# GraphNER: Using Corpus Level Similarities and Graph Propagation for Named Entity Recognition*

Golnar Sheikhshab[1,2], Elizabeth Starks[2], Aly Karsan[2], Readman Chiu[2], Anoop Sarkar[1], and Inanc Birol[1,2]

***Abstract*— The rapidly growing amount of research papers in computational biology makes it difficult for researchers to keep up to date on new results. The motivation behind this paper is to use natural language processing to automatically understand relevant concepts from the large amount of text data in published papers in computational biology. As a proof-of-concept, we focus on the gene mention detection task, which allows us to identify genes that are being discussed in papers, making it possible to search for concepts like genes rather than searching on words. In this paper we introduce GraphNER, a semi-supervised machine learning model for named entity recognition (NER). In particular, we use GraphNER to identify gene mentions in natural language data such as biomedical papers. It combines training data where the gene mentions are identified by human experts and unlabelled data that contains many other relevant gene mentions. The labeled and unlabeled data are linked together using similarities between $n$-grams that occur in the two data sources (an $n$-gram is a contiguous sequence of $n$ words in the text). GraphNER uses the information gleaned from this graph, and combines it with a conditional random field (CRF) model for NER. We consider two different CRF-based NER systems on two different datasets combined with our graph model for semi-supervised learning for the task of gene mention detection. We show that GraphNER consistently improves the overall quality of gene mention detection due to its higher precision. GraphNER is freely available at `http://www.bcgsc.ca/platform/bioinfo/software/graphner`.**

## I. Introduction

Detecting the named entities in a document is often the first step in many natural language processing (NLP) tasks, such as relation extraction and knowledge discovery. Named entities of interest in the biomedical domain include mentions of genes, mutations, proteins, and diseases in text.

Current approaches formulate named entity recognition (NER) as a sequence tagging problem, where the input is a sentence represented as a sequence of words $x_1, x_2, ..., x_l$, and the output is a sequence of corresponding tags $t_1, t_2, ..., t_l$. The tag $t_i$ also marks if the term $x_i$ represents the *beginning*, *inside*, or *outside* of a named entity type, the first two corresponding to the first, and internal terms of a mention, respectively, and the last one indicating that it is not part of a named entity mention. Hence, the tag-set consists of two tag types for any of the desired entity types (B and I), and a tag type for all other terms (O). For example, if the task is to detect genes, mutations, and diseases, the tag-set will be {B-Gene, I-Gene, B-Mutation, I-Mutation, B-Disease, I-Disease, O}.

The following sentence shows a sample input/output for a gene mention detection task:

```
Recently/O ,/O the/O mutation/O
of/O lymphocyte/B adaptor/I
protein/I (/O LNK/B or/O SH2B3/B
)/O was/O detected/O in/O MPN/O
```

where B, I, and O are used for B-Gene, I-Gene, and O, since we are only considering gene mentions in this example. Based on the output tags, we can conclude there are three distinct gene mentions in this sentence: "lymphocyte adaptor protein", "LNK", and "SH2B3".

Conditional random field (CRF) [12] is one of the most successful supervised methods for sequence tagging problems. Many popular NER systems, including some popular biomedical systems such as BANNER [15], Gimli [4], and BANNER-ChemDNER [23], are based on CRF.

The CRF model is an undirected graphical model for the conditional probability of the entire output sequence $\mathbf{t}$ given the entire input sequence $\mathbf{x}$. In contrast to CRF, Markov Random Fields model the joint probability of both sequences. CRFs can be defined on any factor graph. Chain CRFs, where the factor graph is a chain, have useful properties for NLP, as they are suitable for sequence modelling, and are computationally tractable using dynamic programming. In a chain CRF, $t_i$ (the $i^{\text{th}}$ element of the sequence $\mathbf{t}$) depends on $x$ (the entire input sequence) and previous $d$ labels, $d$ referring to the order of the CRF model. Therefore, in a second order CRF, the probability of having tag sequence $\mathbf{t}$ for sentence $\mathbf{x}$ is:

$$p(\mathbf{t}|\mathbf{x}) = \frac{exp(score(\mathbf{t}|\mathbf{x}))}{\sum_{t'} exp(score(\mathbf{t'}|\mathbf{x}))})$$

where

$$score(\mathbf{t}|\mathbf{x}) = \sum_{j=1}^{m}\sum_{i=1}^{n} \lambda_j f_j(x, i, t_i, t_{i-1})$$

and $\lambda_j$ is the hyper-parameter corresponding to $f_j$, the $j^{th}$ of the $m$ feature extractors that extract features from the sequence, the position, and the previous labels.

