Identifying Relevant Full-text Articles for Database Curation

Chih Lee, Wen-Juan Hou and Hsin-Hsi Chen

Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan, 106

E-mail: {clee, wjhou}@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

In this paper, we propose an approach for identifying curatable articles from a large pool. Our system currently considers three parts of an article as three individual representations of the article, and utilizes two domain-specific resources to reveal the deep knowledge contained in the article in order to generate more representations of the article. Cross-validation is employed to find the best combination of representations and an SVM classifier is trained out of this combination. The cross-validation results and results of the official runs are listed. The experimental results show overall high performance.

1 Introduction

Organism Database plays a crucial role in genomic and proteomic research. It stores the up-to-date profile of each gene of the species interested. For example, the Mouse Genome Database (MGD) provides essential integration of experimental knowledge for the mouse system with information annotated from both literature and online sources (Bult *et al.*, 2004). The Mouse Gene Expression Database (GXD) provides information about expression profiles in different mouse strains and mutants (Hill *et al.*, 2004). The Tumor Gene Database (TGDB) (http://www.tumor-gene.org/TGDB/tgdb.html) provides a standard set of facts (e.g., protein size, biochemical activity, chromosomal location, *etc*.) about all known cancer-causing mutations; proto-oncogenes and tumor suppressor genes. FlyBase (http://flybase.org) is a database for information on the genetics and molecular biology of the insect family Drosophila (Drysdale *et al*., 2005). To provide biomedical scientists with easy access to complete and accurate information, curators have to constantly update databases with new information. Published literature written in natural languages has long been the main source of information because of its high accuracy. With the rapidly growing rate of publication, it is impossible for database curators to read every published article. However, since current fully-automated curation systems have not met the strict requirement of high accuracy and recall, database curators still have to read some (if not all) of the articles sent to them. Therefore, the classification of biological literature is an important research topic. It will be very helpful if a triage system is able to correctly identify the curatable or relevant articles in a large number of biological articles.

Text classification systems are originally designed to classify documents to different categories. Most approaches to text classification are based on statistical natural language processing (Manning and Schutze, 1999). In the past, text classification mainly focused on general domains. Recently, several attempts have been made to classify documents from biomedical domain (Hirschman *et al*., 2002). When classifying biological articles, statistical classification systems usually need a training set of documents, e.g. MEDLINE[1], in order to build a classification model. Couto *et al*. (2004) used the information extracted from related web resources to classify biomedical literature. They validated it by testing it on the KDD 2002 Cup challenge: bio-text task (Yeh *et al*., 2002). Hou *et al*. (2005) used the reference corpus to help classifying gene annotation. They generated the

[1] PubMed database at the National Library of Medicine
http://www.ncbi.nih.gov/PubMed

predicted Gene Ontology terms (Camon *et al*., 2003). The extraction of keywords related to classification documents is also helpful with classification tasks. For example, Andrade and Valencia (1998) and Shatkey *et al*. (2000) mainly detected words related to function. Pouliot *et al*. (2001), Xie *et al*. (2002), Lee *et al*. (2004) and Perez *et al*. (2004) found keywords related to GO terms. It shows that many researchers are interested in biological text classification problems (Hersh *et al*., 2004).

In this paper, we propose an approach for identifying curatable articles from a large pool. The rest of this paper is organized as follows. Section 2 presents the overview of our system architecture. In Section 3 we describe our methods in detail. The results achieved by the proposed methods are shown and discussed in Section 4. Finally, we express our main conclusions in Section 5.

