Beyond the Bag of Words: a Text Representation for Sentence Selection

Maria Fernanda Caropreso[1]. Stan Matwin[1,2].

1- School of Information Technology and Engineering. University of Ottawa
Ottawa,Ontario K1N 6N5
{caropres,stan}@site.uottawa.ca
2- Institute for Computer Science, Polish Academy of Science, Wavsaw.

Abstract. Sentence selection shares some but not all the characteristics of Automatic Text Categorization. Therefore some but not all the same techniques should be used. In this paper we study a syntactic and semantic enriched text representation for the sentence selection task in a genomics corpus. We show that using technical dictionaries and syntactic relations is beneficial for our problem when using state of the art machine learning algorithms. Furthermore, the syntactic relations can be used by a first order rule learner to obtain even better performance.

1 Introduction

Sentence selection (SS) consists in identifying the relevant sentences for a particular purpose. This is a necessary step in many document-processing tasks, such as Text Summarization (TS) and Information Extraction (IE). The proportion of sentences considered relevant for the above tasks in a given document is usually low, making some pre-filtering a prerequisite.

Sentence selection can be considered a particular case of Automatic Text Categorization (ATC), which consists in automatically building programs capable of labeling natural language texts with categories from a predefined set. ATC is performed using standard Machine Learning methods in a supervised learning task. The standard text representation used in ATC is the Bag of Words (BOW), which consists of representing each document by the words that occur in it. This representation is also used in related tasks such as Information Retrieval (IR) and IE. Different ways of expanding this representation have been tried on these areas of research, some of the expansions aiming to add some semantic or syntactic knowledge. For example, on the semantic side, stemming words[1], clustering similar terms together [10], and using background knowledge [25] have been tried. Less work has been done on the syntactic side. The latest include using noun phrases [11] [14]

[1] Even when stemming requires only morphological processing, we consider it to semantically expand the representation since several words will represent the same sense once stemmed.

and statistical phrases [13] [1] in the representation, defining position related predicates in an Inductive Logic Programming (ILP) system [2], and incorporating the order of noun phrases in the representation [8].

Even when SS and ATC are related, not all their characteristics are the same. One of the differences is that the sentences are short in length, with few words from the vocabulary happening in each of them. That would result in an even more sparse representation than in the ATC case. Another difference is that ATC is usually used to recognize the general topic of a document, while SS concentrates on more specific details. Because of these differences, some variations to the standard representations and techniques usually used for ATC might be beneficial for SS.

We address the task of sentence selection working on a corpus of texts on genetics. The sentences are short in length and the vocabulary of this corpus is highly specific. We believe that, because of these characteristics, the use of syntactic and semantic knowledge could be even more beneficial than in a collection of a more general nature. The extensions that we propose in this paper have, to our best knowledge, been tried neither for document classification nor for sentence selection.

Our work is devoted to identification of relevant sentences in scientific abstracts on genetics. Those abstracts are written in natural language and can be searched via the Internet using keyword queries. However, the queries would retrieve a large superset of relevant papers [17] from which we would like to identify the sentences that express an interaction between genes and/or proteins. Due to the continuous submission of new abstracts, this task becomes repetitive and time consuming. Because of that, automatic sentence selection is considered of interest to the scientific community. We automatically learn classifiers that categorize the sentences from the abstracts into two classes: those that describe an interaction between genes and/or proteins and those that do not. In those classifiers we study the usefulness of including syntactic and semantic knowledge in the text representation. We accomplish this by adding into the representation pairs of related words (to which we will refer as syntactic bi-grams) obtained from a syntactic parser together with technically related dictionaries. Our experiments include the state of the art machine learning algorithms Naïve Bayes and Support Vector Machine, as well as a relational learner for which a particular relational representation was created.

In the remainder of this paper we first introduce some related work and we present the details of our approach and our dataset. Afterwards we present the representations that we used and the experiments we performed together with their results and their analysis. We finish the paper presenting our conclusions and future work.

2 Related Work

The usefulness of syntactic and statistical phrases compared to the BOW was first studied by Fagan [4] in the IR context. In these experiments it was shown that

statistical phrases were not only easier to obtain but they also improved performance more than syntactic phrases.

