separation hyperplane from a set of training examples, which splits classified entities into two subsets according to a certain classification property. The optimisation part is used to maximise the distance (called the margin) of each of the two subsets from the hyperplane.

The SVM approach has been used for different classification tasks quite successfully, in particular for document classification, where the method outperformed many alternative approaches (Joachims, 1998). Similarly, SVMs have been used for term classification. For example, a bag-of-simple-words approach with *idf*-like weights was used to learn a multi-class SVM classifier for protein cellular location classification (Stapley et al., 2002). Proteins were represented by feature vectors consisting of simple words co-occurring with them in a set of relevant Medline abstracts. The precision of the method was better than that of a classification method based on experimental data, and similar to a rule-based classifier.

Unlike many other classification methods that have difficulties coping with huge dimensions, one of the main advantages of the SVM approach is that its performance does not depend on the dimensionality of the space where the hyperplane separation takes place. This fact has been exploited in the way that many authors have suggested that “there are few irrelevant features” and that “SVMs eliminate the need for feature selection” (Joachims, 1998). It has been shown that even the removal of stop-words is not necessary (Leopold and Kindermann, 2002).

Few approaches have been undertaken only recently to tune the original SVM approach by selecting different features, or by using different feature weights and kernels, mostly for the document classification task. For example, Leopold and Kindermann (2002) have discussed the impact of different feature weights on the performance of SVMs in the case of document classification in English and German. They have reported that an entropy-like weight was generally performing better than *idf*, in particular for larger documents. Also, they suggested that, if using single words as features, the lemmatisation was not necessary, as it had no significant impact on the performance.

Lodhi et al. (2002) have experimented with different kernels for document classification. They have shown that a string kernel (which generates all sub-sequences of a certain number of characters) could be an effective alternative to linear kernel SVMs, in particular in the sense of efficiency.

In the case of term classification, Kazama et al. (2002) used a more exhaustive feature set containing lexical information, POS tags, affixes and their combinations in order to recognise and classify terms into a set of general biological classes used within the GENIA project (GENIA, 2003). They investigated the influence of these features on the performance. For example, they claimed that suffix information was helpful, while POS and prefix features did not have clear or stable influence.

While each of these studies used some kind of orthographical and/or lexical indicators to generate relevant features, we wanted to investigate the usage of semantic indicators (such as domain-specific terms) as classification features, and to compare their performance with the classic lexically-based features.

3 Feature selection and engineering

The main aim while selecting classification features is to find (and use) textual attributes that can improve the classification accuracy and accelerate the learning phase. In our experiments we examined the impact of different types of features on the performance of an SVM-based gene name classification task. The main objective was to investigate whether additional linguistic pre-processing of documents could improve the SVM results, and, in particular, whether semantic processing (such as terminological analysis) was beneficial for the classification task. In other words, we wanted to see which textual units should be generated as input feature vectors, and what level of pre-processing was appropriate in order to produce more accurate predictions.

We have experimented with two types of textual features: in the first case, we have used a classic bag-of-single-words approach, with different levels of lexical pre-processing (i.e. single words, lemmas, and stems). In the second case, features related to semantic pre-processing of documents have been generated: a set of automatically extracted multi-word terms (other than gene names to be classified) has been used as a feature set. Additionally, we have experimented with features reflecting simple gene-gene co-occurrences within the same documents.

3.1 Single words as features

The first set of experiments included a classic bag-of-single-words approach. All abstracts (from a larger collection, see Section 4) that contained at least one occurrence of a given gene or its aliases have been selected as documents relevant for that gene. These documents have been treated as a single virtual document pertinent to the given gene. All words co-occurring with a given gene in any of the abstracts were used as its features.

A word has been defined as an alphanumeric sequence between two standard separators, with all numeric expressions that were not part of other words filtered out. In addition, a standard list of around 300 stop-words has been used to exclude some frequent non-content words.

An *idf*-like measure has been used for feature weights: the weight of a word *w* for gene *g* is given by

$$\log \frac{1+\sum_{j\in R_g} f_j(w)}{N_w(1+|R_g|)} \tag{1}$$

where R_g is a set of relevant documents for the gene *g*, $f_j(w)$ is the frequency of *w* in document *j*, and N_w is the global frequency of *w*. Gene vectors, containing weights for all co-occurring words, have been used as input for the SVM.

