Exposé for a Master's Thesis

Stefan Selent

January 21, 2017

Working Title:

TF Relation Mining: An Active Learning Approach

Introduction

The amount of scientific literature is ever increasing. Especially during the past few years. Subsequently, keeping an overview on a certain research topic is an enormous task that might result in missed insights and repeated work if done insufficiently. Additionally, the generated knowledge is often only available in natural language text documents putting up obstacles for automatic analysis by machines. Both observations together describe a general problem that can be found in many areas: the absence of an efficient way for knowledge consolidation.

One of those areas is the extraction for information from biomedical publications. More specifically, the gathering of regulatory relationships between *transcription factors* (TF) to construct the human gene regulatory network. A TF regulates genes and/or other TF in complex ways, controlling for example cellular processes like wound healing. A lot of work has been done to unravel the relationships with the help of time-consuming, low-throughput laborious experiments. The results were published across many different unstructured articles. Even tough there are attempts on structuring these findings in databases [5, 23, 8], these collections remain small in comparison to what might be available in the literature altogether. To combat this problem, Thomas et al.[19] proposed an automated workflow to extract candidate sentences from abstracts of PubMed[1] publications. These candidates were automatically ranked by the likelihood of them containing information about TF interactions. The most promising sentences were then evaluated by domain experts, yielding many new regulatory relationships.

Problem Statement

In this thesis I want to build on top of the aforementioned approach by improving the machine learned classification model that was used to predict if a sentence contained a TF-TF relationship. The improvement should be achieved by using manually labeled training data provided by Thomas et al. as well as by utilising an *Active Learning* (AL) approach. Since AL assumes that unlabeled data is available en mass but the labeling process is

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

expensive it fits perfectly to the given scenario [16]. Based on these improvements the following two hypotheses shall be evaluated:

1. Adding data to the training corpus will improve the classification accuracy in comparison to only using the training corpora used in the paper.
2. Applying AL methods to the training of a classifier will yield a lower number of training instances needed to achieve the same level of accuracy compared to the passive learner method in the scenario of TF relation discovery.

Both hypotheses introduce an organizational challenge. The original classifier is not available anymore. Consequently, a new classifier has to be build and evaluated against the results of Thomas et al.

The Data

For building a classifier, data to learn on is needed. It consists of sentence-label pairs in which the label describes whether the corresponding sentence contains a relationship between TFs. Two pre-compiled corpora can be used for this task: The GeneReg corpus [1] and the corpus of the BioNLP'09 shared task [7]. Both corpora are available at the WBI corpora repository[2] and will be preprocessed as described by Thomas et al. yielding around 1500 positive and 12000 negative example sentences. As a third corpus the handcrafted evaluation labels of Thomas et al. will be added, extending the set of training data to around 3400 positive and 12600 negative ones. This set will be used to train and evaluate the model. It is important to note that the imbalance of this set has to be handled [22].

To collect unlabeled data, abstracts from PubMed can be preprocessed by splitting on sentence boundaries. Next, sentences that do not contain at least two human TFs (checked via GNAT [6]) can be pruned yielding at least 76000 unlabeled sentences.

Methodology

Building a Model

The basic model will be build following the limited description provided by Thomas et al. Subsequently, a *support vector machine* (SVM) with the *shallow linguistic* or *all-paths graph* kernel shall be used. These kernels performed best in an evaluation by Tikk et al. [20]. Irrespective of the model class used in the original paper, SVMs are well suited for text classification tasks. Many tasks in the context of bioinformatics employ SVMs as their classifier of choice [3, 4] because they can handle a high dimensional feature space while maintaining a low computational cost [15]. This is especially important if an AL approach is later added.

Applying Active Learning

AL with SVMs is a common topic in research. Schohn and Cohen [14] discuss AL on top of an SVM layer on an abstract level. Silva and Ribeiro [17] use it for text classification.

[2] http://http://corpora.informatik.hu-berlin.de/

Song et al. [18] utilized it for protein-protein interaction extraction which is similar to the task at hand. Miotto et al. [12] even proposed a system architecture for combining AL into a manual curation process in the context of bioinformatics.

The concept of AL gives the learner the ability to choose unlabeled instances and request labels. The labels are chosen by an oracle. In the case of TF-TF relation extraction the oracle is a human being. However, in this thesis the oracle will be emulated by a computer for practical reasons. Many different selection strategies can be utilized to pick an instance [16] that will be presented to the oracle, the most basic one being a random selection. However, the goal is to reduce the number of instances that is used for training. So optimization strategies are needed that increase the information gain. For example, statistical classifiers output a confidence score. If this score is high the classifier is sure that the label is correct; thus, not much information can be gained. In contrast, the instance with the lowest confidence is a good target. This strategy is called *Uncertainty Sampling*. For SVMs the confidence can be approximated by the distance of the instance to the model's hyperplane. Tong and Koller [21] discuss this topic with a focus on text classification.

Other methods use a set of classifiers where each member judges all instances. The instance with the lowest agreement will be chosen [11, 10]. However, this strategy requires a valid selection of distinct classifiers. Many more strategies exist that mostly offer a trade-off between computational cost and effective selection.

Implementation

For implementing an SVM within an AL framework several toolkits can be used. The most prominent beeing libsvm [2], Vowpal Wabbit [9] and scikit-learn [13]. They differ in performance, usability and features offered. For instance, while Vowpal Wabbit includes a basic AL environment, libsvm and scikit-learn do not. However, Yang et al. proposed an AL layer on top of libsvm and scikit-learn [24]. As a result, a short evaluation of these tools is necessary.

