Extracting Protein-Protein Interactions from Text using Rich Feature Vectors and Feature Selection

Sofie Van Landeghem, Yvan Saeys and Yves Van de Peer
Department of Plant Systems Biology, VIB, 9052 Gent, Belgium
Department of Molecular Genetics, Ghent University, 9052 Gent, Belgium
yves.vandepeer@psb.ugent.be

Bernard De Baets
Department of Applied Mathematics, Biometrics and Process Control,
Ghent University, 9000 Gent, Belgium

Abstract

Because of the intrinsic complexity of natural language, automatically extracting accurate information from text remains a challenge. We have applied rich feature vectors derived from dependency graphs to predict protein-protein interactions using machine learning techniques. We present the first extensive analysis of applying feature selection in this domain, and show that it can produce more cost-effective models. For the first time, our technique was also evaluated on several large-scale cross-dataset experiments, which offers a more realistic view on model performance.
During benchmarking, we encountered several fundamental problems hindering comparability with other methods. We present a set of practical guidelines to set up a meaningful evaluation.
Finally, we have analysed the feature sets from our experiments before and after feature selection, and evaluated the contribution of both lexical and syntactic information to our method. The gained insight will be useful to develop better performing methods in this domain.

1 Introduction

Results of genetic studies are published on a daily basis and appear in scientific articles, accessible through online literature services like PubMed (http://pubmed.gov). Over 17 million citations are currently available through PubMed and this resource is still growing exponentially. Fully automated systems that extract biological knowledge from text have thus become a necessity.

Many approaches have been proposed to extract biological information from research articles. The first methods mainly relied on co-occurrence of biological entities. They would classify two proteins as interacting when mentioned in the same sentence, or when their co-occurrence in an abstract is statistically overrepresented (Ding et al., 2002; Rebholz-Schuhmann et al., 2007). Typically, a co-occurrence based technique exhibits high recall, but low precision.

A second important set of techniques apply patterns or rules which are usually hand-crafted, allowing the method to obtain high precision while recall typically drops. The RelEx system uses three rules in combination with information derived from dependency graphs (Fundel et al., 2007). Dependency parsing uses graph topology to represent syntactic relations between individual words of the sentence (Figure 1).

Finally, machine learning techniques use training data to construct a model, which is then applied to a test set to predict protein-protein interactions (PPIs). To extract meaningful features for the model construction, dependency parsing is often used. Both global context, such as the root of the tree, and local context, such as the parent of a particular node, can be taken into account (Katrenko and Adriaans, 2007). Erkan et al. (2007) extract sentences where two proteins co-occur with an interaction word. Extracted features include the interaction words themselves and the parents of the proteins in the dependency graph. Kim et al. (2008) present a walk kernel, consisting of patterns of two vertices and their intermediate edge (*vertex-walk* or *v-walk*), as well as sequences of two edges and their common vertex (*edge-walk* or *e-walk*), extracted from the shortest path between two proteins in the graph. They also conclude that a feature-based approach performans better than direct kernel techniques. A

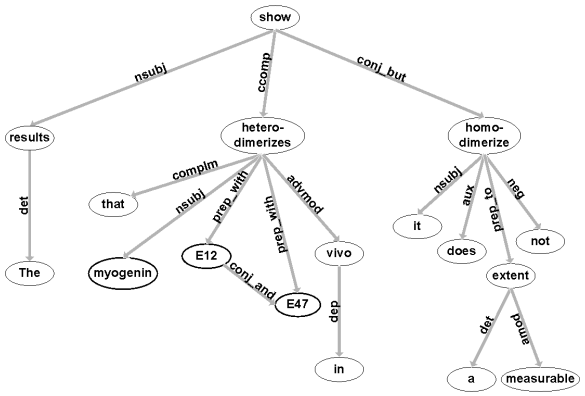

Figure 1: The dependency graph for the sentence 'The results show that myogenin heterodimerizes with E12 and E47 in vivo, but it does not homodimerize to a measurable extent'

more reduced feature set is used by Fayruzov et al. (2008), taking mainly syntactic information into account. Finally, deep syntactic parsing can be combined with a shallow dependency parser to create a more accurate model (Saetre et al., 2008).