It is widely accepted that rare words do not have any significant influence on accuracy (cf. (Leopold and Kindermann, 2002)), neither do words appearing only in few documents. In our experiments (demonstrated in Section 4), we compared the performance between the 'all-words approach' and an approach featuring words appearing in at least two documents. In the latter case, the dimension of the problem (expressed as the number of features) was significantly reduced (with factor 3), and consequently the training time was shortened (see Section 4).

Since many authors claimed that the biomedical literature contained considerably more linguistic variations than text in general (cf. Yakushiji et al., 2001), we applied two standard transformations in order to reduce the level of lexical variability. In the first case, we used the EngCG POS tagger (Voutilainen and Heikkila, 1993) to generate lemmas, so that lemmatised words were used as features, while, in the second case, we generated stems by the Porter's algorithm (Porter, 1980). Analogously to words, the same *idf*-based measure was used for weights, and experiments were also performed with all features and with the features appearing in no less than two documents.

3.2 Terms as features

Many literature-mining techniques rely heavily on the identification of main concepts, linguistically represented by domain specific terms (Nenadic et al., 2002b). Terms represent the most important concepts in a domain and have been used to characterise documents semantically (Maynard and Ananiadou, 2002). Since terms are semantic indicators used in scientific discourse, we hypothesised that they might be useful classification features.

The high neology rate for terms makes existing glossaries incomplete for active and time-limited research, and thus automatic term extraction tools are needed for efficient terminological processing. In order to automatically generate term as features, we have used an enhanced version of the C-value method (Frantzi et al., 2000), which assigns termhoods to automatically extracted multi-word term candidates. The method combines linguistic formation patterns and statistical analysis. The linguistic part includes part-of-speech tagging, syntactic pattern matching and the use of a stop list to eliminate frequent non-terms, while statistical termhoods amalgamate four numerical characteristic of a candidate term, namely: the frequency of occurrence, the frequency of occurrence as a nested element, the number of candidate terms containing it as a nested element, and term's length.

Due to the extensive term variability in the domain, the same concept may be designated by more than one term. Therefore, term variants conflation rules have been added to the linguistic part of the C-value method, in order to enhance the results of the statistical part. When term variants are processed separately by the statistical module, their termhoods are distributed across different variants providing separate frequencies for individual variants instead of a single frequency calculated for a term candidate unifying all of its variants. Hence, in order to make the most of the statistical part of the C-value method, all variants of the candidate terms are matched to their normalised forms by applying rule-based transformations and treated jointly as a term candidate (Nenadic et al., 2002a). In addition, acronyms are acquired prior to the selection of the term candidates and also mapped to their expanded forms, which are normalised in the same manner as other term candidates.

Once a corpus has been terminologically processed, each target gene is assigned a set of terms appearing in the corresponding set of documents relevant to the given gene. Thus, in this case, gene vectors used in the SVM classifier contain co-occurring terms, rather than single words. As term weights, we have used a formula analogous to (1). Also, similarly to single-word features, we have experimented with terms appearing in at least two documents.

3.3 Combining word and term features

The C-value method extracts only multi-word terms, which may be enriched during the normalisation process with some single-word terms, sourcing from e.g. acronyms or orthographic variations. In order to assess impact of both single and multi-word terms as features, we experimented with combining single-word based features with multi-word terms by using a simple kernel modification that concatenates the corresponding feature vectors. Thus, gene vectors used in this case contain both words and terms that genes co-occur with.

3.4 Document identifiers as features

Term co-occurrences have been traditionally used as an indication of their similarity (Ushioda, 1986), with documents considered as bags of words in the majority of approaches. For example, Stapley et al. (2000) used document co-occurrence statistics of gene names in Medline abstracts to predict their connections. The co-occurrence statistics were represented by the reciprocal Dice coefficient. Similar approach has been undertaken by Jenssen et al. (2001): they identified co-occurrences of gene names within abstracts, and assigned weights to their "relationship" based on frequency of co-occurrence.

In our experiments, abstract identifiers (PubMed identifiers, PMIDs) have been used as features for classification, where the dimensionality of the feature space was equal to the number of documents in the document set. As feature weights, binary values (i.e. a gene is present/absent in a document) were used.

We would like to point out that – contrary to other features – this approach is not a general learning approach, as document identifiers are not classification attributes that can be learnt and used against other corpora. Instead, this approach can be only used to classify new terms that appear in a closed corpus used for training.

