Exploiting Tree Kernels for High Performance Chemical Induced Disease Relation Extraction

Nagesh C. Panyam, Karin Verspoor, Trevor Cohn and Kotagiri Ramamohanarao
Department of Computing and Information Systems,
The University of Melbourne, Australia
npanyam@student.unimelb.edu.au
{karin.verspoor, t.cohn, kotagiri}@unimelb.edu.au

Abstract

Machine learning approaches based on supervised classification have emerged as effective methods for Biomedical relation extraction such as the Chemical-Induced-Disease (CID) task. These approaches owe their success to a rich set of features crafted from the lexical and syntactic regularities in the text. Kernel methods are an effective alternative to manual feature engineering and have been successfully used in similar tasks such as text classification.

In this paper, we study the effectiveness of tree kernels for Chemical-Disease relation extraction. Our experiments demonstrate that subset tree kernels increase the F-score to 61.7% as compared to 57.9% achieved with simple feature engineering. We also describe the strengths and shortcomings of tree kernel approaches for the CID relation extraction task.

1 Introduction

Scientific publications in the fields of biomedical and life sciences are vast and growing fast (Haas et al., 2014). Prior research has shown that Chemicals and Diseases and their relationships are among the most searched topics by PubMed users (Dogan et al., 2009), due to their importance in applications such as toxicology, drug discovery and safety surveillance. Efforts to manually curate and extract these important concepts such as Chemicals and Diseases and their relationships have led to the creation of structured databases such as the Comparative Toxicogenomics Database (CTD) (Davis et al., 2012). However, manual curation is unlikely to scale (Baumgartner et al., 2007) and has stimulated research interest in automated relation extraction.

The recent shared task for Chemical-Induced-Disease relation extraction (CID) organized by BioCreative-V (Wei et al., 2015), has made available a large body of annotated PubMed abstracts for the valuable Chemical-Disease relations. The shared task revealed that CID relation extraction is a difficult task with best reported systems achieving an F-score of about 57%. Study of the participating teams' approaches reveals that most approaches (14 out of 18) were based on Support Vector Machines (SVMs) (Burges, 1998), modeling relation extraction as a supervised classification problem. Most of these systems obtain their performance through a rich feature set that is manually crafted by studying the syntactic and lexical regularities in the text. Substantial performance boost is also drawn from custom heuristics such as postprocessing rules (Zhou et al., 2016). Designing such an effective relation extraction system involves extensive feature engineering and domain expertise.

Kernel methods in NLP (Collins and Duffy, 2001) have been designed precisely to address this problem of manual feature engineering. These methods enable an efficient and comprehensive exploration of a very high dimensional feature space and to automatically adapt to the dominant patterns expressed in the training set.

In our work, we show that kernel methods can be used for boosting relation extraction performance without having to manually engineer additional features. We demonstrate through experiments that combining tree kernels over constituent parses with simple lexical and syntactic features can substantially enhance the performance of the CID task. We also discuss the strengths and weaknesses of these methods which can assist in the design of better methods in the future.

2 Related Work

Our system is developed in the context of the CID subtask described in BioCreative-V (Wei et al., 2015). Many teams, including the top scoring team (Wei et al., 2015), model the CID task as a supervised binary classification problem. In addition to the annotated PubMed abstracts, alternate sources of information such as the Chemical Toxicology Database (CTD) (Davis et al., 2012) were used. Similar biomedical relation extraction tasks that have been studied are drug-drug interaction (Bjorne et al., 2011) and protein-protein interaction (Lan et al., 2009). A subsequence kernel was presented by (Bunescu and Mooney, 2005) for protein-protein interaction extraction. Richer kernels that use constituent parses or dependency structures are studied in (Chowdhury et al., 2011; Airola et al., 2008) for the protein-protein interaction extraction. Recent approaches have focused on broadening the scope of word matching from a simple lexical match to more complex semantic matching (Saleh et al., 2014). The suitability of these methods for the CID task remains to be explored.

