Acta Cybernetica **18** (2008) 391–402.

Extracting Human Protein Information from MEDLINE Using a Full-Sentence Parser

Róbert Busa-Fekete* and András Kocsor*†

Abstract

Today, a fair number of systems are available for the task of processing biological data. The development of effective systems is of great importance since they can support both the research and the everyday work of biologists. It is well known that biological databases are large both in size and number, hence data processing technologies are required for the fast and effective management of the contents stored in databases like MEDLINE. A possible solution for content management is the application of natural language processing methods to help make this task easier.

With our approach we would like to learn more about the interactions of human genes using full-sentence parsing. Given a sentence, the syntactic parser assigns to it a syntactic structure, which consists of a set of labelled links connecting pairs of words. The parser also produces a constituent representation of a sentence (showing noun phrases, verb phrases, and so on). Here we show experimentally that using the syntactic information of each abstract, the biological interactions of genes can be predicted. Hence, it is worth developing the kind of information extraction (IE) system that can retrieve information about gene interactions just by using syntactic information contained in these text. Our IE system can handle certain types of gene interactions with the help of machine learning (ML) methodologies (Hidden Markov Models, Artificial Neural Networks, Decision Trees, Support Vector Machines). The experiments and practical usage show clearly that our system can provide a useful intuitive guide for biological researchers in their investigations and in the design of their experiments.

Keywords: feature extraction, human gene interaction, data mining, machine learning, MEDLINE

*Research Group on Artificial Intelligence of the Hungarian Academy of Sciences and University of Szeged H-6720 Szeged, Aradi vértanúk tere 1., Hungary, E-mail: {busarobi,kocsor}@inf.u-szeged.hu

†The author was supported by the János Bolyai fellowship of the Hungarian Academy of Sciences.

1 Introduction

The MEDLINE [1] database is today becoming the most comprehensive biomedical abstract repository among the life sciences literature. Due to its easy access and availability, it is one of the most widely-used sources of scientific data employing several information retrieval systems. The NLM (U.S. National Library of Medicine) maintained MEDLINE database contains over 13 million references from about 4900 journals dating from 1965 to the present, and it is updated weekly. Obviously a crucial task in bioinformatics text mining is to develop an automatic system that extracts information about genes and their interactions. That is why we decided to build an information extraction (IE) system which makes use of natural language processing (NLP) techniques.

In the human life sciences, researchers are mostly interested in the interactions of genes, so in this area of science it would be good if biologists had an IE system that could search for relationships among genes [2, 3, 4, 5]. An interaction means, for instance, the binding of genes, or the existence of a gene that can influence the function of another gene. This kind of IE system can be quite useful in the design of biological experiments and drugs. Hence here we will introduce a system that can extract information about gene interactions which occur in living human cells from the MEDLINE. Because we wanted to avoid the building of huge and costly databases, we used and processed only freely available datasets. The system introduced here relies on the part of speech tagged (POS) and full sentence parsed (FSP) parts of MEDLINE. The main aim of our system is twofold: (a) to explore the MEDLINE abstracts for a set of genes given by the user and (b) to gather information about the interactions of genes that are described in the text of an abstract provided by the user.

One of the cornerstones of our information extraction system is the recognition of gene names. We used a thesaurus containing more than $40,000$ gene names and their $120,000$ synonyms to annotate the gene names in the abstracts. The thesaurus we obtained was built up using two sources: UMLS SPECIALIST Lexicon[7] and the Agilent Technologies [8] database. With this lexicon the identification of gene names can be reliably carried out. Later we will show some results of the efficiency of gene name recognition.

