Exposé: Single-Step Extraction of Protein-Protein Interactions with Support Vector Machines

Tim Rocktäschel <trocktae@informatik.hu-berlin.de>

1 Motivation

Most of the information about protein-protein interactions (PPIs) can only be found in unstructured natural language texts (Ono *et al.*, 2001). PubMed, a publicly accessible database for biomedical literature, nowadays consists of over 20 million citations[1] and is still growing fast. Thus, methods that extract PPIs automatically are of great importance. The extraction of PPIs requires the recognition of relevant entities, in this case proteins, and the relations between them.

To measure the progress made by the research community, competitions such as the BioNLP Shared Task 2009[2], 2011[3] and BioCreative III[4] are held. To separate concerns, in competitions evaluating PPI extraction it is common to use gold standard annotations for all protein mentions. Hence, the recognition of proteins has no impact on the performance and solely the quality of relation extraction for given proteins is measured.

Kabiljo *et al.* (2009) showed that the use of gold standard annotations has a high impact on the performance of methods for PPI extraction, since error propagation caused by named entity recognition (NER) is not considered. They evaluated AkanePPI[5], a state-of-the-art system for PPI extraction, on five different corpora, including the AIMed corpus (Bunescu *et al.*, 2005) and the BioInfer corpus (Pyysalo *et al.*, 2007). Kabiljo *et al.* observed a drop in F-score between 2.2 and 22.7 percentage when using BANNER (Leaman and Gonzalez, 2008) instead of gold standard annotations. BANNER is a state-of-the-art system for NER that is based on Conditional Random Fields (Lafferty *et al.*, 2001) and achieves competitive results in protein recognition tasks[6].

[1] http://www.ncbi.nlm.nih.gov/pubmed

[2] http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/SharedTask/

[3] http://sites.google.com/site/bionlpst/

[4] http://www.biocreative.org/

[5] http://www-tsujii.is.s.u-tokyo.ac.jp/~satre/akane/AkanePPI.v0.1.html

[6] http://banner.sourceforge.net/

2 Goal

The aim of this work is to build and evaluate a single-step extraction system for PPIs based on Support Vector Machines (SVMs) (Joachims, 1998). By combining NER and PPI extraction to one step, we intend to address the problem of error propagation. Moreover, the single-step PPI extractor may lead to a more accurate NER of proteins.

3 Related Work

3.1 SVMs for PPI extraction

SVMs are commonly used kernel-based learners for PPI extraction (Tikk *et al.*, 2010). Three out of the top five extractors in the core event extraction part of the BioNLP Shared Task 2009 used SVMs. However, the core event extraction task is focused on specific molecular events, which are not always equivalent to PPIs.

To extract a PPI, the sentence with its protein mentions is mapped to a high-dimensional feature space. The sentence's feature vector can consist of shallow (e.g. part-of-speech (POS) tags) or deep (e.g. shortest path in a dependency tree) linguistic features to describe whether an interaction between the proteins exists or not. Subsequently, the SVM learns the hyperplane with the maximal margin that separates the positive from the negative examples.

jSRE[7](Giuliano *et al.*, 2006) is a tool for relation (e.g. PPI) extraction written in Java. It utilizes the SVM implementation LIBSVM (Chang and Lin, 2001). jSRE's kernel is based solely on shallow linguistic features.

3.2 Markov Logic Networks

Riedel *et al.* (2009) achieved the 4th best F-score in the core event extraction task with a Markov Logic Network (MLN). A MLN is a set of weighted first order logic formulas with which one can define predicates that describe a relation. These predicates can refer to deep linguistic features

[7] http://hlt.fbk.eu/en/technology/jSRE

(e.g. linking two proteins in a dependency tree) or model constraints that every relation should satisfy.

Riedel *et al.* use a MLN to jointly predict arguments and triggers of molecular events, whereas a common approach is to use several classifiers in a pipeline architecture for the event extraction. However, they point out that interpreters for Markov Logic are not yet able to cope with entities of which the number is unknown. Thus, combining NER and PPI extraction is not yet feasible with a MLN.

4 Agenda

4.1 Combined Feature Vector

We will combine NER and PPI extraction by building a feature vector consisting of BANNER's features for NER and jSRE's features for PPI extraction. Consequently, this feature vector has to be build for each of a sentence's word pair. We will then use jSRE to transform this feature representation back into an input for LIBSVM.

4.2 BioInfer Corpus

We will train the classifier on the BioInfer corpus. Of the 6349 entities in the corpus, 3031 entities either contain a nested entity or are nested within another entity. Moreover, 266 of the 2662 relations have another relationship as their argument (Pyysalo *et al.*, 2007). However, we will focus only on single-token proteins and non-nested interactions for which we will have to write our own corpus reader.

We will evaluate the classifier using document-level 10-fold cross-validation. Subsequently, we will compare our results with those obtained by Kabiljo *et al.*. However, we are aware of the fact that we will overestimate the performance of our classifier as we only consider single-token proteins and non-nested interactions.

4.3 Complexity Reduction

The classifier will need to examine each of a sentence's word pair in order to extract a PPI. Thus, if k is the number of words in the sentence, it will perform $\binom{k}{2}$ classifications. To reduce the number of classifications, we will try to consider only those entities that occur within a noun phrase. We will use OpenNLP[8] to detect POS tags and noun phrases.

[8] `http://incubator.apache.org/opennlp/`

References

Bunescu, R., Ge, R., Kate, R., Marcotte, E., Mooney, R., Ramani, A., and Wong, Y. (2005). Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, **33**(2), 139–155.

Chang, C. and Lin, C. (2001). LIBSVM: a library for support vector machines. *URL: http://www.csie.ntu.edu.tw/cjlin/libsvm*.

Giuliano, C., Lavelli, A., and Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. In *Proceedings of the Eleventh Conference of the European Chapter of the Association for Computational Linguistics (EACL-2006)*, pages 5–7.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98*, pages 137–142.

Kabiljo, R., Clegg, A., and Shepherd, A. (2009). A realistic assessment of methods for extracting gene/protein interactions from free text. *BMC bioinformatics*, **10**(1), 233.

Lafferty, J., McCallum, A., and Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Leaman, R. and Gonzalez, G. (2008). BANNER: An executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, volume 13, pages 652–663.

Ono, T., Hishigaki, H., Tanigami, A., and Takagi, T. (2001). Automated extraction of information on protein–protein interactions from the biological literature. *Bioinformatics*, **17**(2), 155.

Pyysalo, S., Ginter, F., Heimonen, J., Björne, J., Boberg, J., Järvinen, J., and Salakoski, T. (2007). BioInfer: A corpus for information extraction in the biomedical domain. *BMC bioinformatics*, **8**(1), 50.

Riedel, S., Chun, H., Takagi, T., and Tsujii, J. (2009). A markov logic approach to bio-molecular event extraction. In *Proceedings of the Workshop on BioNLP: Shared Task*, pages 41–49. Association for Computational Linguistics.

Tikk, D., Thomas, P., Palaga, P., Hakenberg, J., and Leser, U. (2010). A Comprehensive Benchmark of Kernel Methods to Extract Protein–Protein Interactions from Literature. *PLoS Computational Biology*, **6**(7), 2–8.