4 Experiments and discussions

An experimental environment was set up by using the following resources:

a) corpus: a set of documents has been obtained by collecting Medline abstracts (NLM, 2003) related to the baker's yeast (*S. cerevisiae*), resulting in 52,845 abstracts; this set, containing almost 5 million word occurrences, was used as both training and testing corpus.

b) classification entities: a set of 5007 *S. cerevisiae* gene names has been retrieved from the SGD (Saccharomyces Genome Database) gene registry[1], which also provided synonyms and aliases of genes; 2975 gene names appearing in the corpus have been used for the classification task.

c) classification scheme: each gene name has been classified according to a classification scheme based on eleven categories (see Table 1) of the up-

[1] http://genome-www.stanford.edu/Saccharomyces/registry.html

per part of the GO ontology (Ashburner et al., 2000)[2].

d) training and testing sets: positive examples for each class were split evenly between the training and testing sets, and, also, the number of negative examples in the training set was set equal to the number of positive examples within each class. The only exception was the *metabolism* class, which had far more positive than negatives examples. Therefore, in this case, we have evenly split negative examples between the training and testing sets. Table 1 presents the distribution of positive and negative examples for each class.

d) SVM engine: for training the multi-class SVM, we used SVM Light package v3.50 (Joachims, 1998) with a linear kernel function with the regulation parameter calculated as $avg(<x,x>)^{-1}$.

Category (GO code)	examples		
	training	**testing 1**	**testing 2**
autophagy (GO:0006914)	12/12	11/2940	11/11
cell organisation (GO:0016043)	379/379	378/1839	378/378
cell cycle (GO:0007049)	226/226	225/2298	225/225
intracellular protein transport (GO:0006886)	135/135	134/2571	134/134
ion homeostasis (GO:0006873)	37/37	37/2864	37/37
meiosis (GO:0007126)	45/45	44/2841	44/44
metabolism (GO:0008152)	1118/370	1117/370	370/370
signal transduction (GO:0007165)	68/68	68/2771	68/68
sporulation (*sc*) (GO:0007151)	27/27	27/2894	27/27
response to stress (GO:0006950)	91/91	91/2702	91/91
transport (GO:0006810)	284/284	284/2123	284/284

Table 1. Classification categories and the number of examples in the training and the testing sets

Features have been generated according to the methods explained in Section 3 (Table 2 shows the number of features generated). As indicated earlier, the experiments have been performed by using either all features or by selecting only those that appeared in at least two documents. As a rule, there were no significant differences in the classification performance between the two.

feature	**no. of all features**	**no. of features appearing in >1 docs**
words	160k	60k
lemmas	150k	54k
stems	140k	50k
terms	127k	62k

Table 2. The number of features generated

To evaluate the classification performance we have firstly generated precision/recall plots for each class. In the majority of classes, terms have demonstrated the best performance (cf. Figures 1 and 2). However, the results have shown a wide disparity in performance across the classes, depending on the size of the training set. The classes with fairly large number of training entities (e.g. *metabolism*) have been predicted quite accurately (regardless of the features used), while, on the other hand, under-represented classes (e.g. *sporulation*) performed quite modestly (cf. Figure 1).

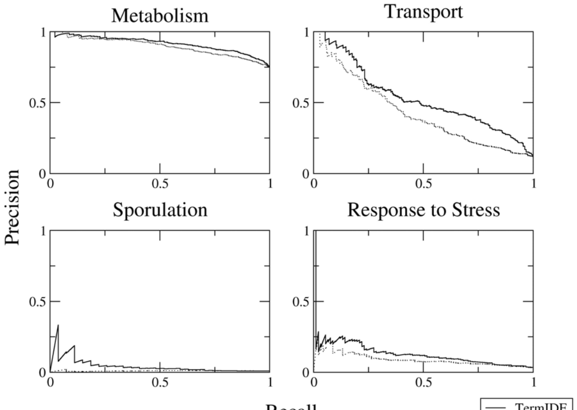

Figure 1. Precision/recall plots for some classes using words and terms

Comparison between performances on different classes is difficult if the classes contain fairly different ratios of positive/negative examples in the

[2] The January 2003 release of the GO ontology was used. A similar classification scheme was used in (Raychaudhuri et al., 2002).

testing sets, as it was the case in our experiments (see Table 1, column testing 1). Therefore, we re-evaluated the results by selecting – for each class – the same number of positive and negative examples (see Table 1, column testing 2), so that we could compare relative performance across classes. The results shown in Figure 2 actually indicate which classes are "easier" to learn (only the performance of single-words and terms are presented).