A hybrid approach is also possible, with handcrafted rules forming the basis for different kernels, which are then aggregated by linear combination (Giuliano et al., 2006).

The application of feature selection in the domain of natural language processing is relatively new. Jiang and Zhai (2007) have investigated the type of features that are potentially useful for relation extraction in general. Feature selection techniques have also been employed for the task of text classification (Wang et al., 2008). However, to the best of our knowledge, this paper presents the first study of applying rich feature vectors in combination with feature selection for protein-protein interaction extraction. Our study using fully automated feature selection methods is clearly different to previous work concerning manually selected varying sets of features (Katrenko and Adriaans, 2007). Furthermore, it is the first time that a broad cross-corpus study has been conducted, offering an evaluation in a more realistic setup than cross-validation on a single dataset.

2 Benchmarking protein-protein interaction extraction techniques

While studying state-of-the art systems that extract PPIs from text, it became clear that this field is struggling with a heterogeneous collection of datasets and evaluation methods (Van Landeghem et al., 2008). In this section we will analyse these problems and introduce practical guidelines in order to improve comparability between extraction methods in this domain.

2.1 Benchmark datasets

The development of standard benchmarking datasets is a step forward towards meaningful comparisons between different information extraction techniques. For genetic interaction extraction, such corpora include AIMed (Bunescu et al., 2005), Bioinfer (Pyysalo et al., 2007), HPRD50 (Fundel et al., 2007), IEPA (Ding et al., 2002) and LLL (Nedellec, 2005). These corpora all have slightly different scopes, ranging from protein-gene interactions concerned with *Basilicus subtilis* transcription to human protein-protein interactions. Recently, conversion software has been introduced to convert these different datasets into a common data format, forming a rich corpus with a broad range of genetic interactions (Pyysalo et al., 2008). Another important resource is the Biocreative initiative, which aims to provide a framework for the construction of suitable 'Gold standard' datasets, applicable for text-mining systems in biology (Hirschman et al., 2005). Finally, the Genia corpus can be useful for benchmarking various subtasks of text-mining algorithms (Kim et al., 2008). It has been shown by Pyysalo et al. (2008) that the choice of benchmark dataset can drastically influence extraction performance. It is therefor advisable to evaluate algorithms on a collection of different corpora.

2.2 Instance extraction

Even when evaluating on the same dataset, different preprocessing steps can yield a varying set of instances. Homodimers, which are self-interacting proteins, are sometimes discarded from the dataset. A similar problem is raised by nested annotations in the corpus which may or may not be discarded, influencing the final number of instances in the dataset. To construct negative examples, the closed world assumption is usually adapted, stating that no interaction exists between two entities when there is no annotated evidence. We believe it is best to always clearly indicate which rules were applied for instance extraction, and to report on the number of retrieved instances.

2.3 The extraction task

For the extraction of protein-protein interactions, it is often assumed that the proteins in the text are known a priori. However, when performing the named entity recognition (NER) step automatically, errors will propagate and cause a drop in performance. We believe that the NER step is a different subtask which should be examined and evaluated separately. Similarly, parse trees can be automatically constructed or manually verified. In our opinion, parsing input sentences in a fully-automated fashion is necessary to provide a scalable method, applicable to large datasets.

2.4 Cross validation

Ideally, abstracts for the testing phase should be completely hidden during training. Saetre et al. (2008) pointed out that some evaluations exhibit an artificial boost of performance by using features from the same sentence in both training and testing steps of the machine learning process. This effect is caused by the fact that one sentence in the dataset yields C_n^2 distinct instances, where n is the number of proteins in the sentence and each instance represents a pairwise combination of proteins. It is therefore best to modify the regular cross-validation approach to include all instances from one sentence in the same fold, or even define folds consisting of complete abstracts.