To predict new gene interactions we first needed a part of MEDLINE that had been annotated manually. In particular we needed an annotation based on the interaction of gene pairs when they occurred in the text of the same abstract. The National Center for Biotechnology Information (NCBI) [6] has many databases about gene interactions that have been arranged taxonomically. Using these data sets we were able to get a subset of MEDLINE containing information about human gene interactions. In this way we could derive a training set suitable for machine learning methods. Actually, many features of a syntactic tree can be represented as a multidimensional vector (i.e. depth and frequencies of different labels), hence each pair of gene names can be represented as a vector. In addition, the database about the interactions allows us to find out whether a sample is positive or negative (i.e. whether it is about the interaction of genes occurring in the text.) Here we

tested many machine learning algorithms on the training set. The results shows that the Decision Tree model is the most suitable one for this purpose.

When designing our system we had to take into account the fact that MEDLINE is a rapidly growing system and that the data is stored in compressed XML file format. So we created a framework which could handle the abstracts and their updates in their raw form, and could incorporate them into our IE system.

Not so long ago there was a workshop devoted to genic interaction extraction using MEDLINE records. The task was quite similar to ours here, and many results were produced by the participants of the workshop. These results are freely available [21], and comparing our results we can say that our results are competitive with the state-of-the-art systems. Moreover, we deal with a much bigger part of the gene set.

The paper is organised as follows. In the next section we will discuss the Feature Extraction task and how we can combine the databases that are available. Then in Section 3 we provide a brief overview of the Machine Learning models we employed in experiments. Section 4 following gives a comprehensive study of the performance of our IE system. Lastly, in Section 5, we summarise our results and offer suggestions for future work.

2 Extracting Information from MEDLINE using Distinct Data Sets

2.1 Relationships of the Applied Databases

With the advent of microbiology the experiments produce such a huge amount of information that it has become necessary to organize them into databases. In the middle of the last century some biologists began to collect and organize the papers on the results of biological and chemical experiments. Later this collection, called MEDLINE, became the main information resource for the experts dealing with biology, pharmacology and the human life sciences. Nowadays with the advent of extensive genome projects MEDLINE is getting bigger and bigger, and it contains an indigestible amount of information about proteins, genes and so on. For fast searching among the 13 billion records each abstract has a unique ID. It is called its PMID, and it makes the identification of entries much easier.

In order to construct a working system in the first step we needed to isolate the part of MEDLINE that is connected with human genes. Hence we generated a comprehensive lexicon containing the names and synonyms of the genes. Because biological work and experiments have gone on in parallel without being synchronized many genes were discovered in different labs at the same time, and they were named in a different way. These things makes the recognition of gene names harder. To resolve this problem we used the most comprehensive lexicons of the big biological research centres and chemical labs. The two lexicons that we then merged are the following:

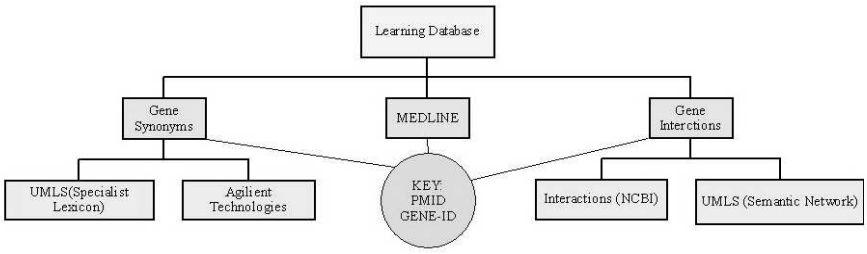

Figure 1: The relationships of the databases

- UMLS Specialist Lexicon
- Lexicon of Aglient Techonolgies

To identify a gene in two distinct lexicons we used a unique gene identification number. This ID is based on a consensus, and it is universal. In this way we obtained a synonym containing more than $40,000$ gene IDs and about $120,000$ synonyms. It allowed us to have a reliable gene name annotation.

Before we began extracting features we needed to have a part of MEDLINE that was not just about human genes, but about their interactions too. For this we used the interaction database of NCBI. It contains triplets, that is the gene ID of the interactants, and the PMID of the abstracts whose content is about their interactions. Using this we generated a subset of MEDLINE with $12,638$ texts of abstracts, and then we annotated the resultant text by applying the synonyms lexicon mentioned above. We made use of this part of MEDLINE for the preparation of the training database.