To assess the global performance of classification methods, we employed micro-averaging of the precision/recall data presented in Figure 2. In micro-averaging (Yang, 1997), the precision and recall are averaged over the number of entities that are classified (giving, thus, an equal weight to the performance on each gene). In other words, micro-average shows the performance of the classification system on a gene selected randomly from the testing set.

The comparison of micro-averaging results for words, lemmas and stems has shown that there was no significant difference among them. This outcome matches the results previously reported for the document classification task (Leopold and Kindermann, 2002), which means that there is no need to pre-process documents.

Figure 3 shows the comparison of micro-averaging plots for terms and lemmas. Terms perform generally much better at lower recall points, while there is just marginal difference between the two at the higher recall points. Very high precision points at lower recall mean that terms may be useful classification features for precise predictions for genes classified with the highest confidence.

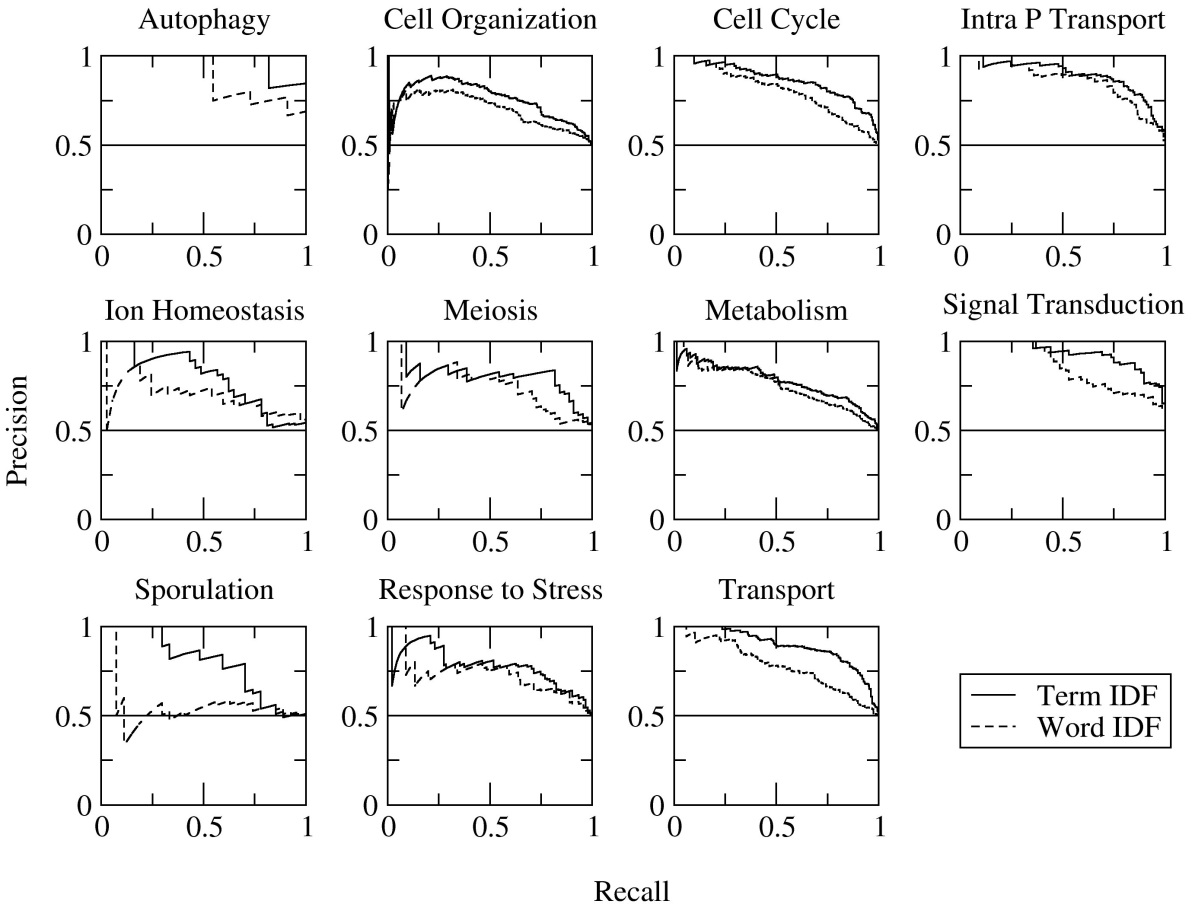

Figure 2. Precision/recall plots for the 11 classes using words and terms (horizontal lines indicate the performance of a random classifier)

