Graph-based Semi-supervised Gene Mention Tagging

Golnar Sheikhshab^{1,2}, **Elizabeth Starks**^{2}, **Aly Karsan**^{2}, **Anoop Sarkar**^{1}, **Inanc Birol**^{1,2}
gsheikhs@sfu.ca,{lstarks, akarsan}@bcgsc.ca,anoop@sfu.ca,ibirol@bcgsc.ca

[1] School of Computing Science, Simon Fraser University, Burnaby, BC, Canada
[2] Canada's Michael Smith Genome Sciences Centre, British Columbia Cancer Agency, Vancouver, BC, Canada

Abstract

The rapidly growing biomedical literature has been a challenging target for natural language processing algorithms. One of the tasks these algorithms focus on is called named entity recognition (NER), often employed to tag gene mentions. Here we describe a new approach for this task, an approach that uses graph-based semi-supervised learning to train a Conditional Random Field (CRF) model. Benchmarking it on the BioCreative II Gene Mention tagging task, we achieved statistically significant improvements in F-measure over BANNER, a widely used biomedical NER system. We note that our tool is transductive and modular in nature, and can be integrated with other CRF-based supervised NER tools.

1 Introduction

Detecting biomedical named entities such as genes and proteins is one of the first steps in many natural language processing systems that analyze biomedical text. Finding relations between entities, and expanding knowledge bases are examples of research that highly depend on the accuracy of gene and protein mention tagging.

Named entity recognition is typically modelled as a sequence tagging problem (Sha and Pereira, 2003). One of the most commonly used models for sequence tagging is a Conditional Random Field (CRF) (Lafferty et al., 2001; Sha and Pereira, 2003).

Many popular and best performing biomedical named entity recognition systems, such as BANNER (Leaman et al., 2008), Gimli (Campos et al., 2013) and BANNER-CHEMDNER (Munkhdalai et al., 2015) use CRF as their core machine learning model built on the MALLET toolkit (McCallum, 2002).

Inspired by the success of graph-based semi-supervised learning methods in other NLP tasks (Subramanya et al., 2010; Zhu et al., 2003; Subramanya and Bilmes, 2009; Alexandrescu and Kirchhoff, 2009; Liu et al., 2012; Saluja et al., 2014; Tamura et al., 2012; Talukdar et al., 2008; Das and Petrov, 2011), we integrated the graph based semi-supervised algorithm of Subramanya et al. (2010) and adapted their approach to improve on the results from BANNER. We show that our approach achieves a statistically significant improvement in terms of F-measure on the BioCreative II dataset for gene mention tagging.

Semi-supervised learning for gene mention tagging is not without precedent. There has been several semi-supervised approaches for the gene mention task and they have always been more successful than fully supervised approaches (Jiao et al., 2006; Ando, 2007; Campos et al., 2013; Munkhdalai et al., 2015).

Ando (2007) used a semi-supervised approach, Alternative Structure Optimization or ASO, in the BioCreative II gene mention shared task along with other extensions, such as using a lexicon or combining several classifiers. ASO ranked first among all competitors in the shared task competition 2007. Ando reported usage of unlabeled data as the most useful part of his system improving the F-measure of the baseline by 2.09 points where the complete (winning) system had a total improvement of 3.23 points over the baseline CRF (Ando, 2007). Jiao et al. (2006) used conditional entropy over the unlabeled data combined with the conditional likelihood over the labeled data in the objective function of CRF (Jiao et al., 2006). Munkhdalai et al. (2015) trained word representations using Brown clustering (Brown et al., 1992) and word2vec (Mikolov et al., 2013) on MEDLINE and PMC document collections and used them as features along with traditional features in a CRF. Like many of these approaches we

Proceedings of the 15th Workshop on Biomedical Natural Language Processing, pages 27–35,
Berlin, Germany, August 12, 2016. ©2016 Association for Computational Linguistics

also use unlabeled data to augment our baseline CRF model. In all these previous studies the unlabelled data was orders of magnitude more than labelled data and distinct from the test data.