Evaluation

The evaluation will be partitioned in three areas. First, a comparison between the newly created classifier and the results of Thomas et al. is needed to measure the difference in performance introduced by reimplementing the model. Secondly, the performance of the classifier trained on the base data versus on the enhanced data needs to be compared. Here, one of the evaluation metrics described by Sebastini [15] can be used. To avoid the aforementioned need for further manual evaluation, a portion of the corpus data could be held back for evaluation. Lastly, to measure the impact of AL the number of instances needed to achieve the same accuracy as the passive approach will be examined.

References

[1] Ekaterina Buyko, Elena Beisswanger, and Udo Hahn. “The GeneReg Corpus for Gene Expression Regulation Events-An Overview of the Corpus and Its In-Domain and Out-of-Domain Interoperability.” In: *LREC*. 2010. (Visited on 01/05/2017).

[2] Chih-Chung Chang and Chih-Jen Lin. “LIBSVM: A Library for Support Vector Machines”. In: *ACM Transactions on Intelligent Systems and Technology (TIST)* 2.3 (2011), p. 27. (Visited on 01/11/2017).

[3] Aaron M. Cohen and William R. Hersh. “A Survey of Current Work in Biomedical Text Mining”. In: *Briefings in bioinformatics* 6.1 (2005), pp. 57–71. (Visited on 01/11/2017).

[4] Ian Donaldson et al. “PreBIND and Textomy–mining the Biomedical Literature for Protein-Protein Interactions Using a Support Vector Machine”. In: *BMC bioinformatics* 4.1 (2003), p. 1. (Visited on 01/11/2017).

[5] Obi L. Griffith et al. “ORegAnno: An Open-Access Community-Driven Resource for Regulatory Annotation”. In: *Nucleic acids research* 36.suppl 1 (2008), pp. D107–D113. (Visited on 01/11/2017).

[6] Jörg Hakenberg et al. “The GNAT Library for Local and Remote Gene Mention Normalization”. In: *Bioinformatics* 27.19 (2011), pp. 2769–2771. (Visited on 01/05/2017).

[7] Jin-Dong Kim et al. “Overview of BioNLP’09 Shared Task on Event Extraction”. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task.* Association for Computational Linguistics, 2009, pp. 1–9. (Visited on 01/05/2017).

[8] Nikolay A. Kolchanov et al. “Transcription Regulatory Regions Database (TRRD): Its Status in 2002”. In: *Nucleic acids research* 30.1 (2002), pp. 312–317. (Visited on 01/11/2017).

[9] John Langford, Lihong Li, and Alex Strehl. *Vowpal Wabbit Online Learning Project.* Technical report, http://hunch. net, 2007.

[10] Ray Liere and Prasad Tadepalli. “Active Learning with Committees for Text Categorization”. In: *AAAI/IAAI.* 1997, pp. 591–596. (Visited on 01/11/2017).

[11] Andrew Kachites McCallumzy and Kamal Nigamy. “Employing EM and Pool-Based Active Learning for Text Classification”. In: *Proc. International Conference on Machine Learning (ICML).* Citeseer, 1998, pp. 359–367. (Visited on 01/11/2017).

[12] Olivo Miotto, Tin Wee Tan, and Vladimir Brusic. “Supporting the Curation of Biological Databases with Reusable Text Mining”. In: *Genome informatics* 16.2 (2005), pp. 32–44. (Visited on 01/06/2017).

[13] F. Pedregosa et al. “Scikit-Learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[14] Greg Schohn and David Cohn. “Less Is More: Active Learning with Support Vector Machines”. In: *ICML.* Citeseer, 2000, pp. 839–846. (Visited on 01/11/2017).

[15] Fabrizio Sebastiani. “Machine Learning in Automated Text Categorization”. In: *ACM computing surveys (CSUR)* 34.1 (2002), pp. 1–47. (Visited on 01/07/2017).

[16] Burr Settles. “Active Learning Literature Survey”. In: *University of Wisconsin, Madison* 52.55-66 (2010), p. 11.

[17] Catarina Silva and Bernardete Ribeiro. “On Text-Based Mining with Active Learning and Background Knowledge Using SVM”. In: *Soft Computing* 11.6 (2007), pp. 519–530. (Visited on 01/05/2017).

[18] Min Song, Hwanjo Yu, and Wook-Shin Han. “Combining Active Learning and Semi-Supervised Learning Techniques to Extract Protein Interaction Sentences”. In: *BMC bioinformatics* 12.Suppl 12 (2011), S4. (Visited on 01/06/2017).

[19] Philippe Thomas et al. “Computer-Assisted Curation of a Human Regulatory Core Network from the Biological Literature”. In: *Bioinformatics* (2014), btu795. (Visited on 01/05/2017).

[20] Domonkos Tikk et al. “A Detailed Error Analysis of 13 Kernel Methods for Protein–protein Interaction Extraction”. In: *BMC bioinformatics* 14.1 (2013), p. 1. (Visited on 01/07/2017).

[21] Simon Tong and Daphne Koller. “Support Vector Machine Active Learning with Applications to Text Classification”. In: *Journal of machine learning research* 2.Nov (2001), pp. 45–66. (Visited on 01/05/2017).

[22] Konstantinos Veropoulos et al. “Controlling the Sensitivity of Support Vector Machines”. In: *Proceedings of the International Joint Conference on AI.* 1999, pp. 55–60. (Visited on 01/05/2017).

[23] Edgar Wingender. “The TRANSFAC Project as an Example of Framework Technology That Supports the Analysis of Genomic Regulation”. In: *Briefings in bioinformatics* 9.4 (2008), pp. 326–332. (Visited on 01/11/2017).

[24] Yao-Yuan Yang et al. *Libact: Pool-Based Active Learning in Python.* https://github.com/ntucllab/libact. 2015. (Visited on 01/08/2017).