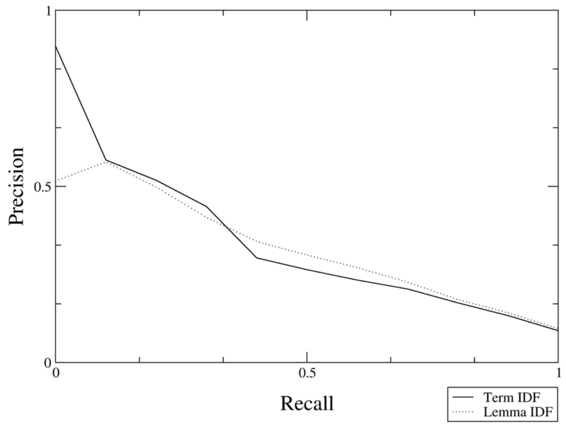

Figure 3. Micro-averaging plot for 11 classes using lemmas and terms

The results obtained by combining terms and words have not shown any improvements over using only terms as classification features. We believe that adding more features has introduced additional noise that derogated the overall performance of terms.

Finally, Figure 4 presents the comparison of classification results using terms and abstract identifiers. Although PMIDs outperformed terms, we reiterate that – while other features allow learning more general properties that can be applied on other corpora – PMIDs can be only used to classify new terms that appear in a closed training/testing corpus.

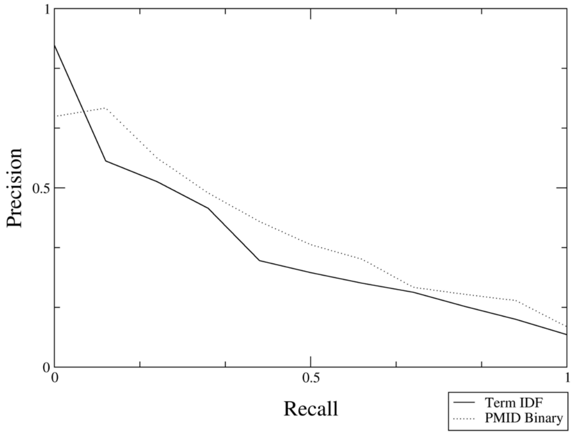

Figure 4. Micro-averaging plot for 11 classes using PMIDs and terms

5 Conclusion

Due to an enormous number of terms and the complex and inconsistent structure of the biomedical terminology, manual update of knowledge repositories are prone to be both inefficient and inconsistent (Nenadic et al., 2002b; Stapley et al., 2002). Therefore, automatic text-based classification of biological entities (such as gene and protein names) is essential for efficient knowledge management and systematic approach that can cope with huge volume of the biomedical literature. Furthermore, classified terms irrefutably have a positive impact on improving the results of IE/IR, knowledge acquisition, document classification and terminology management (Blaschke et al., 2002).

In this paper we have examined the procedures for engineering text-based features at various levels of linguistic pre-processing, and considered their impacts on the performance of an SVM-based gene name classifier. The experiments have shown that simple linguistic pre-processing (such as lemmatisation and stemming) does not have significant influence on the performance, i.e. there is no need to pre-process documents. Also, reducing the feature space by selecting only features that appear in more documents does not result in decrease of the performance, but can significantly reduce the time needed for training. PMID-based classification has shown very good performance, but a PMID-based classifier can be applied only on the training set of documents.

The experiments have also shown that using semantic indicators (represented by dynamically extracted domain-specific terms) can improve the performance compared to the standard bag-of-words approach, in particular at lower recall points, and for rare classes. This means that terms can be used as reliable features for classifying genes with higher confidence, and for under-represented classes. However, terminological analysis requires considerable pre-processing time.

Our further research will focus on generating the biological interpretation and justification of the classification results by using terms (that have been used as key distinguishing features for classification) as semantic indicators of the corresponding classes.

References

M. Ashburner, et al.. 2000. *Gene Ontology: Tool for the Unification of Biology*. Nature, 25:25-29.

C. Blaschke, L. Hirschman and A. Valencia. 2002. *Information Extraction in Molecular Biology*. Briefings in Bioinformatics, 3(2):154-165.