2.5 Counting true positives

Finally, the definition of a true positive is ambiguous in the text-mining domain. Each pair of proteins is usually considered as an individual instance, evaluated independently of others. However, two distinct instances may be expressing the same interaction. Thus, to extract a true protein-protein interaction, retrieving one such instance suffices. The latter evaluation technique naturally exhibits higher recall. Even though this technique is useful for benchmarking complete information retrieval systems, we feel that instance-level evaluation is more representative for the task of extracting interactions between named entities from individual sentences.

3 Methods

In our study, we used all the datasets that have been converted to a common data format by Pyysalo et al. (2008), with the exception of Bioinfer. This corpus is relatively new, and contains extensive annotations of proteins and interactions. For example, the words *alpha 5 integrins* are annotated as being a protein reference in the construct *alpha 5 and beta 1 integrins*. Our extraction method assumes a protein is mentioned as a contiguous stream of tokens, which are replaced by the token *PROT* for all training and testing instances in the dataset. This is why we exclude Bioinfer from further analysis and focus on the other four corpora: AIMed, HPRD5, IEPA and LLL. However, we plan on resolving these issues in the future, as well as considering more corpora to test our method on, such as theBiocreative and Genia datasets.

dataset	positive	negative	total
AIMed	1000	4670	5670
HPRD50	163	270	433
IEPA	335	482	817
LLL	164	166	330
All	1662	5588	7250

Table 1: Number of instances in the four corpora

3.1 Dataset preprocessing

In preparing the datasets we excluded homodimers, as not all corpora support homodimer annotation. Sentences with at least two co-occurring proteins are selected for further processing, creating a distinct instance in the dataset for each pairwise combination of proteins in the sentence. Nested annotations are taken into consideration in all datasets. We apply the closed-world assumption to create negative instances, assuming there is no interaction between two proteins when there is no annotated evidence. For AIMed, the abstracts included in the corpus that contain no interactions are also taken into account. The resulting numbers of positive and negative instances are shown in Table 1.

3.2 Extracting rich feature vectors

Our feature extraction method uses syntactic and lexical patterns derived from the shortest path between two proteins in the dependency graph. These graphs are built automatically using the Stanford parser (de Marneffe et al., 2006). The shortest path in the graph is scanned for all subsequent vertices and their intermediate edge (*v-walk*), as well as all subsequent edges and their common vertex (*e-walk*), taking into account both syntactic and lexical properties of the walks (Table 2, upper four rows). To traverse this path,

Type	Features
Lex v-walk	heterodimer nsubj PROT, heterodimer prep PROT
Syn v-walk	VBZ nsubj PROT, VBZ prep PROT
Lex e-walk	nsubj heterodimer prep
Syn e-walk	nsubj VBZ prep
BOW	PROT, a, and, but, doe, extent, heterodimer, homodimer, in, it, measur, not, result, show, that, the, to, vivo, with
Lex root	heterodimer
Syn root	VBZ

Table 2: Syntactic and lexical features for the pair of proteins (Myogenin, E12) from Figure 1

we go up from the first protein to the root by inverting the original direction of the edges, and go down again from the root to the second protein. To improve generalization of lexical information by the classifier, we apply the Porter stemming algorithm (Porter, 1980). Protein names are substituted by the token *PROT* to enable the classifier to learn interaction patterns, disregarding the specific proteins involved. The walk features are augmented with a bag-of-words (BOW) approach in combination with the stemming algorithm, to capture critical information outside the shortest path of the dependency graph (Table 2, fifth row). This bag-of-words approach will give rise to quite some irrelevant features, which is one of the reasons why we will apply fully automated feature selection techniques after feature extraction. Syntactic and lexical information from the root node are stored as separate features (Table 2, last two rows). Finally there is a numeric feature indicating the length of the shortest path.

All features are encoded by defining one specific numeric feature for each syntactic or lexical pattern, storing the number of times that pattern occurs in the sentence or its derived dependency graph. This encoding technique results in sparse feature vectors and high-dimensional feature sets. For example, when using cross-validation on the AIMed dataset, which is the richest corpus of the four, over 14.000 numeric features are extracted from the training set.