2 Architecture Overview

Figure 1 shows the overall architecture of our system for the categorization task. For each full-text training article, we first preprocess the article and extract several parts from it. Each of the extracted parts is considered a representation of this article. In this task, we considers three parts of an article, which are (1) title and abstract, (2) MeSH terms assigned to this article and (3) figure and table captions. With the help of domain-specific knowledge, we process the three parts and obtain more representations of an article, while the original three representations are kept. The three original representations are denoted as **Abstract**, **MeSH** and **Caption** in the rest of this paper. In the model selection phase, we perform feature ranking on each representation of an article and employ cross-validation to decide the number of features to be kept. Moreover, we use cross-validation to obtain the best combination of all the representations. Finally, a support vector machine (SVM) (Vapnik, 1995; Hsu *et al.*, 2003) classifier is obtained.

3 Methods

3.1 Document Preprocessing

In the preprocessing phase, we perform acronym expansion on the articles, remove the remaining tags from the articles and extract three parts of interest from each article. Abbreviations are often used to replace long terms in writing articles, but it is possible that several long terms share the same short form, especially for gene/protein names. To avoid ambiguity and enhance clarity, the acronym expansion operation replaces every tagged abbreviation with its long form followed by itself in a pair of parentheses. An example of this operation is shown in Figure 2, where a tagged abbreviation "IP_3" will be replaced with "inositol trisphosphase (IP_3)".

3.2 Using Domain-Specific Knowledge

With the help of domain-specific knowledge, we can extract the deep knowledge from a piece of text written in natural language. For example, with a gene name dictionary, we can identify the gene names contained in an article. Moreover, by further consulting organism databases, we can get the properties of the genes occurred in the article. Two domain-specific resources are exploited in this task. One is the Unified Medical Language System (UMLS) (Humphreys *et al.*, 1998) and the other is a list of tumor names obtained from Mouse Tumor Biology Database (MTB)[2].