N. Collier, C. Nobata and J. Tsujii. 2001. *Automatic Acquisition and Classification of Terminology Using a Tagged Corpus in the Molecular Biology Domain*. Journal of Terminology, John Benjamins.

F. Eisenhaber and P. Bork. 1998. *Wanted: Subcellular Localization of Proteins Based on Sequences*. Trends Cell Biology, 8(4):169-170.

K. Frantzi, S. Ananiadou and H. Mima. 2000. *Automatic Recognition of Multi-Word Terms: the C-value/NC-value Method*. International Journal on Digital Libraries 3(2):115-130.

GENIA project. 2003. *GENIA resources.* Available at: http://www-tsujii.is.s.u-tokyo.ac.jp/~genia/

V. Hatzivassiloglou, P. Duboue and A. Rzhetsky. 2001. *Disambiguating Proteins, Genes, and RNA in Text: A Machine Learning Approach*. Bioinformatics, 1(1):1-10.

T. Jenssen, A. Laegreid, J. Komorowski and E. Hovig. 2001. *A literature Network of Human Genes for High-throughput Analysis of Gene Expressions*. Nature Genetics, 28: 21-28.

T. Joachims. 1998. *Text Categorization with Support Vector Machines: Learning Many Relevant Features*. Proceedings of 10th European Conference on Machine Learning, Springer-Verlag, Heidelberg, 137-142.

J. Kazama, T. Makino, Y. Ohta and J. Tsujii. 2002. *Tuning Support Vector Machines for Biomedical Named Entity Recognition*. Proceedings of the Workshop NLP in Biomedicine, ACL 2002.

E. Leopold and J. Kindermann. 2002. *Text Categorization with Support Vector Machines. How to Represent Texts in Input Space?* Machine Learning, 46:423-444.

H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini and C. Watkins. 2002. *Text Classification using String Kernels*. Journal of Machine Learning Research, 2:419-444.

D. Maynard and S. Ananiadou. 2000. *Identifying Terms by their Family and Friends*. Proceedings of COLING 2000, Saarbrucken, Germany, 530-536.

K. Nishikawa and T. Ooi. 1982. *Correlation of the Amino Acid Composition of a Protein to its Structural and Biological Characters.* Journal of Biochemistry (Tokyo), 91(5):1281-1824.

G. Nenadic, I. Spasic and S. Ananiadou. 2002a. *Automatic Acronym Acquisition and Term Variation Management within Domain-Specific Texts*. Proceedings of LREC-3, Las Palmas, Spain, 2155-2162.

G. Nenadic, H. Mima, I. Spasic, S. Ananiadou and J. Tsujii. 2002b. *Terminology-based Literature Mining and Knowledge Acquisition in Biomedicine*. International Journal of Medical Informatics, 67(1-3):33-48.

NLM, National Library of Medicine. 2003. *Medline*. Available at http://www.ncbi.nlm.nih.gov/PubMed/

M. Porter. 1980: *An Algorithm for Suffix Stripping*. Program, 14(1):130-137.

S. Raychaudhuri, J. Chang, P. Sutphin and R. Altman. 2002. *Associating Genes with Gene Ontology Codes Using a Maximum Entropy Analysis of Biomedical Literature*. Genome Research, 12:203-214.

B. Stapley and G. Benoit. 2000. *Bibliometrics: Information Retrieval and Visualization from Co-occurrence of Gene Names in Medline Abstracts*. Proceedings of the Pacific Symposium on Bio-computing, PSB 2000

B. Stapley, L. Kelley and M. Sternberg. 2002. *Predicting the Sub-Cellular Location of Proteins from Text Using Support Vector Machines*. Proceedings of the Pacific Symposium on Bio-computing, PSB 2002.

A. Ushioda. 1996. *Hierarchical Clustering of Words*. Proceedings of *COLING 96*.

V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer Verlag, Heidelberg.

A. Voutilainen and J. Heikkila. 1993. *An English Constraint Grammar (ENGCG) a Surface-Syntactic Parser of English*. In Fries, U et al. (Eds.): Creating and Using English Language Corpora, Rodopi, Amsterdam/Atlanta, 189-199.

A. Yakushiji, Y. Tateisi, Y. Miyao and J. Tsujii. 2001. *Event Extraction From Biomedical Papers Using a Full Parser*. Proceedings PSB 2001, Hawaii, USA, 408-419.

Y. Yang. 1997. *An Evaluation of Statistical Approaches to Text Categorization.* Information Retrieval, 1(1/2):69-90.