In [11] and [12] Lewis compared different representations using either words or syntactic phrases (but not a combination of both) for IR and ATC. The results with the phrases representation showed no significant improvement with respect to the representation using the BOW. Mitra et al. [14] study the usefulness of linguistic knowledge for an IR system. The results indicate that the noun phrases are useful for lowly ranked answers but not so much for the highly ranked answers where the words alone perform well. Similar results were obtained in ATC by Furnkranz et al. [6] when building syntactic phrases following some particular syntactic patterns learned from the data by an extraction system. Dumais et al. [3] studied the use of syntactic phrases with a variety of text classifiers on the Reuters-21578 collection showing no benefit at all from the use of this representation. Scott and Matwin [19] also noted no significant improvement of the performance by adding noun phrases to the representation of the same corpus but using a different Machine Learning algorithm.

Furnkranz et al. [5], Mladenic and Grobelnik [13] and Caropreso et al. [1] studied the usefulness of statistical phrases in ATC. The more discriminating phrases were added to the BOW. The experiments showed that the use of these phrases could in some cases improve the classification.

Maarek's system GURU [27] used lexical affinities for indexing purposes in an IR task. Linguistically, lexical affinities are words that are involved in a modifier-modified relationship and that appear often together in the language. This work, however, only takes into consideration the closeness of the chosen words.

Cohen and Singer [2] study the importance of introducing the order of the words in the text representation by defining position related predicates in an ILP system. This has been extended by Goadrich et al. [8] in recent research in the IE area, incorporating the order of noun phrases into the representation. In other work in IE, Ray and Craven [18] incorporate syntactic phrases to a Hidden Markov Model (HMM) that recognizes the grammatical structure of sentences expressing biomedical relations. The results show that this approach learns more accurate models than simpler HMMs that do not use phrases in the representation. One more approach to IE that uses syntactic information is Temkin and Gilder work [23]. In this work a Context Free Grammar (CFG) was defined to recognize protein, gene and small molecule interactions. The results show that efficient parsers can be constructed for extracting these relations.

Several studies have introduced semantic knowledge in ATC. Siolas [20] does that by building a kernel that take into account the semantic distance: first between the different words based on WordNet, and then using Fisher metrics in a way similar to Latent Semantic Indexing (LSI). Zelikovitz and Hirsh [25] show that the ATC accuracy can be improved by adding extra semantic knowledge into the LSI representation coming from unclassified documents.

3 Our Approach and Dataset

We study the usefulness of including syntactic and semantic knowledge in the text representation for the selection of sentences from technical genomic texts. In this specific context, the occurrence (or not) of specialized terms is expected to discriminate between sentences that contain information about genes and/or proteins interaction, and those that do not contain that information. We expect syntactic bi-grams formed by words that are syntactically linked to provide detailed information on whether two genes and/or proteins are interacting with each other. Such phrases could be formed for example by an adjective modifying a noun, the main noun in the subject or object role of a sentence together with its verb, or the main noun in a prepositional phrase together with either the noun or verb it modifies. Using the syntactic bi-grams together with their single words, we represented the sentences and we evaluated the classification performance of this representation compared to the BOW. Our experiments include state of the art machine learning algorithms Naïve Bayes and Support Vector Machine, and they were performed using Weka [24]. A relational representation was also obtained using the links information, and its performance evaluated using the relational learner Aleph [21].

It is understood by linguistics that syntactically related words express semantic concepts [26]. By using syntactic bi-grams we are then already incorporating into the representation some basic semantics. We further enrich the representation by introducing some more semantic knowledge to help with the specific vocabulary. A list of proteins and genes was extracted from the SwissProt Protein Knowledgebase[2]. The words found in this list were replaced in our representation by a lexical marker (the word `geneprot`). A list of words commonly used in the genetic bibliography to denote interactions was borrowed from Temkin and Gilder work [23] and included as facts in one of the experiments with the relational representation.

Our experiments were done on a corpus created by, the CADERIGE project[3]. The examples consist of only one sentence, which were automatically selected from MedLine abstracts with a query *Bacillus subtilis transcription*. The sentences were then pre-filtered to keep only those 932 that contain at least two names of either genes or proteins. The remaining sentences were manually categorized as positive or negative according to whether they describe or they do not describe a genomic interaction. The resulted was a balanced dataset with 470 positive and 462 negative examples.