Chain CRF's are considered efficient models, where training is done by maximizing the conditional log likelihood of labelled data. Then, given the model parameters, the optimal

* This work has been partially supported by Genome Canada, Genome British Columbia and British Columbia Cancer Foundation. 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.

[1]Golnar Sheikhshab(gsheikhs@sfu.ca), Anoop Sarkar, and Inanc Birol are affiliated with School of Computing Science, Simon Fraser University, 8888 University Dr, Burnaby, BC

[2]Golnar Sheikhshab, Elizabeth Starks, Readman Chiu, Aly Karsan, and Inanc Birol are affiliated with Canada's Michael Smith Genome Sciences Centre, British Columbia Cancer Agency

978-1-5386-5555-9/18/$31.00 ©2018 IEEE
DOI 10.1109/IPDPSW.2018.00047

tag sequence can be "decoded" using the Viterbi dynamic-programming algorithm [33].

One weakness of a CRF model is that it is fully supervised, and it ignores information outside the sentence. To mitigate, CRF models can be extended to include unlabelled data using graph-based semi-supervised learning (SSL) methods, as demonstrated in many NLP applications [35], [29], [1], [20], [27], [32], [31], [7], including a sequence tagging problem for part of speech tagging [30].

In graph-based SSL, a graph connects labelled and unlabelled data points in a large partially labelled dataset, and pushes similar data-points towards similar labels. This leverages unlabelled data, which is usually more readily available than labelled data, and takes corpus-level similarities into account.

Motivated by the success of Graph-based SSL in other sequence tagging tasks, we extended the algorithm of [30] from part of speech tagging to named entity recognition, and implemented GraphNER, a biomedical named entity recognition tool. GraphNER inputs the probabilistic output of CRF-based systems such as BANNER or BANNER-ChemDNER, and improves their label assignments using a graph constructed over a given partially labelled corpus.

We compared our work to the best-performing methods in the BioCreative II gene mention (BC2GM) shared task, as well as more recent methods reporting results on the corpus of that shared task. The best performing method in the BC2GM shared task was a semi-supervised method from IBM Watson Research Center [2] reporting a F-score of 87.21%. A more recent semi-supervised tool, BANNER-ChemDNER [23], takes advantage of abundant unlabelled data by using Brown clustering [3] and word-to-vector (word2vec) embeddings [22], leveraging these embeddings as features in its otherwise supervised machine learning core (CRF). Brown clustering and word2vec embeddings both capture the syntactic and semantic similarity of words. Brown clustering constructs a cluster hierarchy over the words by maximizing the mutual information of bi-grams, whereas word2vec embeddings are the hidden layer of a neural network, trained to predict each word by using the words in its context.

We also evaluated a strong neural network method for named entity recognition, based on a bi-directional long short-term memory with a CFR layer, LSTM-CRF [13], and benchmarked it on the BC2GM corpus. LSTM-CRF has been shown to outperform bi-directional LSTM and simplified LSTM (S-LSTM) [13]. It achieves comparable results with LSTM with a convolutional neural network layer (LSTM-CNN) [5] on the same benchmark corpus in the absence of lexicons.

While we have previously shown the efficacy of graph propagation in improving BANNER on BC2GM corpus [28], the contributions of this paper are as follows.

1) We introduce GraphNER as a gene mention detection tool for use on biomedical literature (abstract or full text).
2) We show that GraphNER improves both CRF-based BANNER and BANNER-ChemDNER, providing statistically significant gains in performance.
3) We demonstrate that GraphNER's competitive performance on the BC2GM corpus can be replicated on an orthogonal dataset.
4) We show that this performance was reached incurring only modest execution time and memory use compared to resources used for training and testing the CRF models.

## II. Approach

We developed GraphNER, a graph-based SSL for named entity recognition, specifically for gene mention detection. Following the practice introduced by Subramanya et al. [30], we first construct a graph where the vertices are 3-grams, and edge-weights encode corpus-level similarity of these 3-grams. We use this graph to push similar 3-grams towards taking similar labels. This is a semi-supervised method, because the graph is constructed on labelled and unlabelled data. While this method is theoretically capable of using abundant unlabelled data, we chose a transductive approach: the only unlabelled data we use in graph construction is the test data. The rationale behind this decision is the time complexity of graph construction, as the scalability of graph construction for large datasets remains an open problem.

Algorithm 1 shows the train and test procedure of GraphNER. The training stage of the algorithm comprises an expansion of a base CRF model, where we scan the labelled data ($D_l$) and compute an average label distribution $X_{\text{ref}}(v)$ for any 3-grams $v$ that appear in $D_l$ (we call the set of such 3-grams $V_l$).