Summarizing the above-mentioned points, in our studies we generated tree databases for the preparation of the training set: an extended synonym lexicon, a collection of interactions and a part of MEDLINE. The connections between them can be seen in Figure 2.1 later on. The unique PMID and gene ID were of course used as the primary key.

2.2 Relevant nouns and verbs

During our study we noticed that only a few nouns and verbs rarely occur between the pairs of gene names in the abstracts. One reason for this is that the texts were mostly written by researchers who were non-native speakers of English. Another was that a technical text always has a poor vocabulary. For instance in the following sentence the verb *interact* characterizes the given situation very well: *Here we show that BRCA1 interacts and colocalizes with topoisomerase IIalpha in S phase cells.*

In our studies we collected the nouns and verbs that frequently occur between the interactant genes and, based on their distributions we chose the relevant verbs and nouns. In our case a verb was deemed relevant if it occurred more than 1% in any case. These results eventually gave us 35 nouns and 15 verbs. We used this information as a discrete feature in the machine learning task as well.

2.3 Part of Speech Tagging and Syntactic Parsing

A morpho-syntactically analyzed text contains many possible parts of speech tags based on the word stem. The aim of the Part Of Speech (POS) Tagging is the selection of the appropriate POS Tags for each word according to its grammatical role in the sentence. The widespread approaches are based on machine learning techniques available today. But here we used the POS Tagger developed by the Human Language Technology Group of the University of Szeged. This tagger utilizes the internationally acknowledged MSD (Morpho-Syntactic Description) scheme [20] that is also used for encoding words. Due to the fact that the MSD encoding scheme is extremely detailed (one label can store morphological information on up to 17 positions), we did not exploit the granularity of this sort of annotation scheme. We only employed the following groups:

Adjective	Particle
Conjunction	Adverb
Determiner	Postposition
Interjection, sentence word	Article
Numeral	Verb
Noun	Other, unknown word
Special tokens	Abbreviation
Pronoun	

Syntactic parsing is the process of finding the immediate constituents of a sentence, that is a sequence of words. Syntactic parsing is an important part of the field of natural language processing and it is useful for supporting a number of large-scale applications including information extraction. Here we carried out the syntactic parsing of the texts of abstracts using the Link Grammar Parser [16]. The syntax trees of annotated sentences contain various types of phrases, as shown in the following list:

Noun phrase (NP)	Verb prefix (PREVERB)
Adjective phrase (ADJP)	Conjunction (C)
Adverb phrase (ADVP)	Pronoun phrase (PP)
Verb phrase (VP)	Clause (CP)
Infinitive(INF)	Sentence (S)
Negative (NEG)	

To build a learning dataset we collected different types of numerically encodable information describing each tag (part of speech and syntactic). These constituted the vector of attributes for the classification. After we made use of the number of

distinct POS tags that can be found between two given gene names. Here we also utilised the distinct number of syntactic tags that can be found on the only path between the pairs of the genes in the syntactic tree as features. Thus this gave us 35 features based on the sum of the number of POS tags and syntactic tags we used.
In summary the features we employed were the following:

- the number of words between two protein names
- part-of-speech code syntactic labels (for the two protein names themselves and for the words between them)
- the relevant nouns and verbs that occur in a sentence

3 The learning models

To solve classification problems effectively it is worth applying various types of classification methods. The features we used are discrete. Therefore we applied the C4.5 decision tree model, which usually works well on discrete feature set. The SVM classifier was also applied using binarized kernel function because these functions can be more suitable for discrete features than the traditional ones such as Gaussian RBF kernel and polynomial kernel function. We compared these two models to the Hidden Markov Model and Artificial Neural Network which are widely used models. Now we will provide a brief overview of the learning models we applied to the problems.