Some earlier work done on this corpus is presented in [15]. It reports the recall and precision result obtained by the C4.5 algorithm and a variation of Naïve Bayes (NB) algorithm (that specializes it for the case of short documents). The attributes were all

---

[2] Swiss-Prot is an annotated protein sequence database available on-line at http://ca.expasy.org/sprot/. Among all the information provided is a "Short description of entries in Swiss-Prot" from which we extracted the names of proteins and genes.

[3] CADERIGE Project, http://caderige.imag.fr/

words after stemming, stop word removal and some filtering using Information Gain. The best results were 84.12% recall and 87.89% precision with the variation of NB.

While modifying the representation of the corpus for our experiments, around 5% of the examples were lost due to failure of the parser on those sentences. Our final dataset contains 885 examples, being 440 positive and 445 negative.

4 Syntactic Representation and Experiments

In this section we present an example of the analysis performed by the Link Parser [22], the links it recognized in our collection and how they are used in the text representation. We then present the experiments that we performed and the results that we obtained when using that representation to learn a classifier for the positive examples of our dataset.

The Link Parser was selected for specifically providing the relation between words in the sentence by establishing a link between them In order to create a syntactic representation we ran the parser on each sentence of the data collection, identified some syntactic links, such as the object of a verb, and we built syntactic bi-grams with the linked words. Out of the many links identified by the parser, we only took into consideration those ones that we believe could help enrich our representation:

- A and AN: link an adjective or a noun (respectively) to the noun it modifies.
- Ss: links the head of a noun phrase to the verb to which the phrase is the subject.
- Os: links the head of a noun phrase to the verb to which the phrase is the object.
- Pa: links forms of the verb "have" to a participle verb.
- Mg: links nouns with present participles
- MVp and J (or Jp): MVp links the verb to a preposition at the beginning of a prepositional phrase, and J (or Jp) links that preposition to the noun that is head of the noun phrase inside the prepositional phrase. We established the M relation, which links the verb in a MVp to the noun in a corresponding Js.

Figure 1 shows all the links we identified among the set of links returned by the Link Parser for the first sentence of our collection.[4] From this analysis, the following syntactic bi-grams could be built: spo0a_mutant, s210a_mutant, spoiie_activation, promoter_activation, mutant_exhibited, it_was, exhibited_change, exhibited_wild-type, defective_activation, wild-type_binding, was_defective.

The previous are all the syntactic bi-grams we built. In the following experiments only some of them were used at a time, according to the kind of link we were permitting (e.g. when representing noun phrases only the A and AN links were permitted). The type of the link and the morphological information (i.e. which words are nouns, adjectives and verbs, which is also provided by the Link Parser) were not

[4] The Link parser returns for each sentence several different links sets, and a cost vector value associated to each of them. Only the highest ranked links set was used in these experiments.

included in the representation. We are planning to include this information in our future work. Some of the previous syntactic bi-grams were modified because they contained a gene or protein name from the SwissProt list. Thus s210a_mutant was replaced by geneprot_mutant. Unfortunately spo0a and spoiie were not found in the list and therefore were not replaced by our lexical marker.

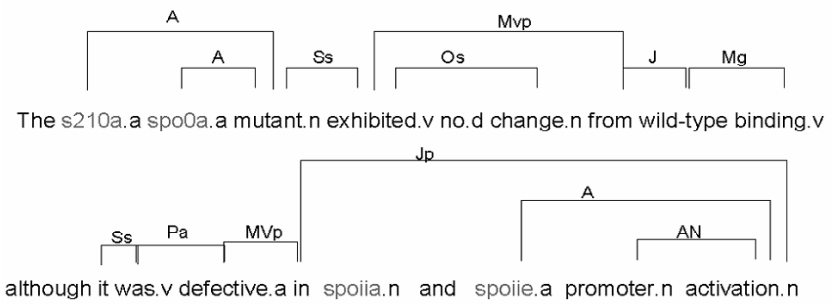

Figure 1. Links identified for the first sentence of our collection.

It must be noticed that the extra effort of parsing is reduced in our experiments since the abstracts were pre-filtered and only few sentences possibly containing an interaction between genes/proteins were kept. In larger datasets a more efficient parsing approach could be taken, as for example the partial parsing within a fixed size window presented by Jacquemin [28].