The testing procedure is more complex than the training stage. First, we run the CRF to extract the posteriors and transition probabilities. Then, we compute the average of these posteriors for each possible 3-gram $v$ to form the initial belief about label distribution of $v$, which we will call $X(v)$. These distributions are then propagated on the graph to ensure similar 3-grams get similar distributions, as described in further detail in the Graph propagation section. Finally, we assign labels to all words $w$ that appear in the context of $(w_{-1}, w, w_1)$ in an input sentence $S$, using a mixture of graph and CRF results. That is, we combine the graph's belief, $P_s(S, i)$ ($i$ indicating the index that corresponds to a given word $w$ in the sentence $S$), and the CRF's belief, $X(w_{-1}, w, w_1)$, about a label of $w$ by computing $\alpha P_s(S, i) + (1 - \alpha)X(w_{-1}, w, w_1)$, and get a final Viterbi decoding on the sequence of these values and transition probabilities of CRF. For an illustration of these procedures, see Figure 1.

Our algorithm differs from [30] in that we have a transductive setting where we train and test once, whereas they have an inductive setting: they expand the labelled data-set by treating the output of Viterbi decoding (line 9) as correct and iterating over the train and test procedures, overwriting these labels until convergence or the 10th iteration. Note that we follow the same iterative graph propagation algorithm (elaborated in the Graph propagation section) as they do

**labelled data:**
drug/O response/O was/O significant/O in/O wilms/B tumor/I -/I 1/I positive/O patients/O ./O
we/O observed/O the/O following/O mutations/O in/O wilms/B tumor/I -/I 1/I ./O
we/O did/O not/O observe/O this/O mutation/O in/O the/O patient/O '/O s/O tumor/O -/O 1/O subclone/O ./O

**unlabelled data:**
wilm ' s tumor - 1 ( wt1 ) gene was highly expressed .
we did not observe this mutation in the patient ' s tumor - 2 subclone .

**Input:** labelled and unlabelled data that includes the sentences shown above and a graph constructed over the whole partially labelled corpus. Part of the input graph is shown in (a).

**Output:** BIO labels for every word in all unlabelled sentences. We focus on "-" in the two unlabelled sentences shown above.

**After training:** Reference labels, the average label distributions based on the labelled data are associated with some vertices in the graph as shown in (b).

**After line 5 in Algorithm 1:** The following posterior probability distributions are extracted from CRF. For "-" in the first sentence we get (B,I,O)=(0,0.45,0.55) and for "-" in the second sentence we get (B,I,O)=(0,0.15,0.85). Label O is preferred for both instances of "-". In the second case this preference is correct and much more enunciated. In the first case however, O is not the right label. One reason that the CRF could make such a mistake is that it has seen the sequence ¡' s tumor - 1¿ before in the labelled data with annotation O for "-".

**After line 6:** The average of extracted posterior probabilities are put on the graph as illustrated in (c).

**After line 7:** As a result of graph propagation, the vertex [tumor - 1] has a label distribution that peaks at I (note the label distributions in (d)). The reason is that this vertex will have many neighbor vertices where I is preferred. Examples of such neighbors are [wilms tumor -] that is shown in the figure and [wilms tumour 1], [wilms ' tumor], and [wilms tumor ,] that are not shown due to lack of space.

**After line 8:** The posteriors extracted from CRF and the distributions on the graph after propagation are linearly combined with coefficients $\alpha$ and $1-\alpha$ respectively. Let us pick the coefficient $\alpha$ to be 0.1 since smaller $\alpha$ values were consistently preferred in our cross validations. This will give us new distributions for "-" in the first and second sentences: (B,I,O)=(0,0.77,0.23) for the first "-" and (B,I,O)=(0,0.24,0.76) for the second "-".

**After line 9:** Labels of both instances of "-" are correct after Viterbi algorithm chooses the most probable tag sequence for the unlabelled sentences.