In this paper we take a transductive approach and use the test set as our unlabelled data. Moreover, our approach is orthogonal to all these approaches and can be used to augment many of them. This approach can be easily implemented as a post-processing step in any system that uses a CRF model. Examples of such systems include Gimli (Campos et al., 2013) and BANNER-CHEMNDNER (Munkhdalai et al., 2015). These tools have achieved the highest F-scores in the literature after ASO (Ando, 2007). Our approach relies on the extraction of label distributions from the CRF and augments the decoding algorithm to incorporate the new information about gene mentions from the graph-based learning approach we describe in this paper.

2 Method

Like many previous studies (Leaman et al., 2008; Munkhdalai et al., 2015; Campos et al., 2013), we formulate the gene mention tagging problem as a word level sequence prediction problem, where labels for each word in the input are either Gene-Beginning, Gene-Inside, and Outside (not a gene). This representation is called IOB (for inside-outside-beginning). We applied a graph-based semi-supervised learning (SSL) approach, previously shown effective on a similar labelling task, part-of-speech tagging, for gene mention tagging. (Subramanya et al., 2010)

In graph-based SSL, a graph is constructed to represent partially labelled data. Each node in the graph represents a single word-level gene mention tagging decision and the edges between the nodes represent similarity between the nodes. The goal is to associate probability distributions over the IOB tags to all vertices. Label distributions for vertices that appear in labelled data are estimated based on the reference labels and propagate to vertices for unlabelled data in the graph. These label distributions are combined with the CRF decoding algorithm used for labelling the test data. Graph-based SSL is categorized into inductive and transductive approaches. In inductive settings (e.g. Subramanya et al. (2010)), a model is trained and can be used as-is for unseen data. In transductive settings however, the unlabelled data includes test data. We took a transductive approach in constructing our graph on the union of train set and test set as labelled and unlabelled data.

Since the graph is the cornerstone of the algorithm, let us describe its construction and usage before the overall algorithm.

2.1 Graph Construction

We use the following steps for constructing the graph for the gene mention tagging task adapted from the graph construction for part-of-speech tagging described in Subramanya et al. (2010):

1. Each vertex represents a 3-gram type and the middle word of this 3-gram is the word which is tagged as a gene mention using the IOB tags. The label distribution for this middle word is learned during graph propagation and subsequently combined with the CRF model at test time.

2. A vertex is represented by a vector of point-wise mutual information values between feature instances and its 3-gram type.

3. Edge weights represent the similarity between vertices and are obtained by computing the cosine similarity of feature vectors of their two end vertices.

4. For each vertex only the K nearest neighbours are kept (default = 10).

We considered several feature sets, namely contextual features (Table 1), simplified contextual features (Table 2), all features from the base CRF model, and the most informative features from the base CRF model. We picked the simplified contextual features based on preliminary results using cross-validation on our development set. To represent a vertex v with 3-gram $w_{-1}w_0w_1$, we look at all occurrences of its 3-gram in the text, consider the larger context $w_{-2}w_{-1}w_0w_1w_2$ and get the lemmas of these words. v is represented by a vector of point-wise mutual information values between all possible feature instances (e.g. all possible lemmas for w_{-2}) and $w_{-1}w_0w_1$.

We eliminated extremely frequent features (default $>$ 10,000) to reduce the time complexity of graph construction. This should not affect the structure of the graph substantially because the point-wise mutual information between a feature and any given vertex decreases as the frequency

Description	Feature
3-gram + Context	$w_{-2}\ w_{-1}\ w_0\ w_1\ w_2$
3-gram	$w_{-1}\ w_0\ w_1$
Left Context	$w_{-1}\ w_{-2}$
Right Context	$w_1\ w_2$
Center Word	w_0
Trigram − Center Word	$w_{-1}\ w_1$
Left Word + Right Context	$w_{-1}\ w_1\ w_2$
Left Context + Right Word	$w_{-2}\ w_{-1}\ w_1$

Table 1: Complete set of contextual features.

Description	Feature
Left Context Word	w_{-2}
Left Word	w_{-1}
Center Word	w_0
Right Word	w_1
Right Context Word	w_2

Table 2: Simplified set of contextual features.

of the feature increases leaving extremely frequent features with relatively small weights.

2.2 Graph Propagation

In graph propagation we associate any given vertex u with a label distribution X_u that represents how likely we think each label is for that vertex.

The goal of graph-based SSL is to propagate existing knowledge about the labels through the graph. The initial knowledge about graph nodes is provided by the labeled data and potentially some prior knowledge. Figure 1 shows how graph propagation can assign label distributions to unlabelled vertices and change the label distributions coming from labelled data.