After learning and evaluating classifiers for the different representations, the results were compared using Accuracy, Precision, Recall and F1-measure. Given a contingency table containing TP (True Positives), FP (False Positives), FN (False Negatives) and TN (True Negatives), the previous measures are defined as:

- Accuracy = (TP + TN) / (TP + TN+ FP +FN)
- Precision (Pr) = TP / (TP + FP)
- Recall (Re) = TP / (TP + FN)
- F1 = 2*Pr*Re / (Pr + Re)

As a baseline we use the BOW representation (all the words that appear in any of the links.) We compare its performance with the one obtained when using it together with some or all of the recognized syntactic bi-grams. We differentiate the Noun Phrases representation, the Subject and Object representation, the Prepositional Phrases representation and the representation using all the links together.

Previous to the classification, a subset of the total set of features was selected using the information gain metric. This filter kept some of the syntactic bi-grams among the most discriminating features, giving a first confirmation of their usefulness for the classification task [1]. Among the selected syntactic bi-grams, several include our lexical marker, even when by itself it was not selected as discriminant when using the BOW representation. Some examples of these syntactic bi-grams are `geneprot-protein`, `mutant-geneprot`, `encodes-geneprot`. Some examples of other

syntactic bi-grams also kept after filtering are `bacillus-subtilis, indicated-analysis, protein-encodes.`

Table 1 shows the Accuracy, Precision, Recall and F1-measure obtained by the Naïve Bayes Simple and the Support Vector Machine learners of the Weka Package. The experiments were performed using the default parameters for each learner and the number of features that resulted in the best accuracy in preliminary experiments.

Table 1. Averaged Accuracy, Precision, Recall and F1-measure in 10 runs of 10-fold cross-validation. The results in bold denote a statistically significant increase over the BOW (first column). The use of italic denotes a statistically significant decrease with respect to the BOW.

Learning Algorithm	Performance measure	Words in links	Noun Phrases Bi-grams	Subject and Object Bi-grams	Prep. Phrases Bi-grams	All the syntactic Bi-grams
Naïve Bayes 500 features	Accuracy	0.81	0.81	0.81	*0.80*	**0.83**
	Precision	0.81	*0.80*	0.81	**0.82**	**0.83**
	Recall	0.82	0.83	*0.81*	*0.78*	**0.85**
	F1-measure	0.82	0.81	*0.81*	*0.80*	**0.84**
SVM 500 features	Accuracy	0.77	0.77	0.77	**0.78**	**0.81**
	Precision	0.78	**0.79**	0.78	**0.80**	**0.84**
	Recall	0.76	0.76	**0.77**	0.75	**0.79**
	F1-measure	0.77	0.77	0.77	0.77	**0.81**

In most cases, the results do not show statistically significant difference with respect to the BOW when using noun phrase bi-grams and subject/object bi-grams to enrich the BOW representation. When using prepositional phrase bi-grams the results are mixed depending on the measure of choice. However, when using all the syntactic bi-grams together, the results show a consistent statistically significant increase for the four considered measures. That indicates that at least two kinds of the used syntactic bi-grams are relevant to the classification when combined.

We also performed preliminary experiments using the Decision Tree Learner from the Weka Package. Although the pattern of increased accuracy when using all the syntactic bi-grams holds, the values were only in the 70% range. This low performance was already noticed by Ould [16] who argued that it might be due to the sparseness of the representation.

5 Relational Representation and Experiments

As noted in the previous experiments, the syntactic relation between some words in a sentence seems to be relevant to the sentence selection task we are performing. It is

natural then to think of a relational representation that can capture these relations. This new representation can then be used by a relational learner system, exploiting the advantages of this kind of systems [7]. Among others, predicates to help the classification can be easily defined, and relations among three words could be discovered (as two bi-grams with a transitive relation).

In order to obtain this relational representation, the same links obtained for the syntactic representation were used and relations were built between the linked words. The predicate "link(s,w1,w2)" used in our representation expresses that in the sentence s1 there is a relation between the two words w1 and w2 as indicated by the presence of a link found by the parser.