[2] http://tumor.informatics.jax.org/mtbwi/tumorSearch.do

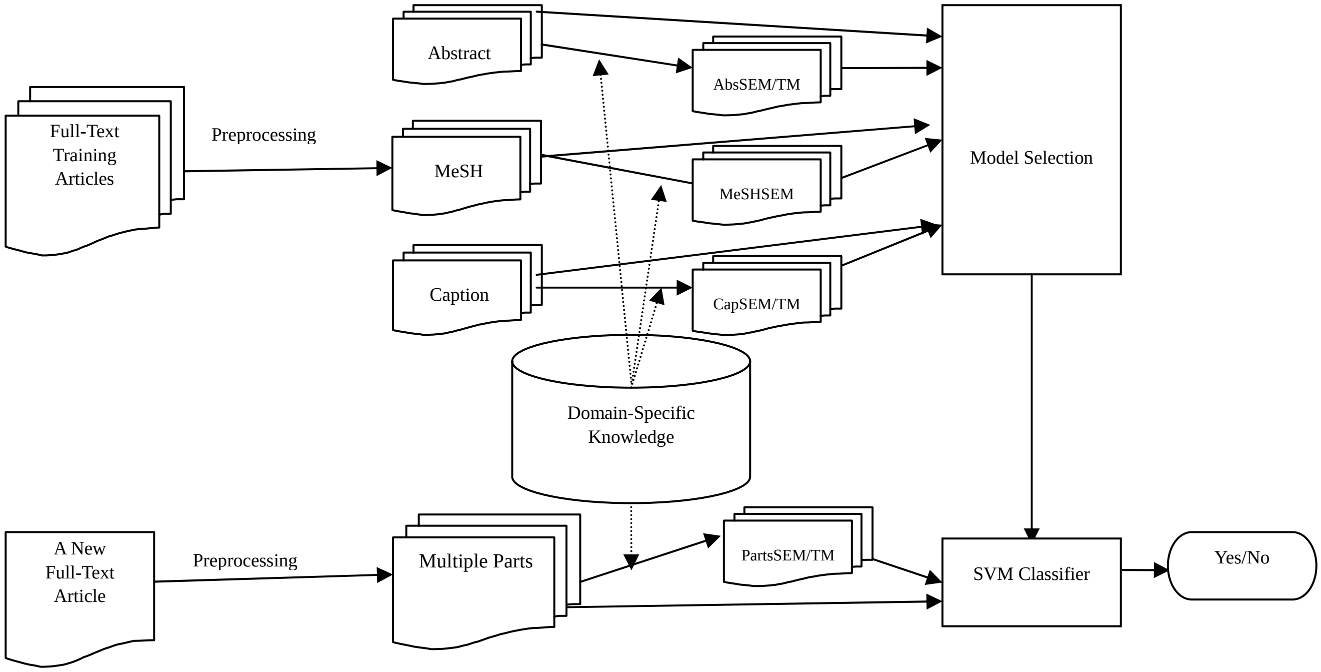

Figure 1. System Architecture

It is presently unclear how these receptors could selectively mediate cAMP responses to sugars and <GLOSREF RID="G3">IP<INF>3</INF></GLOSREF> responses to artificial sweeteners.

It is presently unclear how these receptors could selectively mediate cAMP responses to sugars and inositol trisphosphate (IP<INF>3</INF>) responses to artificial sweeteners.

Figure 2: An Example of Acronym Expansion Operation.

UMLS contains a very large dictionary of biomedical terms – the UMLS Metathesaurus and defines a hierarchy of semantic types – the UMLS Semantic Network. Each concept in the Metathesaurus contains a set of strings, which are variants of each other, and belongs to one or more semantic types in the Semantic Network. Therefore, given a string, we can obtain a set of semantic types to which it belongs. For each part of an article extracted during preprocessing, we obtain another representation of the article by gathering the semantic types found in the part of the article. Consequently, we got three more different representations of an article after this step. They are denoted as **AbstractSEM**, **MeSHSEM** and **CaptionSEM**.

We use the list of tumor names only on the Tumor subtask. We first tokenize all the tumor names and stem each unique token, where the tokenization operation considers a sequence of successive alphanumeric characters as a token. With the resulting list of unique stemmed tokens, we use it as a filter to remove the tokens not in the list from the **Abstract** and **Caption** representations, which produce representations **AbstractTM** and **CaptionTM**.

3.3 Model Selection

As mentioned above, we have several representations for an article. In this section, we explain how feature selection is done for each representation and how the best combination of the representations of an article is obtained. In the following paragraphs, the word “token” refers to different concepts, depending on the representations of an article. A token is a stemmed sequence of consecutive alphanumeric characters for the **Abstract**, **MeSH**, **Caption**, **AbstractTM** and **CaptionTM** representations. For the **AbstractSEM**, **MeSHSEM** and **CaptionSEM** representations, a token is a semantic type.

For each representation, we first rank all the tokens in the training documents via the chi-square test of independence. Assuming the ranking perfectly reflects the effectiveness of the tokens in classification, we then decide the number of tokens to be used in classification by 4-fold cross-validation. In cross-validation, we use the well-known bag-of-word model with TF*IDF (term frequency inverse document frequency) weighting. Each feature vector is normalized to a unit vector after weighting. We adopt SVMs as our classification system and set C_+ to $u_r * C_-$ because of the relatively small number of positive examples, where C_+ and C_- are the penalty constants on positive and negative examples in SVMs. After cross-validation, we obtain the optimal number of tokens and the corresponding SVM parameters C_- and *gamma*, a parameter in the radial basis kernel. In the following paragraphs, **Abstract30** denotes the **Abstract** representation with top-30 tokens, **CaptionSEM10** denotes **CaptionSEM** with top-10 tokens, and so forth.

After feature selection is done for each representation, we try to find the best combination of the representations by a simple algorithm, where combining two or more representations is achieved by simply concatenating the feature vectors. Therefore, under a combined model of *N* representations, each article is represented as an *N*-unit long feature vector. The algorithm is described below.

> Given the candidate representations with selected features, e.g. **Abstract10**, **Caption10** and **Mesh30**, we start with an initial set containing some or zero representation. For each iteration, we add one representation to the set by picking the one that enhances the cross-validation performance the most. The iteration stops when we have exhausted all the representations or adding more representation to the set doesn’t improve the cross-validation performance.