3.1 C4.5

C4.5 [19] is based on the well-known ID3 tree learning algorithm. It is able to learn pre-defined discrete classes from labeled examples. The result of the learning process is an axis-parallel decision tree. This means that during the training, the sample space is divided into subspaces by hyperplanes that are parallel to every axis but one. In this way, we get many n-dimensional rectangular regions that are labeled with class labels and organized in a hierarchical way, which can then be encoded into the tree. Since C4.5 considers attribute vectors as points in an n-dimensional space, using continuous sample attributes naturally makes sense. For knowledge representation, C4.5 uses the "divide and conquer" technique, meaning that regions are split during learning whenever they are insufficiently homogeneous. Splitting is done by axis-parallel hyperplanes, and hence learning is very fast. One great advantage of the method is time complexity; in the worst case it is $O(dn^2)$, where d is the number of features and n is the number of samples. Based on this we ran the C4.5 algorithm numerous times to perform preliminary tests to decide whether the inclusion of additional features was beneficial to the model or not.

3.2 Hidden Markov Model (HMM) approach

The Hidden Markov Modelling (HMM) technology it is assumed that the observation vectors belonging to a given state are independent, which in turn implies that the corresponding likelihood values can be combined by multiplication. With this poind in mind we adopted this probabilistic approach to predict new interactions. Each class was represented by a HMM, and the decision rule was based on the maximum posteriori probability derived from the HMMs. When we just used morpho-syntactic information the observation was the POS tags between the pairs of gene names. We also tried out the HMM approach on syntactic information (i.e. on the syntactic tags between two gene names), where the input data was the set of syntactic tags on the only path between the interactants. Here we varied the number of states between 2 and 5, because this was found to be the best empirically.

3.3 Artificial Neural Networks (ANN)

Since it was realized that, under the right conditions, ANNs can model class posteriors [13], neural nets have become evermore popular in the Natural Language Processing field. ANNs are based on the parallel architecture of the brain. We can view it as a simple multiprocessor system with a large number of interconnections and interactions between the processing units that use scalar messages. However, describing the mathematical background of ANNs is beyond the scope of this article. Besides, we believe that they are already well known to those who are acquainted with pattern recognition. In the ANN experiments we utilised the most common feed-forward multilayer perceptron network with the backpropagation learning rule.

3.4 Support Vector Machines and Binarized Kernel Functions

Theoretical discoveries generally have their own very different, unique histories before they find any practical application. One such example is the "kernel-idea", which had appeared in several fields of mathematics and mathematical physics before it became a key notion in machine learning. The kernel idea can be applied in any case where the input of some algorithm consists of the pairwise dot (scalar) products of the elements of an n-dimensional dot product space. In this case, simply by a proper redefinition of the two-operand operation of the dot product, we can have an algorithm that will now be executed in a different dot product space, and is probably more suitable for solving the original problem. Of course, when replacing the operand, we have to satisfy certain criteria, as not every function is suitable for implicitly generating a dot product space. The family of Mercer Kernels is, however, a good choice, and is based on Mercer's theorem [18]. Here we turn to the well-known and widely used Support Vector Machines (SVMs) [14, 15], which is a kernel method that separates data points of different classes with the help of a hyperplane. This separating hyperplane produced has a margin of maximal size with a verified optimal generalisation capability. Another nice feature of margin

maximization is that the calculated result is independent of the distribution of the sample points. Perhaps the success and popularity of the method can be attributed to this property.