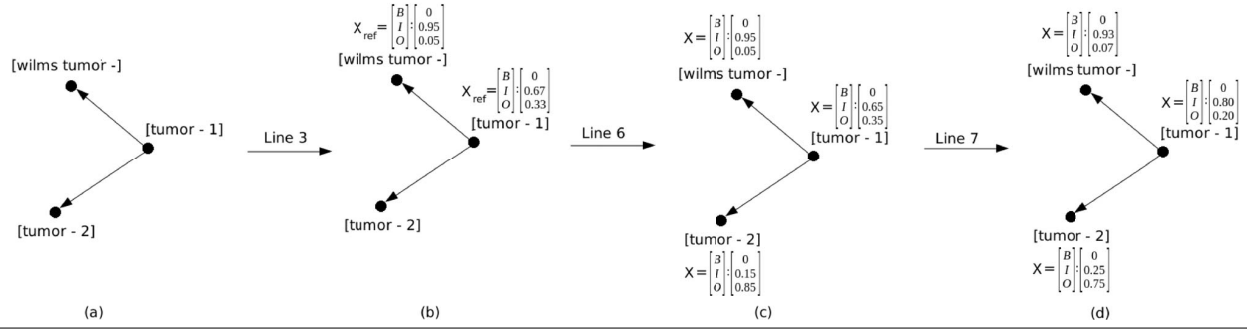


Fig. 1. Illustration of Algorithm 1 with an example.

and the hyper-parameter $\#iterations$ (line 7) refers to this iterative algorithm.

**Algorithm 1** GraphNER

1: **procedure** TRAIN
2: CRF_train($D_l$)
3: $X_{\text{ref}}, V_l \leftarrow$ Set_ReferenceDistributions($D_l$)
4: **procedure** TEST
5: $P_s, T_s \leftarrow$ CRF_Posteriors_And_Transitions($D_l \cup D_u$)
6: X $\leftarrow$ Average($P_s$,$V$)
7: X $\leftarrow$ Propagate($X, X_{\text{ref}}, \mu, \nu, \#iterations$)
8: $P'_s \leftarrow$ Combine ($P_s, X, V, \alpha$)
9: finalLabels $\leftarrow$ Viterbi($P'_s, T_s$)

### A. Graph propagation

The intuition behind graph propagation is that similar text elements should have similar labels, and we use representations of text elements on a graph to infer similarities. Using the topology of the graph, we push the label distribution of a vertex towards the label distributions of its neighbours. We would also want the label distribution of $V_l$ to be similar to their reference distributions ($X_{\text{ref}}$). Finally, when available, we would want to incorporate prior knowledge as well. In general, it would be desirable to prefer one label over others only if there is strong evidence. In our implementation, we achieve this by enforcing a preference for staying close to a uniform distribution.

The following loss function combines the mentioned intuitions.

$$\begin{aligned} C(X) \quad = \quad & \sum_{u \in V_l} ||X(u) - X_{\text{ref}}(u)||_2^2 \\ & + \mu \sum_{u \in V} \sum_{k \in N(u)} w_{u,k} ||X(u) - X(k)||_2^2 \\ & + \nu \sum_{u \in V} ||X(u) - U||_2^2. \end{aligned} \tag{1}$$

where $N(u)$ refers to the set of neighbours of a vertex $u$ in the graph, $w_{u,k}$ stands for the weight of the edge between vertices $u$ and $k$, and $U$ denotes the uniform distribution. All other terms are as they were defined above.

Now, since this is a loss function, we would want to minimize it by setting its derivative to zero. Taking Euclidean distance as the distance function, we can calculate the

derivative of $C$ with respect to $X(i)_y$ (the probability of label $y$ in vertex $i$) as follows.

$$\begin{aligned}\frac{\partial C}{\partial X(i)_y} &= 2\{\delta(i \in V_l))(X(i)_y - X_{\text{ref}}(i)_y) \\ &\quad +\mu \sum_{k \in N(i)} w_{i,k}(X(i)_y - X(k)_y) \\ &\quad +\nu[(X(i)_y) - \frac{1}{Y}]\}.\end{aligned}$$

where $Y$ is the number of labels and $\delta$ is the identification function: $\delta(P) = 1$ if and only if $P$ is true.

Setting this derivative to zero, we will get the update rule for $X(i)_y$ in graph propagation (line 7 in Algorithm 1):

$$\begin{aligned}X(i)_y^{\text{new}} =& \frac{\gamma_i(y)}{k_i} \\ \gamma_i(y) =& X_{\text{ref}}(i)_y \delta(i \in V_l) \\ &+ \mu \sum_{k \in N(i)} w_{i,k} X(k)_y + \nu \frac{1}{Y} \\ k_i =& \delta(i \in V_l) + \nu + \mu \sum_{k \in N(i)} w_{i,k}.\end{aligned} \tag{2}$$

Graph propagation is performed by iteratively updating label distributions using equation 2 and involves three hyper-parameters:$\mu$, $\nu$, and number of iterations. These hyper-parameters can be tuned by cross-validation.

