Detecting Relations in the Gene Regulation Network

Thomas Provoost
Marie-Francine Moens
Department of Computer Science
KU Leuven
Celestijnenlaan 200A, 3000 Leuven, Belgium
{thomas.provoost, sien.moens}@cs.kuleuven.be

Abstract

The BioNLP Shared Task 2013 is organised to further advance the field of information extraction in biomedical texts. This paper describes our entry in the Gene Regulation Network in Bacteria (GRN) part, for which our system finished in second place (out of five). To tackle this relation extraction task, we employ a basic Support Vector Machine framework. We discuss our findings in constructing local and contextual features, that augment our precision with as much as 7.5%. We touch upon the interaction type hierarchy inherent in the problem, and the importance of the evaluation procedure to encourage exploration of that structure.

1 Introduction

The increasing number of results in the biomedical knowledge field has been responsible for attracting attention and research efforts towards methods of automated information extraction. Of particular interest is the recognition of information from sources that are formulated in natural language, since a great part of our knowledge is still in this format. Naturally, the correct detection of biomedical events and relations in texts is a matter which continues to challenge the scientific community. Thanks to the BioNLP Shared tasks, already in the third instalment, researchers are given data sets and evaluation methods to further advance this field.

We participated in the Gene Regulation Network (GRN) Task (Bossy et al., 2013), which is an extension of the Bacteria Gene Interactions Task from 2011 (Jourde et al., 2011). In this task, efforts are made to automatically extract gene interactions for *sporulation*, a specific cellular function of the bacterium *bacillus subtilis* for which a stable reference regulatory network exists. An example sentence can be seen below. Note that all entities (except for event triggers, i.e. action entities like *transcription* in figure 1) are given as input in both training and test phases. Therefore, this task makes abstraction of the entity recognition issue, putting complete focus on the subproblem of relation detection.

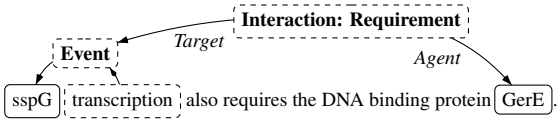


Figure 1: Example sentence: there is an Interaction:Requirement relation defined between entities GerE and sspG, through the action event of transcription. Full-line entities are given in the test phase, while dashed-lined ones are not.

As this is our first participation in this task, we have built a simple, yet adaptable framework. Our contributions lie therefore more in the domain of feature definition and exploration, rather than in designing novel machine learning models.

Predictions could be given in two ways. Either all events and relations could be predicted, from which the regulation network would then be inferenced (cfr. figure 1, detect all dashed-lined entities, and the relations between them). Or, a specification of the regulation network itself is directly predicted (in the example, this amounts to finding $GerE \rightarrow sspG$, and the type (*Requirement*)). We chose to implement the latter method. In section 2 we will lay out the framework we constructed, and the tools we used. In that section, we will also look at some of the design choices for our feature construction. Finally we discuss our results in section 3, and touch upon opportunities to exploit the available interaction hierarchy in this data.

Proceedings of the BioNLP Shared Task 2013 Workshop, pages 135–138,
Sofia, Bulgaria, August 9 2013.

2 Implementation

Basic Framework For this interaction detection task, we implement a Support Vector Machine (SVM) (Vapnik, 1995), with the use of the SVMLight (Joachims, 1999) implementation in the Shogun Machine Learning Toolbox. Per given sentence, we construct our data points to be all pairs of genic entities in that sentence, i.e., all possible interaction agent/target pairs. Note that since the regulation network is a directed graph, the order of the nodes matters; each such pair therefore occurs twice in the data. It is obvious from this construction that this leads to a great imbalance: there are a lot more negatively labelled data points than positive ones. To respond to this, we tried applying differential weighing (as seen in (Shawe-Taylor and Cristianini, 1999) and (Veropoulos et al., 1999)). This amounts to appointing a bigger regularisation parameter C to the positive data points when training the SVM, thus tightening the boundary constraint on the margin for these points. The results of this were unconvincing however, so we decided not to implement it.
For each interaction type (there are 6 of them), we then train a separate binary (local, hence one-versus-all) SVM classifier[1], with a Gaussian Radial Basis Function (RBF) kernel as in (Aĭzerman et al., 1964) and (Schölkopf et al., 1997). We evaluated several types of kernels (linear, polynomial, Gaussian) in a 25-fold cross-validation over the union of training and validation set, and the RBF-kernel consistently gave better results.