3 Approach

Our goal is to minimize task specific and domain specific feature engineering. We therefore explore the power of domain independent techniques such as kernel methods for effective relation extraction. Kernel methods automatically explore a large feature space and can reduce the need for rich hand crafted features. In our system, we do not employ any preprocessing or custom filtering techniques. We use simple text based features and a knowledge base (CTD) look up as our primary feature set. Further knowledge extraction from text is accomplished through tree kernels.

We cast the CID relation extraction as a binary classification problem. The input to the classifier is a pair of chemical and disease mentions. From the set of predicted relation mentions, we extract their normalized entity ids (MeSH Ids) and add it to the final list of Chemical-Disease relations. We built and tested two types of classifiers, namely linear classifier and tree kernel classifier. The linear classifier uses a flat list of simple features. The tree kernel classifier uses kernel methods over constituent parse trees of input sentences. The detailed steps are described below:

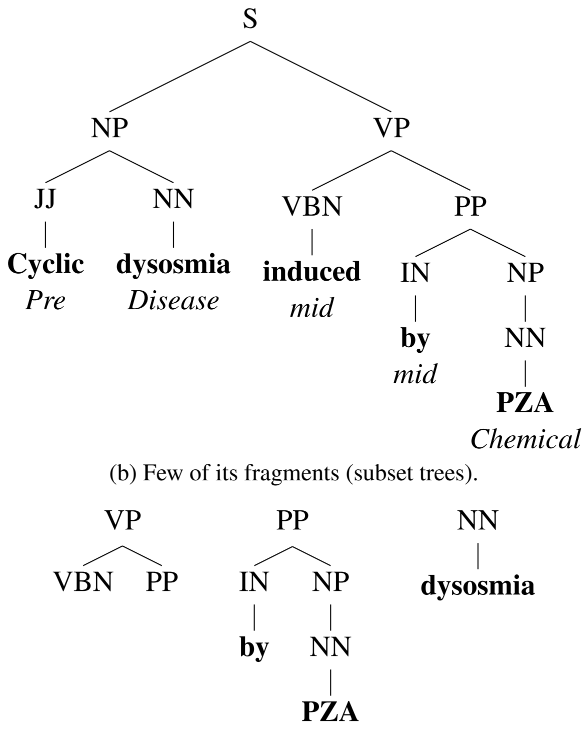
3.1 Linear classifier

1. Every chemical mention (C) that appears in the article is paired with every other disease mention (D) to generate an entity pair (C-D) for classification. An entity pair in which both the entity mentions are within a sentence are referred to as *intrasentence pairs*, and those that cross a sentence boundary are referred to as *intersentence* pairs. The *full test* data is the union of intrasentence and intersentence entity pairs.
2. Intersentence and intrasentence pairs are grouped separately for training and testing with two separate classifiers. No further filtering or post-processing of (C-D) pairs is performed.
3. At training time, we label a (C-D) pair as positive if there exists a valid CID relation between these entities, using the relation annotations. At test time, the label is inferred from the classifier output.
4. Features for intrasentence pairs include verbs, bag of words, POS tags, dependency parse and the token distance between entity mentions in the sentence.
5. Features for intersentence pairs include the POS tags and bag of words of the two sentences containing entity mentions, distance (number of sentences) between them, statistical features of the entity mentions in the document (frequency of mentions), entity frequencies and zonal information (document zone containing the mentions).
6. We use the Chemical Toxicology Database (Davis et al., 2012) to generate a binary feature $I_{ctd}(C, D)$ that evaluates to 1 if the (C-D) pair is known to be related in the CTD database and 0 otherwise.

We used a Support Vector Machine (SVM) with linear kernel from Scikit (Pedregosa et al., 2011) to classify candidate entity pairs. The predicted (C-D) pairs from the sentence level and document level classifiers are combined to form the final list of document level CID relations. We refer to this system as “Linear Classifier”.