Given the same sentence from figure 1, the following relations could be built:

```
link(s1,mutant,s210a),                  link(s1,was,it),
link(s1,mutant,geneprot),               link(s1,defective,was),
link(s1,exhibited,mutant),              link(s1,activation,defective),
link(s1,change,exhibited),              link(s1,activation,spoiie),
link(s1,wildtype,exhibited),            link(s1,activation,promoter).
link(s1,binding,wildtype),
```

This relational representation introduces the syntactic relations of the words in the sentences. This representation was compared in our experiments with a baseline using propositional logic denoting whether a word occurs or not in a sentence but missing the information of the relations between words, which is equivalent to the BOW. Instead of physically creating the propositional representation, we simulated it by defining the predicate *lexexist* that represents the presence of a word in a particular sentence and the fact that the word is involved in a link (some words, as for example the articles, were not included in any of the considered links). For this, we use the unbounded variable "_" that will take any value. The predicate definition is:

```
lexexist(S,W) :- link(S,W,_).
lexexist(S,W) :- link(S,_,W).
```

We also compared the previous relational representation performance with the one obtained when adding extra background knowledge. For this purpose, we added to the representation the list of words denoting interactions presented in [23]. Each of the words in the list was given as a parameter of the fact *interaction*. The predicate *interacts* was defined representing the fact that a particular word in a sentence is linked to an interaction word.

```
interaction(initiate).
interaction(stimulate).      interacts(S,W):-link(S,W,I),interaction(I).
interaction(regulate).       interacts(S,W):-link(S,I,W),interaction(I).
. . .
```

In this way we performed three different experiments by providing Aleph with only one file containing the genomic information in a relational representation and by instructing it on what kind of rules could be learned. Figure 2 shows some of the rules learned when allowing words, syntactic links and interactions. In the following we analyze those rules.

Our first observation is that few training examples are covered by each rule, the first one covering 29 positive examples and 1 negative example, and thereafter dropping to 16 examples (approximately 2% of the training examples.) The second

observation is that geneprot is already chosen in the third rule, marking the usefulness of having replaced the technical vocabulary by this lexical marker. However, the gene/protein *sigmak,* which was not found in the Swissprot list and therefore was not replaced, seems to be very discriminating in this dataset. This gives us the hint that we might want to consider different levels in a hierarchy of genes/proteins instead of a single-level list. We will consider this in our future work.

```
[Rule1]pos(S):-lexexist(S,sigmak).[29;1]
[Rule2]pos(S):-lexexist(S,_expression),lexexist(S,fusion).[16;0]
[Rule3]pos(S):-lexexist(S,geneprot),lexexist(S,vivo).[15;1]
[Rule4]pos(S):-link(S,processing,B).[14;1]
[Rule5]pos(S):-link(S,B,geneprot),link(S,B,bdependent).[14;1]
[Rule6]pos(S):-link(S,geneprot,transcription),link(S,is,B).[13;0]
[Rule7]pos(S):-link(S,show,B),interacts(S,C).[12;1]
. . . .
[Rule12]pos(S):-link(S,geneprot,protein).[11;1]
[Rule13]pos(S):-link(S,geneprot,gene).[10;1]
. . . .
[Rule17]pos(S):-interacts(S,transcription),interacts(S,geneprot).[11;0]
. . . .
```

Figure 2 Some rules learned when allowing words, syntactic links and interactions.

Starting with rule 4 the links help to discover discriminating rules. In rule 4 we notice that it is not always a pair of words that is important, but as in this case, the fact that the word *processing* is linked to another one makes it discriminating of the class. In rule 5 there is a word linked to both *geneprot* and *bdependent*, being the double link relevant for the discriminating rule. In rules 6, 12 and 13 we find the pairs *geneprot_transcription, geneprot_protein* and *geneprot_gene* being the most discriminating after various other rules have been applied.

Finally, we observe the use of the predicate *interacts* in rule 7 where it establishes that there is a word linked to an interaction term. In rule 17 it establishes that both *transcription* and *geneprot* are linked to an interaction word.

When running the experiments, around 80 rules were learned. They were used without any pruning on the test sets. The average Accuracy, Precision, Recall and F1-measure obtained after 10 runs of 10-fold cross-validation for the 3 experiments are shown in table 2.

We observe a higher accuracy with respect to the results obtained by Naïve Bayes and Support Vector Machine, even when the representation equivalent to the BOW is used. We explain this as the result of two main characteristics:

1. the flexibility of the relations: letting the learner choose what are the important parts of a relation, as if only one or both words were fixed, makes them more flexible than the pre-fixed phrases used in the syntactic representation. The relational representation also gives the opportunity of bridging two non-linked words by the mean of a third one linked to both (see rule 5 in the examples).
2. the sparseness of the collection: having many rules, each one adjusted to the few examples it covers, seems to be beneficial for this short sentences collection with very sparse vocabulary.