### B. CRF Models used by GraphNER

We used two different CRF-based NER systems as the base model that is extended by the GraphNER model; that is in Algorithm 1, we used these tools in lieu of CRF. In order to extract posterior and transition probabilities, we have modified the source code of these tools. The modified version of these tools is included in our publicly available implementation.

*1) BANNER:* BANNER [15] is a popular biomedical named entity recognition system that is frequently cited [6], [11], [21], [8], [10], and is used for gene mention tagging [19], [9], [16], [25], [19], [17], [14]. The features used in our graph construction were extracted from BANNER.

*2) BANNER-ChemDNER:* BANNER-ChemDNER [23] is a tool built on BANNER, and uses features extracted from large unlabelled datasets. As such, it is considered a semi-supervised version of BANNER.

### C. Graph construction

The central idea in GraphNER is to have a graph that tells us what data points are similar, so that we can assign similar labels to them. We followed a popular approach in graph-based semi-supervised learning for NLP applications (specifically, the ones that can be formulated as tagging problems), putting 3-grams as vertices, and representing them with a vector of feature values. That is, a vertex is represented as a vector of pointwise mutual information between the 3-gram associated with it and possible feature instances such as surrounding words.

The edge weight between two vertices is the cosine similarity of their vector representations. The graph is usually kept sparse by keeping only $k$ nearest neighbors for each vertex, which means the final graph is a directed one.

Different choices of feature sets change the vector representations, and consequently the edge weights and structure of the graph, leading to a different performance in GraphNER. We considered using all features extracted by BANNER (All-features), only lemmas of the words in a window of length 5 (Lexical-features), and features that have high mutual information with the tag assigned by BANNER (MI greater than some fixed threshold).

### D. Datasets

*1) BC2GM corpus:* This dataset, introduced for the BioCreative II shared task in 2006, contains 15,000 training and 5,000 test sentences from published abstracts. Annotations are given by the starting and finishing character indices of genes in sentences. The space characters are ignored. Some sentences have alternative annotations presented in a separate file. This dataset is publicly available, and many studies have reported results on it, including the leading studies we compared against in this work [4], [2], [26].

*2) AML corpus:* This is a collection of 80 full text articles related to acute myeloid leukemia (AML) clinical variant interpretation. The annotations are provided in the same format as the BC2GM corpus.

We divided the corpus into train and test sets by randomly selecting 22 full text articles for the test set, and we placed the rest in the train set. The training set contains 10,504 sentences and the test set contains 3,952.

### E. A note on time complexity

We can discuss the time complexity of GraphNER in three different phases: graph construction, model training, and model testing.

*1) Time complexity of graph construction:* Graph construction consists of a. constructing the feature vectors for vertices and b. computing the cosine similarity of every pair and keeping only K nearest neighbours for each vertex.

In the first step, constructing feature vectors, we need to go through all 3-gram tokens in the corpus and try to extract all relevant features. The time complexity of this step is a linear function of the number of 3-gram tokens in the corpus ($N$). The constant in the linear function depends on not only the number of features ($f$) that we need to extract for each 3-gram token, but also the difficulty of that feature extraction.

In the second step, we will get the cosine similarity of all possible pairs of vertices (unique 3-grams). Therefore, the time complexity depends on the number of vertices ($V$) and the size of the feature vector($F$). The feature vector can be large, because there are as many elements in the feature vector as there are feature instances, which can be as many as $Nf$. The worst case scenario of size $Nf$ would happen if all the feature instances for any 3-gram token are unique. Computing the cosine similarity between all pairs of vertices would have a time complexity of $O(V^2F)$. Keeping only $K$

nearest neighbours adds another $K$ or ($log(K)$ if we use a heap data structure) factor to the time complexity.

Overall, the time complexity of graph construction can be summarized as $O(Nf + V^2FK)$.

*2) Added time complexity to train and test:* As Algorithm 1 shows, the train and test contain training and testing CRF as well as other steps. The question is how much GraphNER adds to the time complexity of CRF.

In training, GraphNER needs to set the reference distributions (line 3 of Algorithm 1), which involves going through all 3-grams of training set, keeping the number of occurrences and the number of any tag for all vertices (unique 3-grams), and finally loop over all vertices and divide the number of tags by the number of occurrences to get the distributions. The time complexity that this procedure adds is of $O(N_l+V_l)$ where $N_l$ and $V_l$ are the numbers of 3-gram tokens and unique 3-grams in the training set.