3.3 Classification model

For our experiments, we made use of a linear support vector machine classifier (SVM, Boser et al. (1992)). The SVM is a data-driven method for solving two-class classification tasks, based on the concept of large margins, and is known to perform well in high-dimensional spaces (Saeys et al., 2007). We used the Weka[1] implementation of LibSVM, with an internal 5-fold cross-validation loop on the training portion of the data to determine the optimal C-parameter.

3.4 Feature selection

Feature selection (FS) techniques are a class of dimensionality reduction techniques that aim at identifying a subset of the most relevant features from a potentially large initial set of features. In contrast to other reduction techniques such as methods based on projection, FS techniques only select a subset of the original set of features, preserving the original semantics.

Advantages of applying feature selection include its potential to improve generalization performance (by avoiding overfitting), faster and more cost-effective models and gaining a deeper insight into the underlying processes that generated the data. Depending on the interaction with the model, three classes of FS techniques can be defined (Guyon and Elisseeff, 2003). In this work, we will focus on the class of *filter* methods, which perform feature selection by looking only at the intrinsic properties of the data, thus being independent of the classification model used afterwards. Advantages of this class of methods include their scalability to high-dimensional datasets (such as the ones we deal with in this work) and their speed. An in-depth analysis of the different classes of FS techniques, as well as their application in bioinformatics can be found in (Saeys et al., 2007).

The filter method we used in this work is based on the information-theoretic concept of *gain ratio*. A given set of training patterns S can be regarded as a distribution over the class labels, and its entropy can be calculated as

$$H(S) = -\sum_{i=1}^{s} p(c_i) \log_2 p(c_i)$$

where $p(c_i)$ denotes the proportion of patterns in S belonging to class c_i. The *information gain* $IG(S, D)$ then represents the expected reduction in entropy (uncertainty) when splitting on a feature D, and can be calculated as

$$IG(S, D) = H(S) - H(S|D)$$

[1]Available at `http://www.cs.waikato.ac.nz/ml/weka/`

$$= H(S) - \sum_{j \in V(D)} \frac{|S_j|}{|S|} H(S_j)$$

where $V(D)$ denotes the possible values for feature D and S_j is the subset of S for which feature D has value j.

To adjust the bias towards features with a larger number of possible values, the information gain should be scaled by the entropy of S with respect to the values of feature D, resulting in the *gain ratio* $GR(S, D)$:

$$GR(S, D) = \frac{IG(S, D)}{-\sum_{j \in V(D)} \frac{|S_j|}{|S|} \log_2 \frac{|S_j|}{|S|}}$$

Applying the gain ratio to every feature in the dataset gives an estimate of the feature's importance, and all features can be ranked from most influential to least influential by sorting their gain ratios. The top k features can then be used to construct a simplified classifier.

3.5 Evaluation strategy

For benchmarking our PPI extraction method, we use instance-level evaluation. We have applied regular 10-fold cross-validation (*Instance CV*), as well as the modified version of 10-fold cross validation, with folds consisting of complete abstracts (*Abstract CV*). We use the gold-standard protein annotations which are available for all corpora. As not all datasets provide annotation of the direction of interactions, we consider interactions to be symmetric. As a performance measure, the F-measure is used, which is common practice in this domain. It is defined as the harmonic mean between precision (p), which expresses how many of the predictions are correct, and recall (r), which expresses how many of the true interactions are correctly predicted.

In addition to training and testing on a single dataset using CV, we have conducted a large-scale evaluation using all four corpora. The rationale for this approach was to analyse the scalability of our approach. Most datasets have been constructed using specific keywords (e.g. LLL : *Bacillus subtilis transcription*), which causes a bias in the classifier towards this particular domain. However, when using features from three different datasets and testing on an independent dataset, we obtain a more diverse model, which is more representative for the real world task of extracting interactions from various PubMed abstracts. We conducted four experiments, each time using a different corpus as test set, while including the other three in the training data. To the best of our knowledge, this is the first time such a large-scale cross-dataset comparison has been conducted.