Table 2. Averaged Accuracy, Precision, Recall and F1-measure in 10 runs of 10-fold cross-validation running Aleph. The results in bold denote a statistically significant increase over the basic relational representation (Words and Links, in the central column). The use of italic denotes a statistically significant decrease with respect to the basic relational representation.

Learning Algorithm	Performance measure	Only Words	Words and Links	Words, Links and Interactions
Aleph	Accuracy	*0.90*	0.93	**0.94**
	Precision	*0.94*	0.95	0.96
	Recall	*0.87*	0.92	**0.93**
	F1-measure	*0.90*	0.94	**0.94**

Similar to the previous results, we observe a statistically significant increase in the performance (according to the four considered measures) when the links are used in the representation. That marks once again the importance of a syntactic representation.

Finally we also observe a statistically significant increase in accuracy and recall when adding some semantic background knowledge to the representation, i.e. the interactions list (the last column in table 3). This increase in the recall was not at the expense of the precision.

6 Conclusions and Future Work

In this paper we have presented the problem of sentence selection from a genetic corpus and how we envisioned the contribution of semantic and syntactic knowledge in this task. We directly introduced semantic knowledge in the representation by replacing the words found in a list of genes/proteins. Basic semantic knowledge was also incorporated in the representation by mean of syntactic relations. This was accomplished extending the set of features with bi-grams obtained from a syntactic parser. We have empirically showed that this knowledge is useful for sentence selection from this genetic corpus when using several different machine learning methods. We have also shown that the relational learner Aleph performs better than the other algorithms tried, even when an analogous to the BOW was used. The use of the syntactic information in the relational representation highly significantly improved the performance (e.g. an increase of 0.04 for the F1 measure with respect to the representation using only words). This confirmed the results previously obtained with other algorithms using the syntactic bi-grams. Adding extra semantic knowledge to this representation by identifying the interaction words further helped with the classification by improving the recall with no decrement of the precision.

In the future we plan to extend the use of semantic background knowledge to include hierarchies of genes/proteins. One possible source for that could be the

publicly available Gene Ontology. We also plan to extend the use of syntactic knowledge by differentiating the links according to the kind of relation they denote (noun phrases, subject, etc.) and introducing morphological information (whether a word is a noun, an adjective, a verb, etc.) We would also like to use the relational representation in state of the art classification methods by transforming the predicates into features in a vector space or probabilistic model. We plan to do this by applying propositionalization as presented by Kramer [9]. Finally, we plan to try this approach on a similar but larger dataset in the genetic abstracts context, as well as on a different domain on Legal documents, the HOLJ Corpus created by Hachey and Grover [29].

Acknowledgements

This work is supported by the Natural Sciences and Engineering Council of Canada and the Ontario Centres of Excellence.

References

1. Caropreso, M.F., Matwin, S, and Sebastiani, F. "A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization**".** In Amita G. Chin (ed.), *Text Databases and Document Management: Theory and Practice*, Idea Group Publishing, Hershey, US, 2001, pp. 78-102.
2. W. W. Cohen and Y. Singer (1999): Context-sensitive learning methods for text categorization in ACM Trans. Inf. Syst. 17(2): 141-173 (1999).
3. S. T. Dumais, J. Platt, D. Heckerman, and M. Sahami. Inductive learning algorithms and representations for text categorization. In G. Gardarin, J. C. French, N. Pissinou, K. Makki, and L. Bouganim, editors, Proceedings of CIKM-98, 7th ACM International Conference on Information and Knowledge Management, pages 148{155, Bethesda, US, 1998. ACM Press, New York, US.
4. J. L. Fagan. Experiments in automatic phrase indexing for document retrieval: a comparison of syntactic and non-syntactic methods. PhD thesis, Department of Computer Science, Cornell University, Ithaca, US, 1987.
5. J. Furnkranz. A study using n-gram features for text categorization. Technical Report TR-98-30, Oesterreichisches Forschungsinstitut Artificial Intelligence, Wien, AT, 1998.
6. J. Furnkranz, T. M. Mitchell, and E. Rilo®. A case study in using linguistic phrases for text categorization on the WWW. In Proceedings of the 1st AAAI Workshop on Learning for Text Categorization, pages 5{12, Madison, US, 1998.
7. J. Furnkranz. Inductive Logic Programming (a short introduction and a thesis abstract).
8. M. Goadrich, L. Oliphant and J. Shavlik (2004). Learning Ensembles of First-Order Clauses for Recall-Precision Curves: A Case Study in Biomedical Information Extraction. *Proceedings of the Fourteenth International Conference on Inductive Logic Programming*, Porto, Portugal.
9. S. Kramer. Relational Learning vs. Propositionalization. PhD. Thesis, Vienna University of Technology, Vienna, Austria, 1999.
10. D. D. Lewis and W. B. Croft. Term clustering of syntactic phrases. In Proceedings of SIGIR-90, 13th ACM International Conference on Research and Development in Information Retrieval, pages 385{404, Bruxelles, BE, 1990.