There are many kernel functions for us to use, and there are also many ways of deriving functions from the existing ones. From the functions available, the two most popular are:

$$\text{Polynomial kernel:} \quad k_1(\mathbf{x},\mathbf{y}) = \left(\mathbf{x}^\top \mathbf{y}\right)^d, \qquad d \in \mathbb{N}, \tag{1}$$

$$\text{Gaussian RBF kernel:} \quad k_2(\mathbf{x},\mathbf{y}) = \exp\left(-||\mathbf{x}-\mathbf{y}||^2/r\right), \qquad r \in \mathbb{R}_+ \tag{2}$$

$$\text{Cosine polynomial kernel} \quad k_3(\mathbf{x},\mathbf{y}) = \left(\frac{\mathbf{x}^T\mathbf{y}}{\|\mathbf{x}\|\,\|\mathbf{y}\|} + \sigma\right)^q, \qquad q \in \mathbb{N},\ \sigma \in \mathbb{R}_+. \tag{3}$$

Our feature set consists of discrete features. Hence we will investigate a derivation technique for kernel functions. This technique will make the kernel functions more suitable for discrete features. The main idea behind it is to use the binarized form of the input vectors. In detail if we have a feature having a domain set $|D| = k$, then we can map D into a binary vector space. This bijection represents the elements of D as binary vectors with a fixed length of k. Each binary vector has precisely one non-zero element. Thus each coordinate in the binary representation corresponds to a value in the domain set. Let us denote the mapping by $\eta(\mathbf{x})$ that carries out this bijective mapping componentwise for the input space. Using this η mapping we can extend the well-known kernel functions:

$$k^b(\mathbf{x},\mathbf{y}) = k(\eta(\mathbf{x}), \eta(\mathbf{y})) \tag{4}$$

All of the well-known kernel functions have a binarized form, and the experiments clearly show that, using this kind of kernel on a discrete feature set, the SVM can achieve a higher classification performance.

4 Experiments

4.1 The performance of the learning method

We carried out our experiments using the dataset described in Section 2. We then obtained a learning database containing 22195 positive and 90656 negative samples. The evaluation method we used here was a 10-fold cross validation.

We tried out various learning methods that were outlined in Section 3. In tests we found that the C4.5 decision tree learning method slightly outperformed the other machine learning algorithms. Here the confidence factor was set to 0.33. The ANN method had one hidden layer of one and half times more the number of hidden units than input neurons, used sigmoid activation functions, and 50 training session had a 0.3 learning rate. The SVM method gave a better performance using the binarized form of the well-known kernel function (linear, polynomial and cosine kernels). The best results using the SVM approach was achieved with a binarized

NEGATIVE POSITIVE	PRECISION(%)	RECALL(%)	F-MEASURE(%)
SVM (linear)	61.02 59.56	60.54 60.05	60.78 59.8
SVM (cosine pol.)	63.29 60.29	58.67 64.84	60.89 62.49
SVM (binarized lin.)	63.0 60.56	59.86 63.67	61.39 62.08
SVM (binarized cos.)	67.3 64.11	62.92 68.42	65.04 66.19
ANN	72.07 68.58	70.33 70.39	71.19 69.47
C4.5	71.64 71.89	73.53 69.93	72.58 70.9
HMM(POS)	59.52 56.39	53.03 62.74	56.09 59.39

Table 1: The classification performance of the various algorithms. In each cell the upper and lower values correspond to the performance of the different machine learning models on the negative class and positive class, respectively.

cosine polynomial kernel of 3rd degree. The performance of the HMM was better when we used POS tags as observations. The number of states was always tested in the range $1-5$. The experiments revealed that the 3-state HMM best fit the problem, although ironically it gave the worst performance. As the reader will notice in Table 4.1 below, each cell contains the percentage accuracy for the positive and negative classes separately. The columns on the other hand list the precision, recall and F-measure for each method we tested.

4.2 Description of the system

The system can be accessed through a web-based user interface that has two types of queries. First, the user can request a query concerning gene names. In response, the system can provide information about the MEDLINE records that describes the given genes. In addition, the system can visualise the results using Multi Dimensional Scaling. Hence the users will be better able to understand the relationships of the genes in question. Second, the user can also provide a text of an abstract as an input for the system. The system will then collect the gene names that crop up in the text, and it will may discover a possible interaction pattern among the genes that are listed in the abstract. A schematic overview of this is given in Figure 2.