Propagation is accomplished by optimizing an objective function over the label distributions at each node in the graph. The objective function consists of three types of constraints:

1. For any labeled vertex u, the associated label distribution X_u should be close to the reference distribution $\hat{X}_u$ (obtained from labeled data);
2. Adjacent vertices u and k should have similar label distributions X_u and X_k;
3. The label distributions of all vertices should comply with the prior knowledge, if such knowledge exists, or be uniformly distributed, otherwise.

The following objective function represents these three components:

$$\begin{aligned} C(X) \quad = \quad & \sum_{u \in L} ||X_u - \hat{X}_u||_2^2 \\ & + \mu \sum_{u \in V} \sum_{k \in N(u)} w_{u,k} ||X_u - X_k||_2^2 \\ & + \nu \sum_{u \in V} ||X_u - U||_2^2 \qquad (1) \end{aligned}$$

where u and v are nodes in the graph, L is the set of labelled vertices, V is the set of all vertices, $N(u)$ is the set of neighbours of u, U is the uniform distribution over all labels, and μ and ν are weight constants for constraints 2 and 3, respectively. We used Euclidean distance as the distance metric.

While the first two terms in the objective function, and their corresponding constraints make intuitive sense, the uniformity constraint needs further explanation. The rationale behind using distance from uniform distribution is to avoid preferring a label over others in the absence of strong evidence.

The objective function is optimized using stochastic gradient descent. We implement the optimization algorithm for this as described in Subramanya et al. (2010):

$$\begin{aligned} X_i^{(m)}(y) =& \frac{\gamma_i(y)}{k_i} \\ \gamma_i(y) =& \hat{X}_i(y)\delta(i \in L) \\ & + \sum_{k \in N(i)} w_{i,k} X_k^{m-1}(y) + \nu \frac{1}{Y} \qquad (2) \\ k_i =& \delta(i \in L) + \nu + \mu \sum_{k \in N(i)} w_{i,k} \end{aligned}$$

$X_i^{(m)}$ and $X_i^{(m-1)}$ denote the label distributions of vertex i in iterations m and $m-1$, respectively, $\delta(i \in L)$ is 1 if and only if i is a labeled vertex, and Y is the number of labels.

2.3 Overall algorithm

Once propagated the label distributions through the graph, we would need to combine what we learned in the graph with the tagging results from the CRF model. For that we use a self-training algorithm, shown in Figure 2.

On an input of a partially-labeled corpus, we first train a CRF model in a supervised fashion on the labeled data (crf-train, line 1); we then use this trained CRF model to assign label probability distributions to each word in the entire (labeled + unlabeled) corpus (posterior decode, line 4). As a result, each n-gram token in the corpus has a label distribution (the posteriors). For each n-gram type u (a vertex in the graph), we find all instances (n-gram tokens) of u and average over the label distributions of these instances to get a label distribution for u (token to type, line 5). Next, we perform graph-propagation (i.e. we optimize the objective function in equation 1) to learn the label distributions for all vertices. Finally, we linearly interpolate the trained CRF model and the label distributions from the graph:

$$X_{int}(t) = \alpha X_{CRF}(t) + (1-\alpha)X_{Graph}(t) \quad (3)$$

where t is a 3-gram token in a specific sentence, $X_{CRF}(t)$ denotes the posterior probability from the CRF model for the middle word in t, $X_{Graph}(t)$ denotes the label distribution of the 3-gram *type* t after graph propagationn, and $\alpha \in [0, 1]$ is the mixture parameter between the CRF and graph models. The best label for all words in the entire corpus is then found using Viterbi-decoding for the CRF using X_{int} instead of X_{CRF} (viterbi-decode, line 7). Viterbi decoding provides us with the best label for every n-gram token in the unlabeled corpus, which implies that our labeled set has grown to include the unlabeled corpus. We re-train the CRF on this expanded training set (crf-train, line 8); and iterate until convergence.

```
1. $\Lambda^s$ =crf-train($D_l, \Lambda^0$)
2. Set $\Lambda_0^{(t)} = \Lambda^{(s)}$
3. while not converged do
4.     $\{p\}$ =posterior_decode($D_u, \Lambda_{old}$)
5.     $\{q\}$ =token to type($\{p\}$)
6.     $\{q'\}$ =graph propagate($\{q\}$)
7.     $D_u^{(1)}$ =viterbi_decode($\{q'\}, \Lambda_{old}$)
8.     $\Delta_{n+1}^{(t)}$ = crf-train($D_l \cup D_u^{(1)}, \Delta_n^{(t)}$)
9. return $t$
```

Figure 2: Iterative semi-supervised training of CRF with label distributions from the graph. (Subramanya et al., 2010).