	Corpus	p	r	F
Inst. CV	AIMed	0.66	0.58	0.62
	HPRD50	0.71	0.71	0.71
	IEPA	0.74	0.69	0.71
	LLL	**0.79**	**0.84**	**0.82**
Abstr. CV	AIMed	0.49	0.44	0.46
	HPRD50	0.60	0.51	0.55
	IEPA	0.64	0.70	0.67
	LLL	**0.72**	**0.73**	**0.73**
Co-occ.	AIMed	0.18	1.00	0.30
	HPRD50	0.38	1.00	0.55
	IEPA	0.41	1.00	0.58
	LLL	**0.50**	1.00	**0.66**

Table 3: Evaluation on the four individual datasets

4 Results

4.1 Individual dataset evaluation

As a baseline, we evaluated our method on all datasets separately, using 10-fold instance CV (Table 3, first row). We then evaluated the method using the modified version of 10-fold CV, clustering instances originating from the same sentence in the same fold (Table 3, second row). For the evaluation on AIMed, the original abstract splits were used (Bunescu et al., 2005). We noticed an artificial boost of performance of up to 0.16 F-measure when using instance CV. In both experiments we find a significant difference in F-measure between the best results (LLL) and the worst results (AIMed), ranging between 0.20 and 0.27 F-measure. To demonstrate the inherent differences between the four individual datasets, we have included the results for a simple co-occurrence based technique, assigning a

	Method	p	r	F
AIMed abstr. cv	Rich features	0.49	0.44	0.46
	Fundel et al. (2007)	0.40	0.50	0.44
	Giuliano et al. (2006)	0.61	**0.57**	**0.59**
	Saetre et al. (2008)	**0.64**	0.44	0.52
AIMed inst. cv	Rich features	0.66	0.58	0.62
	Erkan et al. (2007)	0.60	0.61	0.60
	Fayruzov et al. (2008)	0.41	0.50	0.45
	Katrenko and Adriaans (2007)	0.45	**0.68**	0.54
	Saetre et al. (2008)	**0.78**	0.63	**0.70**
LLL inst. cv	Rich features	0.79	0.84	**0.82**
	Fayruzov et al. (2008)	0.72	**0.86**	0.78
	Fundel et al. (2007)	**0.85**	0.79	**0.82**

Table 4: Comparison to existing techniques for individual datasets.

	features	p	r	F	syn	lex	bow
AIMed	14.000	**0.49**	**0.44**	**0.46**	15	61	20
	10.000	0.48	0.43	0.45	16	61	19
	7.500	0.41	0.41	0.41	17	61	18
	5.000	0.44	0.38	0.41	16	59	21
HPRD50	2.600	0.60	0.51	0.55	21	44	29
	1.500	0.51	0.60	0.55	23	48	23
	750	0.57	0.61	0.59	23	52	20
	500	**0.61**	**0.62**	**0.61**	23	45	28
	250	0.58	0.36	0.45	23	51	23
IEPA	6.900	**0.64**	0.70	0.67	17	49	30
	5.000	0.61	0.71	0.65	14	43	38
	2.500	0.63	**0.75**	**0.68**	22	51	21
	1.000	0.54	0.66	0.60	20	42	34
LLL	1.600	0.72	0.73	**0.73**	22	44	28
	800	**0.75**	0.71	**0.73**	27	48	19
	400	0.68	**0.77**	**0.73**	33	44	18
	200	0.54	0.66	0.60	35	58	3

Table 5: FS on individual datasets, showing the distribution of the three most important type of features in percentages (syntactic walks, lexical walks and BOW-features). Evaluation using Abstract CV.