In the categorization task, we run the algorithm twice. We first start with an empty set and obtain the best combination of the basic three representations, e.g., **Abstract10**, **Caption10** and **Mesh30**. Then, starting with this combination, we attempt to incorporate the three semantic representations, e.g., **Abstract30SEM**, **Caption10SEM** and **Mesh30SEM**, and obtain the final combination. Instead of using this algorithm to incorporate the **AbstractTM** and **CaptionTM** representations, we use them to replace their unfiltered counterparts **Abstract** and **Caption** when their cross-validation performance is better.

4 Results and Discussions

Table 1 lists the individual cross-validation result (in NU measure) of each representation for each subtask. For subtask Allele, the **Caption** representation performs the best among the basic representations, while **AbstractSEM** performs the best among the semantic representations. For subtask Expression, we can see that captions in articles play an important role in identifying relevant documents, which agrees with the finding by the winner of KDD CUP 2002 task 1 (Regev *et al.*, 2002). Similarly, we can infer that MeSH terms are crucial to the GO subtask, which also supports the wide-spread using of MeSH terms by top-performing teams (Dayanik *et al.*, 2004;

Fujita, 2004) in TREC Genomics 2004. Looking at the Tumor subtask, we can tell that MeSH terms are important, but after semantic type extraction the **AbstractSEM** representation exhibits relatively high cross-validation performance. Since only 10 features are selected for the **AbstractSEM** representation, using this representation alone may be susceptible to over-fitting. Finally, by comparing the performance of the **AbstractTM** and **Abstract** representations, we find the list of tumor names helpful for filtering abstracts.

Table 1: Partial Cross-validation Results.

	Allele	**Expression**	**GO**	**Tumor**
	# Tokens / NU	# Tokens / NU	# Tokens / NU	# Tokens / NU
Abstract	10 / 0.7707	10 / 0.5586	10 / 0.4411	10 / 0.8055
MeSH	10 / 0.7965	10 / 0.6044	**10 / 0.4968**	**30 / 0.8106**
Caption	**10 / 0.8179**	**10 / 0.7192**	10 / 0.4091	10 / 0.7644
AbstractSEM	**10 / 0.7209**	10 / 0.4811	10 / 0.3493	**10 / 0.8814**
MeSHSEM	10 / 0.6942	10 / 0.4563	**10 / 0.4403**	10 / 0.7047
CaptionSEM	30 / 0.6789	**10 / 0.5433**	10 / 0.2551	30 / 0.7160
AbstractTM				**30 / 0.8325**
CaptionTM				10 / 0.7498

We list the results of our official runs in Table 2. Column "cv NU" shows the cross-validation NU measure, "NU" shows the performance on the test data and column "combination" lists the combination of representations used for each run. In this table, M30 is the abbreviation for the **MeSH30** representation, CS10 represents the **CaptionSEM10** representation, and so on. The combinations for the first 4 runs are obtained by the simple algorithm described in Section 3, while the combination for tNTUMACwj is obtained by substituting **AbstractTM30** for **Abstract30** in the combination for tNTUMAC. The last run tNTUMACasem uses only the **AbstractSEM10** representation because its cross-validation performance beats all other combinations for the Tumor subtask.

Table 2: Results of Our Official Runs.