The usefulness of the system cannot be underestimated as it considerably facilitates biological and biomedical activities in two ways. First, it supports the comparability of research findings achieved in different countries. Experiences can

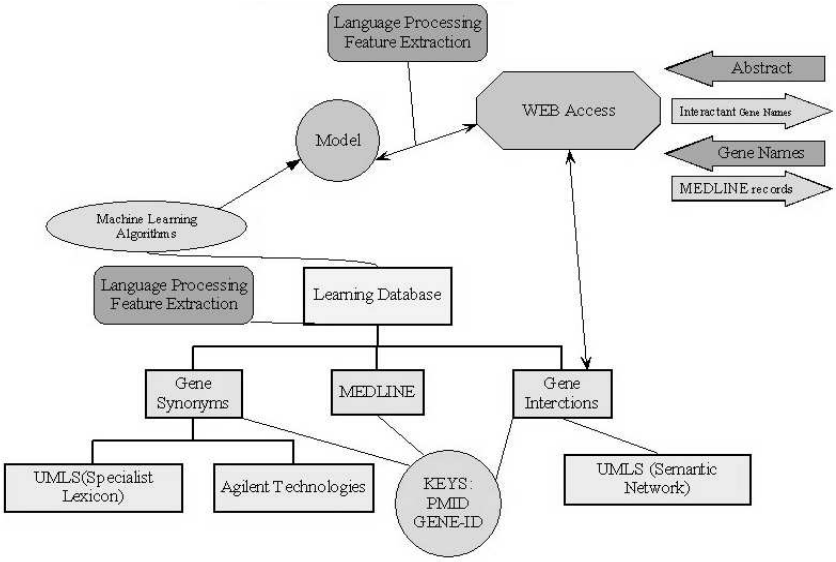

Figure 2: A schematic overview of the system

be accumulated in this way, and conclusions, deductions and extrapolations can be found more easily and in a shorter time. Second, it helps reduce research costs, since results can be obtained in a automated way and this also eliminates the need for many people to work on this time-consuming and laborious task. This way, researchers can focus on a variety of areas using the results produced by the system, and perhaps make new discoveries using our software tool. Below in Figure 3 is a screenshot of the system in operation.

5 Conclusions and Further Work

The information extraction technology presented here differs from existing methods in that it applies semantic-based natural language processing methods to biological content processing problems. As a novel aspect, it can visualise in a graph-like form the information about genes and their interactions that was retrieved, which can then be interactively browsed. This technology facilitates the work of researchers by providing structured, customisable and easily browsable information relating to their daily work. For this reason we think that it is certainly worthwhile developing our system further.

Next, we intend to improve our gene interaction recogniser using syntactic frame matching. This approach is a very commonly used technique in NLP, and we plan to define semantic frames. We hope that with these frames we will able to determine