In testing, we do a similar averaging over all the posteriors (line 6 of Algorithm 1), adding a time complexity of $O(N + V)$ where $N$ and $V$ are the numbers of 3-gram tokens and unique 3-grams in both training and test sets. Then, graph propagation happens that consists of repeating over equation 2 for $V.\#Iterations$ times. Equation 2 itself is of $O(K)$, then the graph-propagation overall is of $O(VK\#Iterations)$. The next line (line 8 of Algorithm 1) involves going through all 3-gram tokens and doing a weighted sum, so the complexity of that is $O(N)$ and finally the complexity of the last line, the Viterbi, is going to be also $O(N)$ because Viterbi has the complexity of $O(LQ^2)$ where $L$ is the length of the sentence and $Q$ is the number of tags which is only 3 in our case.

So, overall, the added time complexity of GraphNER is only $O(N_l + V_l)$ for training and $O(N + VK\#Iterations)$ for testing.

## III. Results

We report the precision, recall, and F-Score obtained from the evaluation script of the Biocreative II gene mention task. The script compares detections with primary gene mentions and their alternatives, and counts exact matches as true positives. Alternative annotations were present in the BC2GM corpus, but not in the AML corpus. The number of false negatives will be the number of primary gene mentions minus the number of true positives; and the number of false positives will be the number of detections minus the number of true positives.

As shown in Table I, GraphNER improves both baselines on the BC2GM Corpus. A significance test with the sigf tool [24] (discussed in further detail in section III-A) revealed that these F-Score improvements were statistically significant.

Table I also shows that we were not able to exactly replicate the published BANNER-ChemDNER's performance on this task. While the authors have reported an F-score of 87.04%, we obtained a slightly lower performance, at 86.49%. Regardless, plugging BANNER-ChemDNER into GraphNER led to an F-score (87.34%) that is greater than published F-score of BANNER-ChemDNER on BC2GM.

It is also worth mentioning that when applying the two leading neural-net based methods [13], [26] in the literature on BC2GM, we had to train them on a subset of train set as they both need a dev-set. We divided the train set into a 12000-sentence train subset and a 3000-sentence dev-set. The fact that we did not have access to the exact train/dev sets that Rei et al. [26] used explains the difference in our F-Scores.

Performance of GraphNER on the AML corpus is reported in Table II. Performance of the baseline CRF based supervised learning systems and GraphNER were substantially higher for the AML corpus relative to the BC2GM corpus. These performance differences were expected, because there were multiple differences in the article curation and manual annotation procedures for the two corpora. The BC2GM corpus was curated broadly from articles in the field of biology, whereas the AML corpus was restricted to human clinical genetics articles. In the general field of biology, gene names may be used inconsistently with a variety of notation styles. Clinical genetics articles have a more standardized discourse about genes, and articles in this field preferentially use a gene nomenclature maintained by HGNC for human genes. This standardized nomenclature would simplify the manual annotation task for the annotators, and would likely improve the performance of the named entity recognition tools as well. BC2GM annotations were performed by undergraduate students with limited training and limited subject knowledge, whereas the AML corpus was curated and edited

TABLE I

Results on the BC2GM corpus. Bold numbers indicate the best performance in each metric. F-score is often used to measure the trade off between the precision and recall, and is the harmonic mean of these metrics.

| Category | Method | Precision (%) | Recall (%) | F-Score (%) |
|---|---|---|---|---|
| Published in the literature | Ando (2007) | 88.48 | 85.97 | 87.21 |
| | Gimli (2013) | **90.22** | 84.32 | 87.17 |
| | BANNER-ChemDNER (2015) | 88.02 | 86.08 | 87.04 |
| | Rei et al. (2016) | - | - | **87.99** |
| Obtained from existing methods | LSTM-CRF | 88.80 | 84.28 | 86.48 |
| | Rei et al. (2016) | 87.04 | **88.72** | 87.77 |
| | BANNER | 86.88 | 82.02 | 84.38 |
| | BANNER-ChemDNER | 87.51 | 85.49 | 86.49 |
| GraphNER | CRF=BANNER | 90.21 | 81.85 | 85.83* |
| | CRF=BANNER-ChemDNER | 89.18 | 85.57 | 87.34* |

TABLE II
RESULTS ON THE AML CORPUS.

| Category | Method | Precision (%) | Recall (%) | F-Score (%) |
|---|---|---|---|---|
| Baselines | BANNER | 96.56 | 94.56 | 95.55 |
| | BANNER-ChemDNER | 97.29 | 96.00 | 96.64 |
| GraphNER | CRF=BANNER | 97.56* | 94.46 | 95.98 |
| | CRF=BANNER-ChemDNER | **97.68*** | **96.08** | **96.87** |

by subject experts in the clinical genetics domain. Due to the difference in expertise in the annotators for the two projects, we expected a higher error rate in the BC2GM manual annotations.