Note that the steps indicated by lines 1, 4, and 8 work on the corpus whereas graph propagation in line 6 works on the graph. So, the step in line 5 takes us from corpus to the graph, and the step in line 7 takes us back from the graph to the corpus.

2.4 Integration with BANNER

BANNER (Leaman et al., 2008) is a well-known open-source biomedical named entity recognizer that is widely used. Many studies have used BANNER for gene mention tagging (Li et al., 2015; Hakala et al., 2015; Leaman et al., 2015; Pyysalo et al., 2015; Li et al., 2015; Lee et al., 2014; Leaman et al., 2013) and many have cited it as a biomedical NER system with good performance (Dai et al., 2015; Krallinger et al., 2015; Luo et al., 2016; Gonzalez et al., 2016; Hebbring et al., 2015).

BANNER uses CRF as its machine learning core, and we used it as our base CRF in lines 1 and 8 in Figure 2. We also modified BANNER's source code in order to extract the posterior proba-

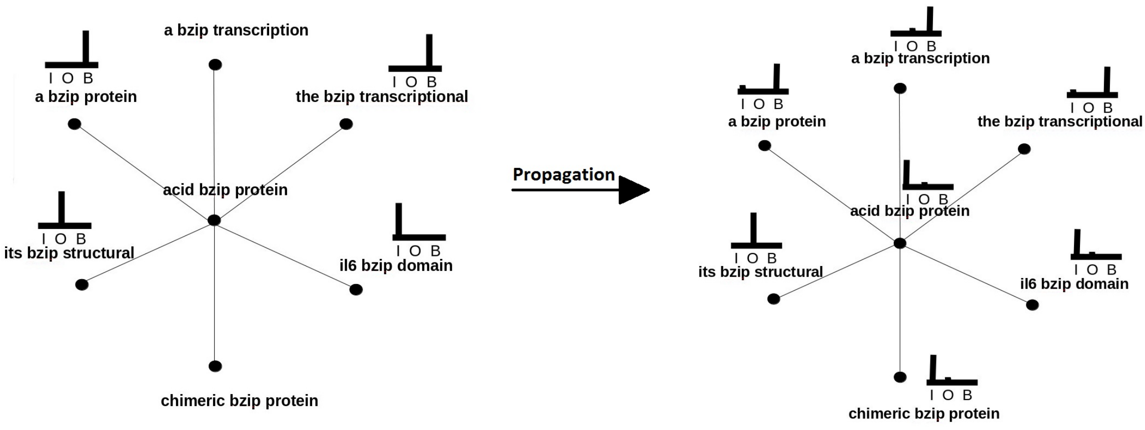

Figure 1: Neighbours of one vertex before and after Propagation. I,O,B stand for Inside-gene, Outside-gene, Beginning-gene.

Category	Method	Precision	Recall	F-Score
Baseline	BANNER	86.27	**85.57**	84.90
Our methods	Graph-based SSL	88.98	82.95	85.86
	Graph + postprocessing	**89.36**	82.95	**86.04**
More recent methods	BANNER-ChemDNER (2015)	88.02	86.08	87.04
	Gimli (2013)	90.22	84.32	87.17
Best performing methods in BioCreative II challenge	(Ando, 2007)	88.48	85.97	87.21
	(Kuo et al., 2007)	89.3	84.49	86.83
	(Huang et al., 2007)	84.93	88.28	86.57

Table 3: Graph-based SSL improves BANNER by increasing the precision.

bilities from the underlying MALLET CRF model (line 4). These probabilities were used in lines 5 through 7 in Figure 2.

Furthermore, the lemmas we used as features in our graph construction (see section 2.1) came from BANNER's lemmatizer.

BANNER also does some post-processing: it discards all the mentions that contain unmatched brackets. We ran our method with and without this post-processing step and verified its utility in our approach as well.