Feature Construction and Selection Consider our data points (i.e., the agent/target pairs) $x_{ijk} = (e_{i_j}, e_{i_k})$, $j \neq k$, where e_{i_j} denotes the jth entity of sentence i. For each such point, the basic (real-valued) feature set-up is this:

$$f(x_{ijk}) = f_{ent}(e_{i_j}) \odot f_{ent}(e_{i_k}) \odot f_{extra}(e_{i_j}, e_{i_k}),$$

a concatenation (the $\odot$ operation) of the respective feature vectors f_{ent} defined separately on the provided entities. To that we add f_{extra}, which contains the Stanford parse tree (Klein and Manning, 2003) distance of the two entities, and the location and count (if any) of *Promoter* entities: these are necessary elements for transcription without being part of the gene itself. For any entity, we then construct the feature vector as:

$$f_{ent}(e_{i_j}) = \frac{1}{N_{i_j}} \sum_{w \in e_{i_j}} f_{base}(w) \odot f_{context}(w, i),$$

where N_{i_j} is the number of words in e_{i_j}. This is an average over all words w that make up entity e_{i_j}[2], with the choice of averaging as a normalisation mechanism, to prevent a consistent assignment of relatively higher values to multi-word entities. Inside the sum is the concatenation of the local feature function on the words (f_{base}) with $f_{context}$, which will later be seen as encoding the sentence context.
The base feature function on a word is a vector containing the following dimensions:

- The entity type, as values $\in \{0, 1\}$;
- Vocabulary features: for each word in the dictionary (consisting of all words encountered), a similarity score $\in [0, 1]$ is assigned that measures how much of the beginning of the word is shared[3]. In using a similarity scoring instead of a binary-valued indicator function, we want to respond to the feature sparsity, aggravated by the low amount of data (134 sentences in training + validation set). While this introduces some additional noise in the feature space, this is greatly offset by a better alignment of dimensions that are effectively related in nature. Also note that, due to the nature of the English language, this approach of scoring similarities based on a shared beginning, is more or less equivalent to stemming (albeit with a bias towards more commonly occurring stems). For our cross-validations, utilisation of these similarity scores attributed to an increase in F-score of 7.6% (mainly due to an increase in recall of 7.0%, without compromising precision) when compared to the standard binary vocabulary features.
- Part-of-speech information, using the Penn-Treebank (maximum entropy) tagger, through the NLTK Python library (Bird et al., 2009). These are constructed in the same fashion as the vocabulary features;

[1] There is a lot of scope for leveraging the hierarchy in the interaction types; we touch upon this in the conclusion.

[2] Note that one entity can consist of multiple words.

[3] To not overemphasise small similarities (e.g. one or two initial letters in common), we take this as a convex function of the proportion of common letters.

- Location of the word in its sentence (normalised to be $\in [0, 1]$). Note that next to being of potential importance in determining an entity to be either target or agent, the subspace of the two location dimensions of the respective entities in the data point $x_{ijk} = (e_{i_j}, e_{i_k})$ also encodes the word distance between these.

- Depth in the parse tree (normalised to be $\in [0, 1]$).

Adding contextual features On top of these basic features, we add some more information about the context in which the entities reside. To this effect, we concatenate to the basic word features the *tree context*: a weighted average of all other words in the sentence:

$$f_{context}(w, i) = \frac{1}{Z} \sum_{w_j \in sentence_i} \alpha^{d_i(w,w_j)} f_{base}(w_j)$$

with f_{base} the basic word features described above, and weights given by $\alpha \leq 1$[4] and $d_i(w, w_j)$ the parse tree distance from w to w_j. The normalisation factor we use is

$$Z = \sum_{w_j \in sentence_i} \alpha^{d_i(w,w_j)}$$

i.e., the value we would get if a feature would be consistently equal to 1 for all words. This normalisation makes sure longer sentences are not overweighted. For the inner parse tree nodes we then construct a similar vector (using only part-of-speech and phrasal category information), and append it to the word context vector.
Note that the above definition of $f_{context}$ also allows us to define $d_i(w, w_j)$ to be the word distance in the sentence, leaving out any grammatical (tree) notion. We designate this by the term *sentence context*.

3 Results and Conclusion

Cross-validation performance on training data Because we have no direct access to the final test data, we explore the model performance by considering results from a 25-fold cross-validation on the combined training and validation set. Table 1 shows the numbers of three different implementations[5]: one with respectively no $f_{context}$ concatenated, and the *tree context* (the official submission method) and *sentence context* versions. We see that a model based uniquely on information from the agent and target entities already performs quite well; a reason for this could be the limited amount of entities and/or interactions that come into play in the biological process of *sporulation*, augmenting the possibility that a pair can already be observed in the training data. Adding context information increases the F-score by 2%, mainly due to a substantial increase in precision, as high as 7.5% for the *sentence* context. Recall performs better in the *tree* variant however, pointing to the fact that grammatical structure can play a role in identifying relations.
Note that we only considered the *sentence* alteration after the submission deadline, so the better results seen here could no longer implore us to use this version of the context features.