3.2 Tree Kernel Classifier

Kernel methods have gained wide spread acceptance, because they allow direct computation of similarity (dot product) between two examples in an implicitly mapped high dimensional


(a) Full constituent parse tree.

(b) Few of its fragments (subset trees).

Figure 1: Illustration for the sentence "Cyclic dysosmia induced by PZA (Pyrazinamide)": a) Full constituent parse tree. b) Few of its fragments (implicitly) considered by the subset tree kernel. Entity focus is incorporated by prefixing special tags such as "pre", "mid", "post", "Chemical" and "Disease".

.

space (Collins and Duffy, 2001; Zelenko et al., 2002). A tree kernel implicitly maps a given tree into a very high dimensional feature space of tree fragments, as illustrated with an example in Figure 1. The kernel score between two trees is the count of common tree fragments between them. We used tree kernels over constituent parse trees of sentences, to efficiently compute the syntactic similarity between two sentences. Different variants of the tree kernels are proposed based on what constitutes a tree fragment, such as subtrees or subsets of nodes. Efficient algorithms with linear time complexity in the average case are presented in (Moschitti, 2006b). The formal definition of the tree kernel is discussed below.

Given two trees T_1 and T_2 and the set of all possible tree fragments $F = \{f_1, f_2, \ldots\}$, an indicator function $I_i(n)$ is defined which evaluates to 1 if the fragment f_i is rooted at node n and 0 otherwise. The unnormalized kernel score is given by

$$k'(T_1, T_2) = \sum_{n_1 \in N_{T_1}} \sum_{n_2 \in N_{T_2}} \Delta(n1, n2)$$

where N_{T_1} and N_{T_2} are the sets of nodes of T_1 and T_2 respectively and $\Delta(n1, n2) = \sum_{i=1}^{|F|} I_i(n_1) I_i(n_2)$. The normalized kernel score is given by

$$K(T_1, T_2) = \frac{k'(T_1, T_2)}{\sqrt{k'(T_1, T_1) \cdot k'(T_2, T_2)}}$$

We experimented with subtree and subset tree kernels over constituent parse trees and found subset tree kernels to be superior for our task. In the rest of the paper, we mean subset tree kernel when we refer to tree kernels. We used Stanford CoreNLP (Manning et al., 2014) to generate the constituent parse trees. For tree kernels we use the SVM-LIGHT-TK toolkit[1] (Moschitti, 2006a) that offers kernel implementation within SVM[2] (Joachims, 1999). For each intrasentence (C-D) pair we get a single parse tree and for each intersentence pair we get a forest of two constituent parse trees, from each sentence containing one of the two entity mentions.

The contribution from flat features as defined in the section 3.1 can be combined with tree kernels by linearly combining the dot products of the flat feature vectors and the tree kernel. That is, the kernel for the new classifier (linear + tree kernel) is computed as the sum of the linear kernel over flat features and the tree kernel over the constituent parse trees. We report results for these classifiers, namely Linear, Tree Kernel and Linear + Tree Kernel classifier in Section 4.

3.2.1 Tree kernels with entity focus

Tree kernels attempt to classify a sentence in its entirety and in its default form are unaware of the entity mentions in the sentence. This approach is suitable if our goal is to simply detect if a sentence expresses a relation or not. However, to render greater focus on the entity mentions, we can preprocess the sentence to highlight the location of a word with reference to entity mentions. We prefixed all words in the sentence with "pre", "mid", and "post" tags, based on whether they are located prior to, in between, or post entity mentions, before generating the constituent parse trees.