3 Experiments

We show improvements over BANNER on the dataset of BioCreative II Gene Mention Tagging Task. This data set contains 15,000 training sentences and 5,000 test sentences. Annotations are given by the starting character index and finishing character index of the gene in the sentence (space characters are ignored). Some sentences have alternative annotations presented in a separate file.

The upper part of Table 3 shows the results of BANNER; Graph-Based SSL without post-processing; and Graph-Based SSL with post-processing. The hyper-parameters of Graph-Based SSL were chosen by cross-validation over different train/test splits with different hyper-parameters tested for each split ($\alpha = 0.02$, $\mu = 10^{-6}$, $\nu = 10^{-4}$, and number of iterations = 2). Table 3 shows that the improvement we get in F-measure is due to better precision which is further boosted by dropping the candidates with unmatched parentheses (which is our only post-processing step).

The lower part of Table 3 puts our method in context. Although our method is competitive with these best performing methods in the literature, it has not outperformed any of them other than BANNER. Its precision however, is better than all other methods with the exception of Gimli. It would be interesting to integrate the graph-based approach to the ones with CRF as their machine

Type Of error	Number	Examples
FN in both BANNER and Graph	882	SST, R
FP in Graph	120	CD18, kinase, homeobox domain, transforming growth factor - beta, GRK6, POZ/Zn, HPR, E1B 19
FP in BANNER	337	oxidase, dose Ara C, mouse amino acid sequence, Ann Arbor, K1F, wild-type R. sphaeroides 2.4.1, SAS GLM, 1.6-kb cDNA, SH2, E3 ubiquitin, Xp22.3
FN in BANNER	158	LDL, bZIP protein, SL1, NF-kappaB, Ig-like domain, immunoglobulin genes, signal transducer and activator of transcription 1, bcr, ACTH, GFR, wnt
FN in Graph	197	SH3A, EGF, VA1, CBP, Decidual/trophoblast prolactin-related protein, CA 50

Table 4: Qualitative comparison by a human domain expert between BANNER and Graph Propagation. FN: false negatives. FP: false positives.

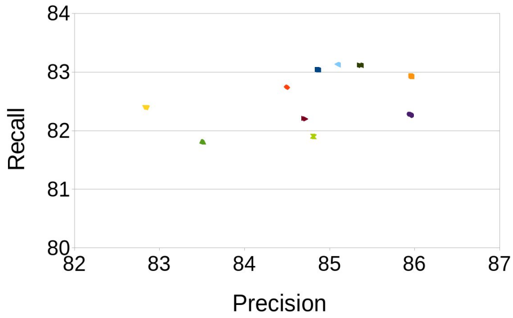

Figure 3: Precision and recall for different train/test splits and hyper-parameter choices. Each color represents a single train/test split. We include only the Pareto optimal points for each split.

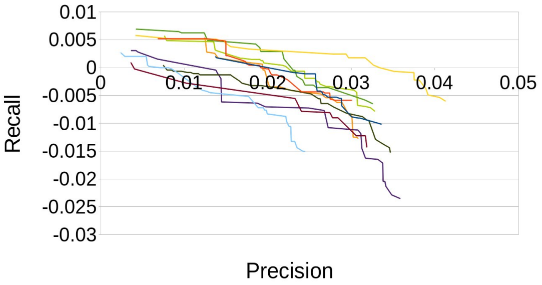

Figure 4: The same points as in Figure 3 shown as the difference from the Banner scores for the same train/test split. The origin in this graph is the BANNER score. Each cluster of points in Figure 3 becomes a line in this graph.

learning core (BANNER-ChemdNER, Gimli, and the approach of (Kuo et al., 2007)) to further test the utility of the graph approach.

3.1 Qualitative analysis

To understand the differences between BANNER and the graph propagation results, a human domain expert compared the errors occurring in their respective outputs. Table 4 shows the number of these errors as well as some examples.