Run Tag	**cv NU**	**NU**	**Recall**	**Precision**	**F-score**	**Combination**
aNTUMAC	0.8717	0.8423	0.9488	0.3439	0.5048	M30+C10+A10+CS10+AS10+MS10
eNTUMAC	0.7691	0.7515	0.8190	0.1593	0.2667	M10+C10+CS10+MS10
gNTUMAC	0.5402	0.5332	0.8803	0.1873	0.3089	M10+C10+MS10
tNTUMAC	0.8742	0.8299	0.9000	0.0526	0.0994	M30+C30+A30+AS10+CS30
tNTUMACwj	0.8764	0.8747	0.9500	0.0518	0.0982	M30+C30+AT30+AS10+CS30
tNTUMACasem	0.8814	0.5699	0.6500	0.0339	0.0645	AS10

The combinations of the first 5 runs illustrate that adding other inferior representations to the best one enhances the performance, which implies that the inferior ones may contain important exclusive information. The cross-validation performance fairly predicts the performance on the test data, except for the last run tNTUMACasem, which relies on only 10 features and is therefore susceptible to over-fitting.

We list the best and median results for each subtask in Table 3. Each row shows the performance of the best/median teams. For example, the best team of Allele subtask achieves NU of 0.8710, the recall rate of 0.9337, the precision rate of 0.4669, and the F-score of 0.6225. The comparing results of each subtask between the best, median and our methods (NTU) are depicted in Figure 3, 4, 5 and 6. Because we submitted 3 official runs for the tumor subtask, there exist NTU1, NTU2 and NTU3 in Figure 6. Comparing to the results, it shows our experimental results have overall high performance.

Table 3: Best and Median Results for Each Subtask.

Subtask	NU (Best/Median)	Recall (Best/Median)	Precision (Best/Median)	F-score (Best/Median)
Allele	0.8710/0.7773	0.9337/0.8720	0.4669/0.3153	0.6225/0.5010
Expression	0.8711/0.6413	0.9333/0.7286	0.1899/0.1164	0.3156/0.2005
GO Annotation	0.5870/0.4575	0.8861/0.5656	0.2122/0.3223	0.3424/0.4107
Tumor	0.9433/0.7610	1.0000/0.9500	0.0709/0.0213	0.1325/0.0417

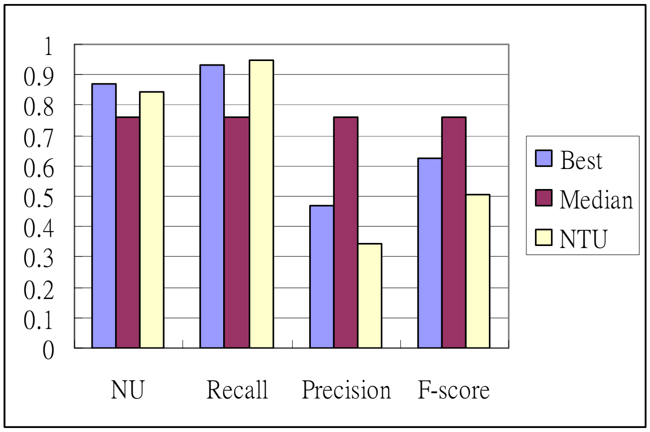

Figure 3. Comparison of Allele Subtask.

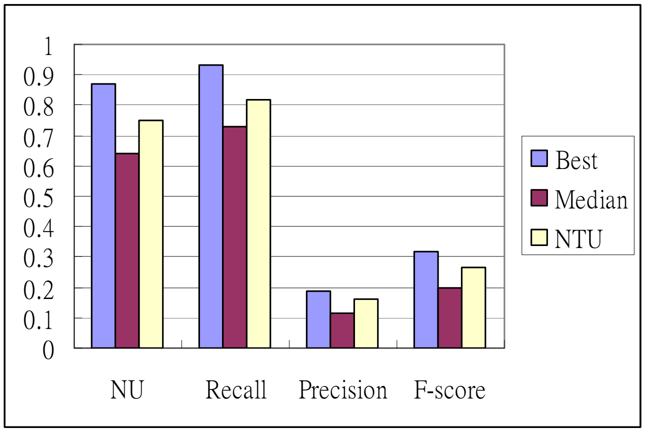

Figure 4. Comparison of Expression Subtask.