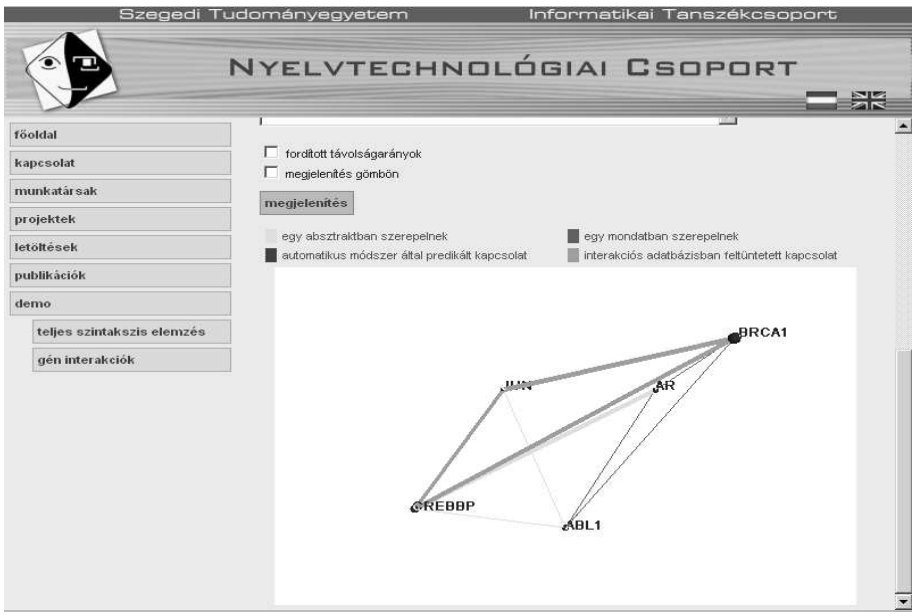

Figure 3: A screenshot of the system

the level of gene interactions as well. This would be a promising start in building a useful informatics tool for bio research and development.

References

[1] http://www.pubmedcentral.nih.gov

[2] Takeshi Sekimizu, Hyun S. Park, Jun'ichi *Identifying the Interaction between Genes and Gene Products Based on Frequently Seen Verbs in Medline Abstracts.* Genome Informatics 9:62-71, 1998.

[3] Marcotte EM, Xenarios I, Eisenberg D. *Mining literature for protein-protein interactions.* Bioinformatics. 2001 Apr;17(4):359-63.

[4] Ono T, Hishigaki H, Tanigami A, Takagi T. *Automated extraction of information on protein-protein interactions from the biological literature.* Bioinformatics. 2001 Feb;17(2):155-61.

[5] Donaldson I, Martin J, de Bruijn B, Wolting C, Lay V, Tuekam B, Zhang S, Baskin B, Bader GD, Michalickova K, Pawson T, Hogue CW. *PreBIND and Textomy–mining the biomedical literature for protein-protein interactions using a support vector machine.* BMC Bioinformatics. 2003 Mar 27;4:11. Epub 2003 Mar 27.

[6] http://www.ncbi.nlm.nih.gov

[7] http://www.nlm.nih.gov/research/umls

[8] http://www.home.agilent.com

[9] Daniel Sleator and Davy Temperley. *Parsing English with a Link Grammar.* Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October 1991.

[10] John Lafferty, Daniel Sleator, and Davy Temperley. *Grammatical Trigrams: A Probabilistic Model of Link Grammar.* Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language, October, 1992.

[11] Dennis Grinberg, John Lafferty and Daniel Sleator. *A robust parsing algorithm for link grammars.* Carnegie Mellon University Computer Science technical report CMU-CS-95-125, and Proceedings of the Fourth International Workshop on Parsing Technologies, Prague, September, 1995.

[12] V. N. Vapnik. *Statistical Learning Theory.* John Wiley and Son, 1998.

[13] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, 1995.

[14] N. Cristianini and J. Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, ISBN 0-521-78019-5, 2000.

[15] B. Schölkopf, C.J.C. Burges, and A.J. Smola. *Advances in Kernel Methods: Support Vector Learning.* MIT Press, Cambridge, MA, 1999.

[16] http://www.link.cs.cmu.edu/link/

[17] http://www.hprd.org

[18] J. Mercer. *Functions of positive and negative type and their connection with the theory of integral equations.* Philos. Trans. Roy. Soc. London, 415–446,1909.

[19] J. R. Quinlan. *C4.5: Programs for machine learning.* Morgan Kaufmann, 1993.

[20] Erjavec, T. and Monachini, M., ed. (1997) Specification and Notation for Lexicon Encoding, *Copernicus project 106 "MULTEXT-EAST"*, Work Package WP1 - Task 1.1 Deliverable D1.1F.

[21] http://genome.jouy.inra.fr/texte/LLLchallenge/