These examples illustrate two important observations. First, there are examples of categories more general than genes in both false positives and false negatives for both systems. For example *Kinase* is a functional group of proteins; *POZ/Zn*, *Ig-like domain*, and *SH2* are protein domains; and *E3 ubiquitin* and *NF-kappaB* are gene families. Anecdotal evidence suggests that this is due to presence of similar annotations in the training/test data set. For example the bZIP protein, a protein family, and Ig-like domain, a gene/protein functional domain were both annotated as genes. This calls for a better gene mention corpus annotated according to more recent gene annotation guidelines. Second, there are some hard to explain false positives in BANNER. Examples include *Ann Arbor*, a city in Michigan, *SAS GLM*, a type of statistical test, and *1.6-kb cDNA*, a molecular length. Our graph-based approach has eliminated these false positives.

3.2 Cross validation study

We conducted extensive cross-validation experiments using different train and test splits in order to explore the hyper-parameter values and to detect trends in the values that were optimal for this task. The results show that graph-propagation consistently improves results over BANNER.

Figures 3 and 4 were created by running graph-propagation over different train and test splits with different hyper-parameter values for each split. For each train/test split, we show only the Pareto optimal points (for each choice of hyper-parameters we include it in the graph only if the performance is not dominated by another choice in both recall and precision). Figure 3 illustrates two points: 1) the precision and recall for the different Pareto optimal points for each train/test split is very similar, and 2) overall the different train/test splits have similar precision and recall values. Figure 4 shows the performance for each train/test split shown as the difference from the BANNER scores for that split. It shows that the precision scores of graph-propagation is always better than the BANNER baseline, while recall is sometimes worse. The F-scores for all train/test splits and for all Pareto optimal points in each split is always better than the BANNER baseline.

We can collect useful statistics about which hyper-parameter values are the most useful in graph-propagation in this task from the extensive set of experiments described above: for different train/test splits and for each split with different hyper-parameter values. Figure 5 shows the number of times different hyper-parameter values have appeared in the set of Pareto optimal points over all the train/test splits.

The hyper-parameter α (see equation 3) controls the interpolation between the BANNER posterior probability over labels and the label distribution from the graph-propagation step. Higher α values would prefer BANNER over graph-propagation. Figure 5 shows that smaller α values are preferred, which implies that the label distribution produced through graph-propagation is found to be more useful than the label distribution produced by BANNER. We also investigated the two extreme cases of $\alpha = 0$ (only graph) and $\alpha = 1.0$ (only BANNER followed by an extra Viterbi decoding step), and observed that both of these options were worse than the BANNER baseline.

In equation (1) higher ν values keep the label distribution at each vertex of the graph closer to the uniform distribution. Higher μ values would allow adjacent vertices to have a greater influence on the label distribution at the vertex. Figure 5 shows that, in our experiments, graph-propagation is sensitive to the values of μ. Lower μ values appear in Pareto optimal points more often. On the other hand, Figure 5 shows that graph-propagation is not as sensitive to different values of ν as long as it is not too high (10^{-1}). This might be due to our setting, where about 73% of vertices are labelled.

We looked for strong correlations between ν values, μ values, and number of iterations in graph propagation and found none.

Finally, for different iteration numbers of graph-propagation, we collected the frequency with which each number appeared in the Pareto optimal results. One iteration of graph-propagation produced 68 Pareto optimal points, two iterations produced 198 points, and three iterations produced 120 points in our experiments. This shows that having more than one iteration of graph-propagation can improve the results.

Our algorithm (Figure 2) has two levels of iterations. One outer iteration (the while loop) and one inner iteration in graph propagation. The numbers mentioned above refer to this inner iteration. All our results reported are for one outer iteration only. Our experiments in this paper were in a transductive setting where the graph was constructed over the test and training data. For this reason we did not experiment extensively with more than one outer iteration. In future work, we plan to experiment with increasing the amount of unlabeled data, and in this setting explore increasing the number of outer iterations.

3.3 A note on scalability

The most time consuming step in our approach was graph construction, where the bootleneck is to compute the edge weights between any possible vertex pairs. We experimented with a naive algorithm, where for every vertex pair the values of feature vectors for shared features were considered, and the cosine similarity was computed. We also implemented a variation on it, where the similarities between all pairs sharing a specific feature instance were computed, and the contributions of individual feature instances were summed to give the final similarity between any given pair. The first algorithm was too slow as expected due to its $O(|V|^2)$ time complexity; the second one was too slow due to high frequency features. This is an important issue since the graph needs to be constructed for our approach to work on a new dataset.