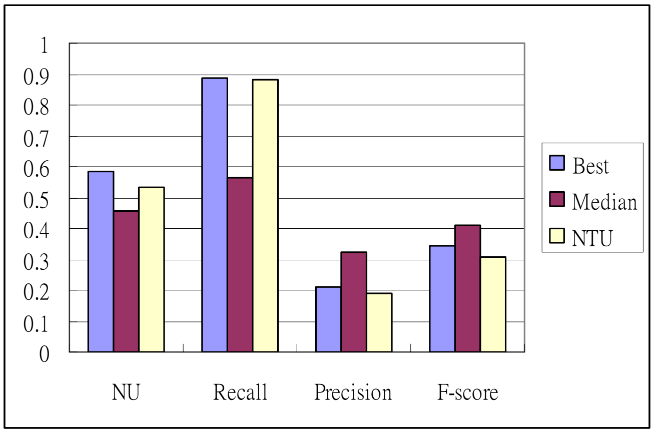

Figure 5. Comparison of GO Annotation Subtask.

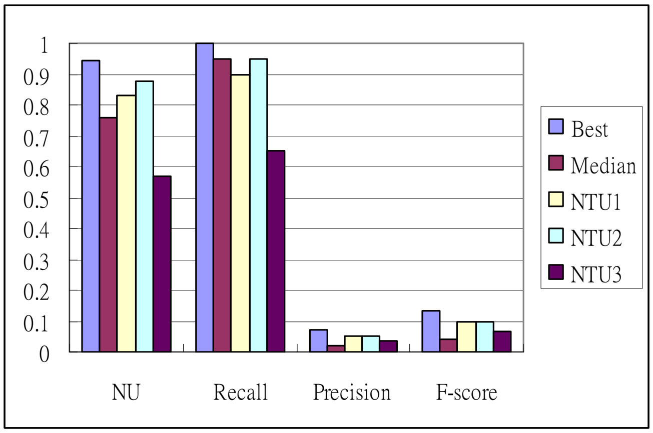

Figure 6. Comparison of Tumor Subtask.

5 Concluding Remarks

In this paper, we demonstrate how our system is constructed. Three parts of an article are extracted and each of them is considered a representation of the article. We incorporate two domain-specific resources, i.e., UMLS and a list of tumor names. By integrating domain knowledge, we obtain 5 more representations of an article. We perform feature selection on each of the 8 representations to obtain the optimal number of features for each of them. For each subtask, we mainly rely on a simple algorithm to get the best combination of the representations and train an SVM classifier out of this combination. The partial cross-validation results and the results of our official runs are listed. Evaluation results show overall high performance in this study.

Acknowledgements

Research of this paper was partially supported by National Science Council, Taiwan, under the contracts NSC94-2213-E-002-033 and NSC94-2752-E-001-001-PAE.

References

Andrade, M.A. and Valencia, A. Automatic Extraction of Keywords from Scientific Text: Application to the Knowledge Domain of Protein Families. *Bioinformatics*, 14, 600-607, 1998.

Bult, C.J., Blake, J.A., Richardson, J.E., Kadin, J.A., Eppig, J.T. and the Mouse Genome Database Group. The Mouse Genome Database (MGD): integrating biology with the genome. *Nucleic Acids Research*, 32, D476–D481, 2004.

Camon, E., Magrane, M., Barrell, D., Binns, D., Fleischmann, W., Kersey, P., Mulder, N., Oinn, T., Maslen, J., Cox, A. and Apweiler, R. (2003) The Gene Ontology Annotation (GOA) project: implementation of GO in SWISS-PROT, TrEMBL, and InterPro. *Genome Research*, 13, 1-11.

Couto, F.M., Martins, B. and Silva, M.J. Classifying Biological Articles Using Web Resources. *Proceedings of the 2004 ACM Symposium on Applied Computing*, 111-115, 2004.