[1] http://disi.unitn.it/moschitti/Tree-Kernel.htm

[2] http://svmlight.joachims.org/

Test Data	Classifier	P	R	F1
Intrasentence	Lin - CTD	54.1	71.5	61.6
Intrasentence	Lin	58.2	75.6	65.8
Intrasentence	TK	55.7	53.6	54.6
Intrasentence	Lin + TK	63.3	75.4	68.8
Intersentence	Lin - CTD	26.9	35.1	30.4
Intersentence	Lin	33.7	39.8	36.5
Intersentence	TK	53.8	2.3	4.5
Intersentence	Lin + TK	65.9	20.0	30.8
Full test	Lin - CTD	46.5	61.3	52.9
Full test	Lin	57.8	65.6	57.9
Full test	TK	55.7	39.2	46.0
Full test	**Lin + TK**	**63.6**	59.8	**61.7**
Full test	(Zhou et al., 2016)	55.6	**68.4**	61.3

Table 1: Results on CID test data for Linear classifier (Lin), Tree Kernel (TK) and their combination (Lin+TK). The performance of the linear classifier without CTD feature (Lin - CTD) is also shown.

4 Evaluation

4.1 Dataset and Evaluation metrics

We work with the dataset provided by BioCreative-V (Wei et al., 2015). It comprises 3 subsets, referred to as training, development and test set. Each subset consists of 500 PubMed articles (Title and Abstract only), that are fully annotated with Chemical and Disease mentions and the CID relations. Our goal is to extract Chemical-Disease relations at the document (PubMed abstract) level and the metrics are standard Precision (P), Recall (R) and F1 measure ($\frac{2PR}{P+R}$).

4.2 Results

We measure the effectiveness of our relation extraction system over the provided test data set, as set out in the CID task. We use the standard entity annotations provided with the data set. Given the limited annotated data, we decided to use both the training and development data set for classifier training with default settings and no custom parameter tuning. Results for intersentence, intrasentence and the full set of (C-D) pairs are presented for linear classifier, tree kernel classifier and their combination, in Table 1. We also present the results for the Linear classifier without the CTD feature. Finally, the table also contains the results reported in a prior work by (Zhou et al., 2016) for the CID task. A comparative study with this prior work is presented in Section 5.

To summarize, our final system (linear + tree kernel) achieves an F-score of 61.7% over the CID test data. Note that the combination of linear and tree kernels outperforms the linear and tree kernel classifiers individually. The Table also reveals the substantial contribution of the CTD look up feature towards the linear classifier's performance.

5 Discussion

Comparison with prior art. Previously published results in the CID BioCreative-V task used custom entity recognition tools. Therefore, their CID performance is not directly comparable without replicating their entity annotation process. A more accurate comparison can be made with (Zhou et al., 2016) who follow a similar evaluation process. Their system uses gold standard entity annotations and is trained on the CID training and development datasets and evaluated on the CID test data. They report an F-score of 61.3%. Significantly, their system relies on task specific post-processing rules, without which their F1 score drops to 56.0%. Our system performs better (61.7%), reflecting a substantive advantage in precision, without using heuristics or task specific rules.

Effectiveness of Tree Kernels. We note that tree kernels can significantly improve the performance of CID relation extraction as illustrated in the results. Also, this additional performance is obtained using PubMed abstracts and not external information sources. These results suggest that a greater amount of information exists in annotated text that is easier to extract with tree kernels as compared to manual feature mining for richer patterns. Further, tree kernels have an effect of increasing the precision of the classifiers, specially for intersentence cases. This is likely due to the fact that tree kernels enable stringent comparison of sentence structures (constituent parse trees) as compared to the lenient approach of bag of words matching with linear kernels.

Further enhancements. Incorporating entity focus to tree kernels (section 3.2.1) produced a slight improvement (to 61.7% from 61.0%). This approach is likely to be beneficial for sentences that express multiple relations ($>$ 1) between multiple entity pairs. In the CID dataset, we found that sentences expressing multiple relations constitute around 14%, 15% and 14% of training, development and test datasets respectively. Alternate approaches that discriminate parts of the sentence based on the relation expressed are likely to further improve the performance.