Apart from the graph construction, the graph based approach is as scalable as CRF if a labeled train set is available for the new domain, as the CRF only needs to be trained on the new labelled set. If we wish to adapt the method in a domain where there is no labelled data in the target domain, there is no need for any training.

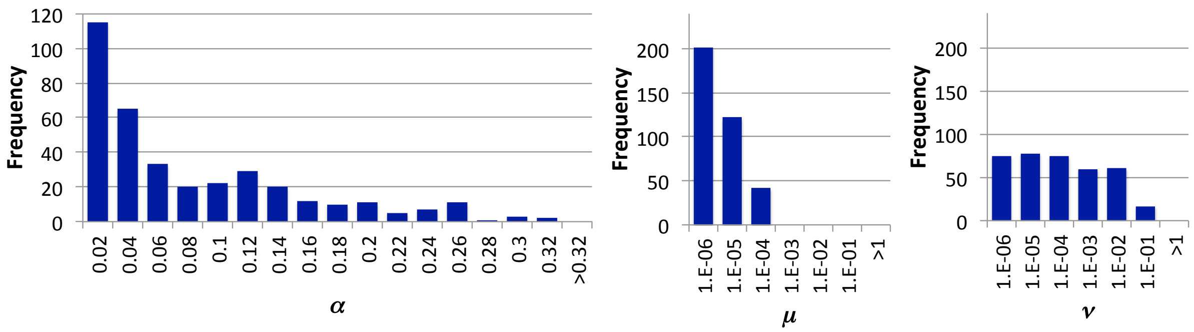

Figure 5: These graphs show the number of times specific hyper-parameter values α, μ and ν appeared in Pareto optimal points over all train/test splits.

.

4 Conclusion and future directions

Our results show that propagating labels from 3-grams present in training set to 3-grams only appearing in the test set can significantly improve BANNER, a well-known frequently used biomedical named entity recognition system for the gene mention tagging task. Our cross-validation study shows the robustness of this improvement. We also presented qualitative comparison by a human domain expert. Our ideas for future work are categorized into three groups:

1. **Adding more unlabelled data**: The only unlabelled data we included in the graph were the test data. Since the success of semi-supervised learning methods is usually due to huge amount of unlabelled data, we plan to use many more PubMed abstracts to construct the graph. This however will be challenging because the graph construction can be time consuming as it was in our case due to high frequency features.

2. **Constructing a better graph**: Contextual features we used to construct our graph are only one of the feature sets that have been shown useful in gene mention tagging task. Other feature sets include orthographic features, contextual features learnt from neural networks, features from parse trees. These features may also prove useful in constructing a graph that represents the similarity between gene mentions. Also, we can preprocess the raw sentences to collapse some collocations into one word so that the middle word in the 3-gram vertices are more meaningful.

3. **Improving the latest approach**: Although BANNER is one of the most frequently used biomedical named entity recognition system, it is not one with the best performance ever. Previous approaches have improved BANNER in a variety of ways, including semi-supervised learning. In particular, Munkhdalia et al. have achieved an F-measure of 87.04 by including word representations learnt from massive unlabelled data as features (Munkhdalai et al., 2015) . We plan to test our approach on their freely available system.

5 Acknowledgements

The authors thank the funding organizations, Genome Canada, British Columbia Cancer Foundation, and Genome British Columbia for their partial support. The research was also partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC RGPIN 262313 and RGPAS 446348) to the fourth author.

References

Andrei Alexandrescu and Katrin Kirchhoff. 2009. Graph-based learning for statistical machine translation. In *NAACL 2009*.

Rie Kubota Ando. 2007. BioCreative II gene mention tagging system at IBM Watson. In *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, volume 23, pages 101–103. Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

David Campos, Sérgio Matos, and José Luís Oliveira. 2013. Gimli: open source and high-performance biomedical name recognition. *BMC bioinformatics*, 14(1):54.

Hong-Jie Dai, Po-Ting Lai, Yung-Chun Chang, and Richard Tzong-Han Tsai. 2015. Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization. *Journal of cheminformatics*, 7(S1):1–10.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 600–609. Association for Computational Linguistics.

Graciela H Gonzalez, Tasnia Tahsin, Britton C Goodale, Anna C Greene, and Casey S Greene. 2016. Recent advances and emerging applications in text and data mining for biomedical discovery. *Briefings in bioinformatics*, 17(1):33–42.