Dayanik, A., Fradkin, D., Genkin, A., Kantor, P., Lewis, D.D., Madigan, D. and Menkov, V. DIMACS at the TREC 2004 Genomics Track. *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

Drysdale, R.A., Crosby, M.A. and the FlyBase Consortium. FlyBase: genes and gene models. *Nucleic Acids Research*, 33, D390–D395, 2005.

FlyBase. http://flybase.org

Fujita, S., Revisiting Again Document Length Hypotheses TREC-2004 Genomics Track Experiments at Patolis. *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

Hersh, W.R., Bhuptiraju, R.T., Ross, L., Johnson, P., Cohen, A.M. and Kraemer D.F. TREC 2004 Genomics Track Overview. *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

Hill, D.P., Begley, D.A., Finger, J.H., Hayamizu, T.F., McCright, I.J., Smith, C.M., Beal, J.S., Corbani, L.E., Blake, J.A., Eppig, J.T., Kadin, J.A., Richardson, J.E. and Ringwald, M. The Mouse Gene Expression Database (GXD): updates and enhancements. *Nucleic Acids Research*, 32:D568-D571, 2004.

Hirschman, L., Park, J., Tsujii, J., Wong, L. and Wu, C.H. Accomplishments and Challenges in Literature Data Mining for Biology. Bioinformatics, 18(12): 1553-1561, 2002.

Hou, W.J., Lee, C., Lin, K.H.Y. and Chen, H.H. A Relevance Detection Approach to Gene Annotation. *Proceedings of the first International Symposium on Semantic Mining in Biomedicine*, http://ceur-ws.org, 148: 15-23, 2005.

Hsu, C.W., Chang, C.C. and Lin C.J. A Practical Guide to Support Vector Classification. http://www.csie.ntu.edu.tw/~cjlin/libsvm/index.html, 2003.

Humphreys, B.L., Lindberg, D.A., Schoolman, H.M. and Barnett, G.O. The Unified Medical Language System: an Informatics Research Collaboration. *Journal of American Medical Information Association*, 5(1):1-11, 1998.

Lee, C., Hou, W.J. and Chen, H.H. Identifying Relevant Full-Text Articles for GO Annotation Without MeSH Terms. *Proceedings of the Thirteenth Text Retrieval Conference (TREC 2004)*, 2004.

Manning, C. and Schutze, H. *Foundations of Statistical Natural Language Processing*. The MIT Press, 1999.

Perez, A.J., Perez-Iratxeta, C., Bork, P., Thode, G. and Andrade, M.A. (2004) Gene Annotation from Scientific Literature Using Mappings between Keyword Systems. *Bioinformatics*, 20(13), 2084-2091, 2004.

Pouliot, Y., Gao, J., Su, Q.J., Liu, G.G. and Ling, X.B. DIAN: a Novel Algorithm for Genome Ontological Classification. *Genome Research*, 11 1766-1779, 2001.

Regev, Y., Finkelstein-Landau, M, and Feldman, R. Rule-based Extraction of Experimental Evidence in the Biomedical Domain - the Kdd Cup (Task 1). *SIGKDD Explorations*, 4(2):90-92, 2002.

Shatkay, H., Edwards, S., Wilbur, W.J. and Boguski, M. Genes, Themes, and Microarrays: Using Information Retrieval for Large-scale Gene Analysis. *International System Molecular Biology*, 8, 317-328, 2000.

Tumor Gene Database. http://www.tumor-gene.org/TGDB/tgdb.html

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

Xie, H., Wasserman, A., Levine, Z., Novik, A., Grebinskiy, V., Shoshan, A. and Mintz, L. Large-scale Protein Annotation through Gene ontology. *Genome Research*, 12, 785-794, 2002.

Yeh, A., Hirschman, L. and Morgan, A. Background and Overview for KDD Cup 2002 task 1: Information Extraction from Biomedical Articles. *SIGKDD Explorations*, 4:87-89, 2002.