In the context of intersentence (C-D) pairs, we are currently using only the two sentences s_i, s_j that contain the entity mentions. However, the actual relationship might be collectively expressed by any subset of the sentences in the document. We attempted to model the whole of the document as a forest of parse trees of all its sentences, but did not observe any improvement in performance. For the CID task where more than 30% of the (C-D) pairs cross sentence boundaries, effective intersentence relation extraction remains a challenge.

6 Summary and Conclusion

In this work, we show that tree kernels were very effective in CID relation extraction and boosted F1 score to 61.7% as compared to 57.9% achieved with a linear classifier using simple handcrafted features alone. In future work, we seek to improve intersentence relation extraction from documents.

References

Antti Airola, Sampo Pyysalo, Jari Björne, Tapio Pahikkala, Filip Ginter, and Tapio Salakoski. 2008. A graph kernel for protein-protein interaction extraction. In *Proceedings of the workshop on current trends in biomedical natural language processing*, pages 1–9. Association for Computational Linguistics.

William A Baumgartner, K Bretonnel Cohen, Lynne M Fox, George Acquaah-Mensah, and Lawrence Hunter. 2007. Manual curation is not sufficient for annotation of genomic databases. *Bioinformatics*, 23(13):i41–i48.

Jari Bjorne, Antti Airola, Tapio Pahikkala, and Tapio Salakoski. 2011. Drug-drug interaction extraction from biomedical texts with SVM and RLS classifiers. *CEUR Workshop Proceedings*, 761:35–42.

Razvan C. Bunescu and Raymond J. Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS*.

Christopher JC Burges. 1998. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167.

Faisal Mahbub Chowdhury, Alberto Lavelli, and Alessandro Moschitti. 2011. A Study on Dependency Tree Kernels for Automatic Extraction of Protein-Protein Interaction. *BioNLP 2011*, (2011):124–133.

Michael Collins and Nigel Duffy. 2001. Convolution kernels for natural language. In *Advances in neural information processing systems*, pages 625–632.

Allan Peter Davis, Cynthia Grondin Murphy, Robin Johnson, Jean M Lay, Kelley Lennon-Hopkins, Cynthia Saraceni-Richards, Daniela Sciaky, Benjamin L King, Michael C Rosenstein, Thomas C Wiegers, et al. 2012. The comparative toxicogenomics database: update 2013. *Nucleic acids research*, page gks994.

Rezarta Islamaj Dogan, G Craig Murray, Aurélie Névéol, and Zhiyong Lu. 2009. Understanding pubmed® user search behavior through log analysis. *Database*, 2009:bap018.

Laura Haas, Melissa Cefkin, Cheryl Kieliszewski, Wil Plouffe, and Mary Roth. 2014. The ibm research accelerated discovery lab. *SIGMOD Rec.*, 43(2):41–48, December.

Thorsten Joachims. 1999. Advances in kernel methods. chapter Making Large-scale Support Vector Machine Learning Practical, pages 169–184. MIT Press, Cambridge, MA, USA.

Man Lan, Chew Lim Tan, and Jian Su. 2009. Feature generation and representations for protein-protein interaction classification. *Journal of Biomedical Informatics*, 42:866–872.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60.

Alessandro Moschitti. 2006a. Efficient convolution kernels for dependency and constituent syntactic trees. In *Machine Learning: ECML 2006*, pages 318–329. Springer.

Alessandro Moschitti. 2006b. Making tree kernels practical for natural language learning. In *EACL*.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830.

I Saleh, Alessandro Moschitti, Preslav Nakov, L Màrquez, and S Joty. 2014. Semantic Kernels for Semantic Parsing. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 436–442.

Chih-Hsuan Wei, Yifan Peng, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Jiao Li, Thomas C Wiegers, and Zhiyong Lu. 2015. Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the fifth BioCreative challenge evaluation workshop, Sevilla, Spain*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2002. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Huiwei Zhou, Huijie Deng, Long Chen, Yunlong Yang, Chen Jia, and Degen Huang. 2016. Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database*, 2016:baw048.