Nonetheless, GraphNER has improved both baselines on AML corpus by showing higher precision. As discussed in detail in section III-A, the improvements in precision were statistically significant.

We also applied the neural-net-based state of the art on BC2GM (character-based bi-directional LSTM with attention mechanism [26]) on AML data set. Since it needs a dev set, we partitioned AML train set to train/dev subsets (82%/18% in sentences). We also re-trained BANNER-ChemDNER and GraphNER with BANNER-ChemDNER on the new smaller train subset to have a fair comparison. Their system achieved an F-Score of 93.62 which was lower than both BANNER-ChemDNER (F-Score=94.32) and GraphNER with BANNER-ChemDNER (F-Score=94.54) when trained on the smaller train subset. Note that GraphNER with semi-supervised learning improved over supervised learning.

The hyper-parameters used to generate reported results for GraphNER were all chosen by cross-validation over different train:test splits. Table IV shows the parameters used for each of the systems.

Finally it is worth mentioning that all results reported in Table I and Table II were obtained with second order CRF's and using java version 1.7. However, while we obtained different numbers for different CRF orders (1 or 2) or java versions (1.7 or 1.8), GraphNER always improved both baselines, and this improvement was consistently due to higher precision.

TABLE IV
HYPERPARAMETERS OF GRAPHNER CHOSEN BY CROSS-VALIDATION.

| Corpus | CRF Model | GraphNER hyperparameters ($\alpha$, $\mu$, $\nu$,$\#iterations$) |
|---|---|---|
| AML | BANNER | $(0.02, 10^{-6}, 10^{-6}, 2)$ |
| | BANNER-ChemDNER | $(0.02, 10^{-6}, 10^{-6}, 2)$ |
| BC2GM | BANNER | $(0.02, 10^{-6}, 10^{-4}, 2)$ |
| | BANNER-ChemDNER | $(0.02, 10^{-6}, 10^{-6}, 3)$ |

### A. Significance testing of the results

We used sigf [24] to test for significant changes to precision, recall and F-scores with the addition of GraphNER. *sigf* is an implementation of an assumption-free significance test based on randomization [34]. When testing the significance of difference in performance of two models m1 and m2, sigf repeatedly constructs statistically identical models m3 and m4 by taking the predictions that are produced by m1 or m2 but not both of them, and randomly assigning those predictions to either m3 or m4. How often m3 and m4 produce results that are at least as different as results of m1 and m2 is interpreted as the p-value in the significance test.

We used sigf with 10,000 repetitions to test the null hypotheses presented in Table V. Bonferroni correction for multiple testing changes the first $\alpha = 0.05$ to $\alpha = 0.006$. The F-score improvement of GraphNER over both baselines (BANNER and BANNER_ChemDNER) is statistically significant while working with the BC2GM corpus. Although the F-score and recall improvements of GraphNER over BANNER and BANNER_ChemDNER were not statistically significant for the AML corpus, the improvements in precision were significant.

TABLE III
EFFECT OF CHOICE OF FEATURE SETS USED IN GRAPH CONSTRUCTION. THE BOLD FIGURES INDICATE BEST PERFORMANCE FOR GRAPHNER USING BANNER AND BANNER-CHEMDNER MODELS.

| Method | CRF Model | Vector-Representation | K | F-Score (%) |
|---|---|---|---|---|
| BANNER | - | - | 10 | 84.38 |
| BANNER-ChemDNER | - | - | 10 | 86.49 |
| GraphNER | BANNER | All-features | 10 | **85.83** |
| | BANNER | Lexical-features | 10 | 85.43 |
| | BANNER | MI $> 0.005$ | 10 | 85.01 |
| | BANNER | MI $> 0.01$ | 10 | 85.00 |
| | BANNER-ChemDNER | All-features | 10 | **87.34** |
| | BANNER-ChemDNER | Lexical-features | 10 | 87.23 |
| | BANNER-ChemDNER | MI $> 0.005$ | 10 | 87.09 |
| | BANNER-ChemDNER | MI $> 0.01$ | 10 | 87.12 |
| | BANNER-ChemDNER | All-features | 5 | 87.32 |