Context	SER	Prec.	Recall	F1
None	0.827	0.668	0.266	0.380
Tree	0.794	0.709	0.285	0.406
Sentence	0.787	0.743	0.278	0.405

Table 1: Results of the cross-validation for several implementations of context features. ($C = 5, \sigma = 8.75$)

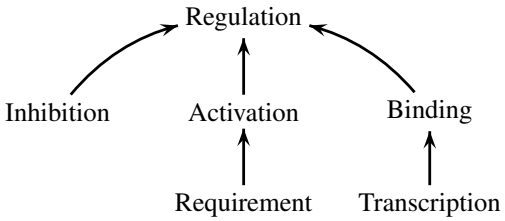
We can identify some key focus points to further improve our performance. Generally, as can be seen in the additional results of table 1, a low recall is the main weakness in our system. These low numbers can in part be explained by the lack of great variation in the features, mainly due to the low amount of data we have. Interesting to note here, is the great diversity of performance of the local classifiers separately: the SVM for *Transcription* attains a recall of 42.0%, in part because this type is the most frequent in our data. However, the worst performers, *Requirement* and *Regulation* (with a recall of 0.0% and 3.7% respectively) are not per se the least frequent; in fact, *Regulation* is the second most occurring. Considerable effort should be put into addressing the general recall issue, and gaining further insight into the reasons behind the displayed variability.

[4] We optimised α to be 0.4, by tuning on a 25-fold cross-validation, only using training and validation set.

[5] For simplicity, we keep all other parameters (C, and the RBF kernel parameter σ) identical across the different entries of the table. While in theory a separate parameter optimisation on each model could affect the comparison, this showed to be of little qualitative influence on the results.

Final results on test data On submission of the output from the test data, our system achieved a Slot Error Rate (SER) of 0.830 (precision: 0.500, recall: 0.227, F1: 0.313), coming in second place after the University of Ljubljana (Zitnik et al., 2013) who scored a SER of 0.727 (precision: 0.682, recall: 0.341, F1: 0.455).

Exploring structure One of the main issues of interest for future research is the inherent hierarchical structure in the interactions under consideration. These are not independent of each other, since there are the following inclusions:



So for example, each interaction of type *Transcription* is also of type *Binding*, and *Regulation*. This structure implicates additional knowledge about the output space, and we can use this to our benefit when constructing our classifier.

In our initial framework, we make use of local classifiers, and hence do not leverage this additional knowledge about type structure. We have already started exploring the design of techniques that can exploit this structure, and preliminary results are promising.

One thing we wish to underline in this process is the need for an evaluation procedure that is as aware of the present structures as the classifier. For instance, a system that predicts a *Binding* interaction to be of type *Regulation*, is more precise than a system that identifies it as an *Inhibition*. Both for internal as external performance comparison, we feel this differentiation could broaden the focus towards a more knowledge-driven approach of evaluating.

Acknowledgements

We would like to thank the Research Foundation Flanders (FWO) for funding this research (grant G.0356.12).

References

Mark A. Aĭzerman, E. M. Braverman, and Lev I. Rozonoer. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837.

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. OReilly Media Inc.

Robert Bossy, Philippe Bessiŕes, and Claire Nédellec. 2013. BioNLP shared task 2013 - an overview of the genic regulation network task. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August. Association for Computational Linguistics.

Thorsten Joachims. 1999. Making large-scale SVM learning practical. In Bernhard Schölkopf, Christopher J. C. Burges, and Alexander J. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*, chapter 11, pages 41–56. MIT Press.

Julien Jourde, Alain-Pierre Manine, Philippe Veber, Karën Fort, Robert Bossy, Erick Alphonse, and Philippe Bessières. 2011. BioNLP shared task 2011 - bacteria gene interactions and renaming. In *Proceedings of BioNLP Shared Task 2011 Workshop*, pages 56–64.

Dan Klein and Christopher D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *Advances in Neural Information Processing Systems*, number 15, pages 3–10. MIT Press.

Bernhard Schölkopf, Kah-Kay Sung, Christopher J. C. Burges, Federico Girosi, Partha Niyogi, Tomaso Poggio, and Vladimir N. Vapnik. 1997. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. *IEEE Transactions on Signal Processing*, 45(11):2758–2765.

John Shawe-Taylor and Nello Cristianini. 1999. Further results on the margin distribution. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, COLT '99, pages 278–285, New York, NY, USA. ACM.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Konstantinos Veropoulos, Colin Campbell, and Nello Cristianini. 1999. Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on AI*, pages 55–60.

Slavko Zitnik, Marinka itnik, Bla Zupan, and Marko Bajec. 2013. Extracting gene regulation networks using linear-chain conditional random fields and rules. In *Proceedings of BioNLP Shared Task 2013 Workshop*, Sofia, Bulgaria, August. Association for Computational Linguistics.