11. Lewis D D, "Representation and Learning in Information Retrieval", Ph.D. dissertation, University of Massachusetts, 1992.
12. D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In N. J. Belkin, P. Ingwersen, and A. M. Pejtersen, editors, Proceedings of SIGIR-92, 15th ACM International Conference on Research and Development in Information Retrieval, pages 37-50, Kobenhavn, DK, 1992. ACM Press, New York, US.
13. Mladenic, D. and Grobelnik, M. Word sequences as features in text learning. Proceedings of *ERK-98, the seventh Electrotecnical and Computer Science Conference* (pp. 145-148). Ljubljana, Slovenia. 1998
14. M. Mitra, C. Buckley, A. Singhal, and C. Cardie, "An Analysis of Statistical and Syntactic Phrases". *5TH RIAO Conference, Computer-Assisted Information Searching On the Internet*, 200-214, 1997.
15. Nédellec C., Ould Abdel Vetah M., and Bessières P., "Sentence Filtering for Information Extraction in Genomics: A Classification Problem," *Proceedings of the International Conference on Practical Knowledge Discovery in Databases* (PKDD'2001), pp. 326–338, Springer Verlag, LNAI 2167, Freiburg, September, 2001.
16. Ould, M. Apprentissage Automatique Applique a l'Extraction d'Information a Partir de Textes Biologiques. PhD Thesis. L'Universite Paris-Sud. France. 2005
17. Ould, M., Caropreso, F., Manine, P., Nedellec, C., Matwin, S., "Sentence Categorization in Genomics Bibliography: a Naïve Bayes Approach", Informatique pour lèanalyse du transcriptome, Paris, 2003.
18. Soumya Ray, Mark Craven. Representing Sentence Structure in Hidden Markov Models for Information Extraction. Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI-2001)
19. Sam Scott, Stan Matwin. Feature Engineering for Text Classification. Proceedings of ICML-99, 16th International Conference on Machine Learning, 1999.
20. Siolas, G. Modèles probabilistes et noyaux pour l'extraction d'informations à partir de documents. Thèse de doctorat de l'Université Paris 6. July 2003.
21. Srinivasan, A. The Aleph Manual. 1993. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph_toc.html
22. D. Sleator and D. Temperley. 1991. Parsing English with a Link Grammar. Carnegie Mellon University Computer Science technical report CMU-CS-91-196, October 1991.
23. Temkin JM, Gilder MR. Extraction of protein interaction information from unstructured text using a context-free grammar. Bioinformatics. 19(16):2046-53, 2003.
24. Ian H. Witten and Eibe Frank. Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.
25. Sarah Zelikovitz and Haym Hirsh. Improving Text Classification with LSI Using Background Knowledge. Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management. 2001
26. Fillmore, Charles J. The Case for Case. In Bach and Harms (Ed.): Universals in Linguistic Theory. New York: Holt, Rinehart, and Winston, 1-88. 1968.
27. Maarek, Y., Berry, D.M. & Kaiser, G.E.: GURU: Information Retrieval for Reuse, in P.Hall (ed.), Landmark Contributions in Software Reuse and Reverse Engineering, 1994
28. Christian Jacquemin. *What is the tree that we see through the window: A linguistic approach to windowing and term variation*. Information Processing and Management, 32(4):445--458, 1996.
29. Ben Hachey and Claire Grover. Sequence Modelling for Sentence Classification in a Legal Summarisation System. In: *Proceedings of the 2005 ACM Symposium on Applied Computing*, 2005.