test	features	p	r	F	syn	lex	bow
AIMed	11.300	0.27	0.67	0.38	12	57	28
	10.000	0.27	**0.69**	**0.39**	12	55	28
	7.500	**0.28**	0.65	**0.39**	13	60	24
	5.000	0.27	0.60	0.37	14	61	21
HPRD50	26.100	0.62	**0.52**	0.57	9	67	21
	20.000	0.69	0.51	**0.59**	9	66	21
	15.000	0.76	0.47	0.58	9	66	22
	10.000	**0.80**	0.25	0.38	7	69	20
IEPA	22.500	**0.87**	**0.27**	**0.41**	10	67	21
	20.000	0.84	0.24	0.38	9	66	21
	15.000	0.84	0.23	0.36	10	66	22
	10.000	0.71	0.16	0.25	11	63	23
LLL	26.700	0.54	0.32	**0.40**	9	67	21
	25.000	0.51	**0.33**	**0.40**	8	66	22
	20.000	0.43	0.21	0.28	9	66	22
	15.000	0.53	0.28	0.37	9	66	22
	10.000	**0.93**	0.15	0.26	7	69	20

Table 6: FS on cross-dataset experiments, showing the distribution of the three most important type of features in percentages (syntactic walks, lexical walks and BOW-features). Evaluation using three datasets as training data and one dataset as test set.

true interaction between each co-occurring pair of proteins. These results exhibit a difference in F-measure of up to 0.36 between AIMed and LLL.

Subsequently, we compared our method using rich feature vectors to other, recently introduced PPI extraction techniques. To allow for a fair comparison, we only consider studies using a similar evaluation setup. The results of this analysis are shown in Table 4. We observe that our method is comparable to state-of-the art performance, and that it achieves particularly good results when using regular CV on the LLL dataset.

4.1.1 Feature selection

Because our extraction method results in high-dimensional, sparse feature vectors, we have investigated the usability of feature selection techniques to improve performance and obtain faster models. The results of these experiments on the individual datasets are shown in Table 5. On HPRD50, recall could be increased with 0.11 resulting in an increase in F-measure of 0.06, while less than 20% of the features were kept. For IEPA and LLL, F-measure remains stable when using respectively 36% and 25% of all available features. These results clearly indicate that FS can reduce the feature set considerably without loss of performance. For the more extensive dataset AIMed, the number of extracted features and training instances are multiplied by a factor 10 in comparison to the other datasets, which induces greater complexity. On AIMed, we can filter out 29% of all features while still obtaining the same performance. If we filter out 64%, keeping only 5000 features of the original set, the F-measure drops with 0.05. However, the time necessary to build the classifier for all ten folds is reduced from 6 hours and 5 minutes to 3 hours and 22 minutes, including the FS step itself. This illustrates the usefulness of feature selection to create more cost-effective models.

Analysing the distribution of feature types before and after FS, we see that in general, syntactic features take up a slightly bigger proportion after filtering, usually accompanied by a reduction of word features (Table 5, last three columns). However, lexical information still takes up the biggest part of the feature set.

4.2 Cross dataset experiments

To assess the performance of our method in a more realistic setup, we have conducted large-scale cross-datasets experiments. For this purpose, we used one dataset for testing, and the other three for training, which will cause less bias to a specific training set. These experiments provide an estimate of the out-of-domain generalization ability, by analysing the artificial boost in performance when only performing a single-domain evaluation. It is the first time such a broad cross-corpus study is conducted.

The results of our experiments are shown in Table 6. We see that testing on HPRD50 achieves the best performance, with 0.62 precision, 0.52 recall and 0.57 F-measure. For this corpus, the

test set	Features	p	r	F
AIMed	all	0.27	0.67	**0.38**
	syntactic	**0.28**	0.58	0.37
	lexical	0.24	**0.72**	0.36
HPRD50	all	0.62	**0.52**	**0.57**
	syntactic	**0.70**	0.48	**0.57**
	lexical	0.60	0.50	0.54
IEPA	all	**0.87**	**0.27**	**0.41**
	syntactic	0.62	0.26	0.37
	lexical	0.82	0.17	0.29
LLL	all	0.54	**0.32**	0.40
	syntactic	**0.64**	0.30	**0.41**
	lexical	0.47	0.28	0.35

Table 7: Cross-dataset experiments using lexical information, syntactic information or both

performance is similar to the single-dataset evaluation. However, we observe a large drop in performance when testing on IEPA and LLL, and to a smaller extent, on AIMed. This shows that studies using single-dataset evaluations, are biased towards the specific properties of the corpus used. It confirms the need for extrinsic evaluations of text mining tools as stated by Caporaso et al. (2008).