Kai Hakala, Sofie Van Landeghem, Tapio Salakoski, Yves Van de Peer, and Filip Ginter. 2015. Application of the evex resource to event extraction and network construction: Shared task entry and result analysis. *BMC bioinformatics*, 16(Suppl 16):S3.

Scott J Hebbring, Majid Rastegar-Mojarad, Zhan Ye, John Mayer, Crystal Jacobson, and Simon Lin. 2015. Application of clinical text data for phenome-wide association studies (PheWASs). *Bioinformatics*, 31(12):1981–1987.

Han-Shen Huang, Yu-Shi Lin, Kuan-Ting Lin, Cheng-Ju Kuo, Yu-Ming Chang, Bo-Hou Yang, I-Fang Chung, and Chun-Nan Hsu. 2007. High-recall gene mention recognition by unification of multiple backward parsing models. In *Proceedings of the* *second BioCreative challenge evaluation workshop*, volume 23, pages 109–111. Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain.

Feng Jiao, Shaojun Wang, Chi-Hoon Lee, Russell Greiner, and Dale Schuurmans. 2006. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 209–216. Association for Computational Linguistics.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(S1):1–17.

Cheng-Ju Kuo, Yu-Ming Chang, Han-Shen Huang, Kuan-Ting Lin, Bo-Hou Yang, Yu-Shi Lin, Chun-Nan Hsu, and I-Fang Chung. 2007. Rich feature set, unification of bidirectional parsing and dictionary filtering for high f-score gene mention tagging. In *Proceedings of the second BioCreative challenge evaluation workshop*, volume 23, pages 105–107. Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*.

Robert Leaman, Graciela Gonzalez, et al. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Pacific Symposium on Biocomputing*, volume 13, pages 652–663. Citeseer.

Robert Leaman, Rezarta Islamaj Doğan, and Zhiyong Lu. 2013. Dnorm: disease name normalization with pairwise learning to rank. *Bioinformatics*, 29(22):2909–2917.

Robert Leaman, Chih-Hsuan Wei, and Zhiyong Lu. 2015. tmchem: a high performance approach for chemical named entity recognition and normalization. *J. Cheminformatics*, 7(S-1):S3.

Hee-Jin Lee, Tien Cuong Dang, Hyunju Lee, and Jong C Park. 2014. Oncosearch: cancer gene search engine with literature evidence. *Nucleic acids research*, page gku368.

Gang Li, Karen E Ross, Cecilia N Arighi, Yifan Peng, Cathy H Wu, and K Vijay-Shanker. 2015. mirtex: A text mining system for mirna-gene relation extraction. *PLoS Comput Biol*, 11(9):e1004391.

Shujie Liu, Chi-Ho Li, Mu Li, and Ming Zhou. 2012. Learning translation consensus with structured label propagation. In *ACL 2012*.

Yuan Luo, Özlem Uzuner, and Peter Szolovits. 2016. Bridging semantics and syntax with graph algorithmsstate-of-the-art of extracting biomedical relations. *Briefings in bioinformatics*, page bbw001.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Tsendsuren Munkhdalai, Meijing Li, Khuyagbaatar Batsuren, Hyeon Park, Nak Choi, and Keun Ho Ryu. 2015. Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. *J. Cheminformatics*, 7(S-1):S9.

Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Jun'ichi Tsujii, and Sophia Ananiadou. 2015. Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013. *BMC bioinformatics*, 16(Suppl 10):S2.

Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. 2014. Graph-based semi-supervised learning of translation models from monolingual data. In *ACL 2014*.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL*.

Amarnag Subramanya and Jeff A Bilmes. 2009. Entropic graph regularization in non-parametric semi-supervised classification. In *Advances in Neural Information Processing Systems*, pages 1803–1811.

Amarnag Subramanya, Slav Petrov, and Fernando Pereira. 2010. Efficient graph-based semi-supervised learning of structured tagging models. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 167–176. Association for Computational Linguistics.

Partha Pratim Talukdar, Joseph Reisinger, Marius Paşca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *EMNLP 2008*.

Akihiro Tamura, Taro Watanabe, and Eiichiro Sumita. 2012. Bilingual lexicon extraction from comparable corpora using label propagation. In *EMNLP-CoNLL 2012*.

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML*, volume 3, pages 912–919.