The cross-dataset experiments give rise to high-dimensional datasets, with up to 26.700 features. We have applied FS in order to filter out irrelevant features, and obtain faster models with less risk of overfitting. The results for all four test cases can be found in Table 6. In most cases, we are able to reduce the feature set significantly without loss of performance. Testing on HPRD50, we achieve a gain in precision of 0.14 while only sacrificing 0.05 recall, when the feature set is reduced to 57% of its original size. Model construction with the entire feature set took one hour and 36 minutes, while the classifier was built after 57 minutes using the reduced feature set. The FS step itself only took an additional 5 minutes. This clearly shows that FS can lead to faster and more cost-effective models.

Testing on HPRD50, precision can rise to 0.84 when even more features are filtered, though recall starts dropping at this point. Nevertheless this faster model may be preferred by a user who only wants to extract highly reliable data. On LLL we also obtain much higher precision when sacrificing recall. When using AIMed as test set, we are able to maintain good results when more than half of the features are filtered out.

4.2.1 Contribution of lexical and syntactic information

In order to gain deeper insight into the importance of certain features, we performed a statistical analysis of the contribution of different categories of features (Table 6, last three columns). In general, we saw that 85-90 % of the features consist of lexical information (lexical walks and BOW features combined). This distribution is roughly maintained after feature selection. This indicates that both lexical and syntactic information are important when extracting protein-protein interactions. We validated this assumption by running the cross-dataset experiments again, once with only lexical information, and once with only syntactic information. The results are shown in Table 7, demonstrating that the global performance of both lexical and syntactic approaches are similar to each other. However, when using only syntactic information and comparing this approach to the full feature set, a gain of precision of up to 0.10 can be achieved (HPRD50, LLL), while producing a similar F-score. The only exception to this general rule seems to be when IEPA is used as testing set. In this particular case, high precision is achieved by mainly lexical information. However, it is clear that a purely syntactic approach can produce satisfying performance, while using only 10-15 % of the original feature set. These results support the hypothesis stated by Fayruzov et al. (2008) that using only syntactic information leads to classifiers that are able to perform well, while being independent of a specific lexicon. To improve recall however, including lexical information might still be useful.

5 Conclusions and future work

We have developed a technique to extract protein-protein interactions using rich feature vectors and machine learning techniques. For the extraction of relevant features, semantic information from dependency graphs was used, as well as lexical information from the sentence expressing an interaction. We have discussed some important issues for benchmarking extraction techniques, and have indicated practical guidelines for setting up a meaningful evaluation. As an important novelty, we have conducted cross-dataset experiments which offer a more realistic view on the performance of our method. Finally, for the first time in this domain, we have applied feature se-

lection techniques to show these can improve the generalization performance and lead to faster and more cost-effective models. Analysing the feature sets from our experiments before and after feature selection, we have shown the importance of combining both lexical and syntactic information for the extraction of interactions from text.

Beyond the approach of rich feature vectors and feature selection, we would like to use the insight gained from these experiments to develop more specific kernel-based approaches for the extraction of protein-protein interactions from text, building further on relation extraction kernels already developed (Kim et al., 2008).

Acknowledgments

SVL would like to thank the Special Research Fund (BOF) for funding her research. YS would like to thank the Research Foundation Flanders (FWO) for funding his research.

References

B. Boser, I. Guyon and V.N. Vapnik. 1992. A training algorithm for optimal margin classifiers. *Proceedings of COLT 1992*, 144-152

R. Bunescu, R. Ge, R. Kate, E. Marcotte, R. Mooney, A. Ramani and Y. Wong. 2005. Comparative Experiments on Learning Information Extractors for Proteins and their Interactions. *Artificial intelligence in medicine*, 33(2):139-155

J.G. Caporaso, N. Deshpande, J.L. Fink, P.E. Bourne, K.B. Cohen and L. Hunter. 2008. Intrinsic evaluation of text mining tools may not predict performance on realistic tasks. *Proceedings of PSB'08*, 640-51

J. Ding, D. Berleant, D. Nettleton and E. Wurtele. 2002. Mining MEDLINE: abstracts, sentences, or phrases? *Proceedings of PSB'02*, 326-337

G. Erkan, A. Ozgur and D. R. Radev. 2007. Extracting interacting protein pairs and evidence sentences by using dependency parsing and machine learning techniques. *Proceedings of BioCreAtIvE 2*

T. Fayruzov, M. De Cock, C. Cornelis and V. Hoste. 2008. DEEPER: a Full Parsing based Approach to Protein Relation Extraction. *Lecture Notes In Computer Science*, 4973, 36-47

K. Fundel, R. Küffner and R. Zimmer. 2007. RelEx—Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365-371

C. Giuliano, A. Lavelli and L. Romano 2006. Exploiting Shallow Linguistic Information for Relation Extraction from Biomedical Literature. *EACL*

I. Guyon and A. Elisseeff 2003. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3 : 11571182.

L. Hirschman, A. Yeh, C. Blaschke and A. Valencia. 2005. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 6(Suppl 1):S1

R. Hoffmann and A. Valencia. 2004. A Gene Network for Navigating the Literature. *Nature Genetics*, 36:664

J. Jiang and C. Zhai. 2007. A Systematic Exploration of the Feature Space for Relation Extraction. *Proceedings of NAACL-HLT 07*, 113-120

S. Katrenko and P. Adriaans. 2007. Learning Relations from Biomedical Corpora Using Dependency Trees. *Lecture notes in Computer Science*, KDECB, volume 4366

J.-D. Kim, T. Ohta and J. Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 19(Suppl 1):i180-i182

S. Kim, J. Yoon and J. Yang. 2008. Kernel approaches for genic interaction extraction. *Bioinformatics*, 9:10

MC. de Marneffe, B. MacCartney and C. D. Manning. 2006. Generating Typed Dependency Parses from Phrase Structure Parses. *Proceedings of LREC'06*

C. Nedellec. 2006. Learning language in logic - genic interaction extraction challenge. *Proceedings of LLL'05*, 31-37

M. F. Porter. 1980. An algorithm for suffix stripping. *Program*, 14(3), 130-137

S. Pyysalo, F. Ginter, J. Heimonen, J. Björne, J. Boberg, J. Järvinen and T. Salakoski. 2007. BioInfer: A corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, 8(50)

S. Pyysalo, A. Airola, J. Heimonen, J. Björne, F. Ginter and T. Salakoski. 2008. Comparative analysis of five protein-protein interaction corpora. *BMC Bioinformatics*, 9(Suppl 3):S6

D. Rebholz-Schuhmann, H. Kirsch, M. Arregui, S. Gaudan, M. Riethoven and P. Stoehr. 2006. EBIMed - text crunching to gather facts for proteins from Medline. *Bioinformatics*, 23(2):e237-e244.

R. Saetre, K. Sagae and J. Tsujii. 2008. Syntactic features for protein-protein interaction extraction. *Proceedings of LBM'07)*

Y Saeys, I. Inza and P. Larrañaga. 2007. A review of feature selection techniques in bioinformatics. *Bioinformatics*, 23(19):2507-2517.

S. Van Landeghem, Y. Saeys, B. De Baets and Y. Van de Peer. 2008. Benchmarking machine learning techniques for the extraction of protein-protein interactions from text. *Proceedings of Benelearn'08*, 79-80.

H. Wang, M. Huang, S. Ding and X. Zhu. 2008. Exploiting and integrating rich features for biological literature classification. *BMC Bioinformatics*, 9(Suppl 3):S4