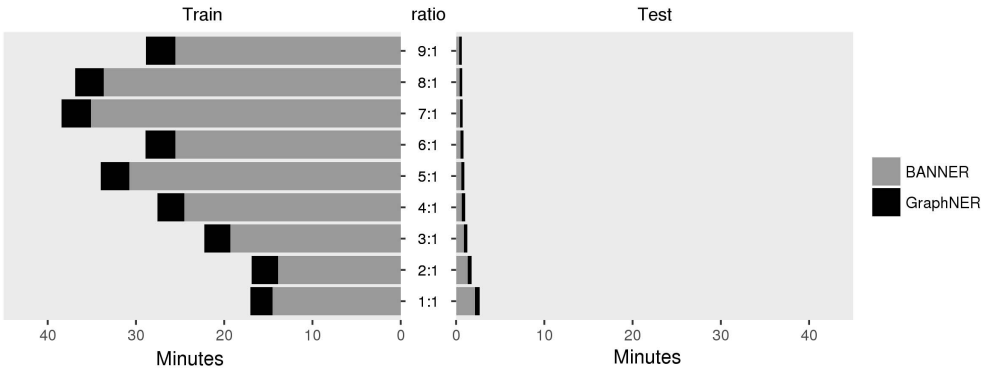


Fig. 2. Time cost to train and test BANNER and GraphNER on the BC2GM dataset. The ratio indicates the relative sizes of train:test partitions.

### B. Effect of different vertex representations

Table III shows how GraphNER improves upon the performance of purely supervised models such as BANNER and BANNER-ChemDNER on the BC2GM corpus by using semi-supervised graph-propagation using different feature sets. The hyper-parameters used are shown in Table IV. It is interesting that while using all features led to the best results with both BANNER and BANNER-ChemDNER, GraphNER consistently improved the baselines even when only 40 ($MI > 0.01$) and 85 ($MI > 0.005$) features were used in graph construction.

Another variable in graph construction is $K$, the degree of the graph. While $K = 10$ was default, we changed $K$ to be 5 for one of the graphs (all features used for vector representation) and saw a small degradation in F-score (going from 87.34 to 87.32).

### C. Added time and memory cost over supervised CRFs

In our tests, GraphNER has consistently improved both BANNER and BANNER-ChemDNER supervised CRF models, and this improvement is achieved with a small additional run time cost over the purely supervised models. Figure 2 shows the extra time GraphNER needs to train and test over BANNER, using different ratios of train:test splits of the BC2GM corpus. All experiments were done in a GNU/Linux environment on a Dell Precision Tower 7910 with 16 Intel(R) Xeon(R) CPU E5-2620 v3 @ 2.40GHz cores and 64GB of RAM.

We observed similar patterns when we experimented with the BANNER-ChemDNER as the supervised model and with the AML dataset. In all our experiments we used the all-features graph constructed over the relevant corpus, and ran the train and test procedures over 10 instances of each train:test ratio.

During graph propagation GraphNER loads the entire graph into memory, which represents the peak memory usage for the algorithm. Thus, the memory footprint of GraphNER can be estimated by the size of the graph description files. This is about 90 MB and 105 MB for the all-feature graphs constructed over AML and BC2GM corpora, respectively.

### D. Statistics of all-feature graphs

In the all-feature graphs, which led to the best results for both corpora, we note that the number of vertices (406,179 for BC2GM, and 348,683 for AML) are comparable. The percentage of labelled vertices is high in both graphs (77.2% for BC2GM, and 51.7% for AML). This is due to the fact that we are experimenting in a transductive setting, where the only unlabelled data is the test data. However, the percentage of labelled vertices, and especially the percentage of positively labelled vertices (vertices that appeared as beginning or inside of a gene in the train set) show marked differences, though they are quite low in both graphs (8.5% in BC2GM, and 1.75% for AML). The low percentage of positively labelled vertices in both graphs explains the higher precision of GraphNER.

Both graphs are weakly connected, and by construction

TABLE V
NULL HYPOTHESES TESTED USING SIGF [24] AND THE CORRESPONDING P-VALUES.

| null hypothesis | p-value |
|---|---|
| BANNER and GraphNER with BANNER has the same F-score on BC2GM corpus | $< 10^{-4}$ |
| BANNER_ChemDNER and GraphNER with BANNER_ChemDNER has the same F-score on BC2GM corpus | $< 10^{-4}$ |
| BANNER and GraphNER with BANNER has the same F-score on AML corpus | 0.018 |
| BANNER and GraphNER with BANNER has the same Recall on AML corpus | 0.72 |
| BANNER and GraphNER with BANNER has the same Precision on AML corpus | 0.0003 |
| BANNER_ChemDNER and GraphNER with BANNER_ChemDNER has the same F-score on AML corpus | 0.035 |
| BANNER_ChemDNER and GraphNER with BANNER_ChemDNER has the same Recall on AML corpus | 0.74 |
| BANNER_ChemDNER and GraphNER with BANNER_ChemDNER has the same Precision on AML corpus | 0.003 |

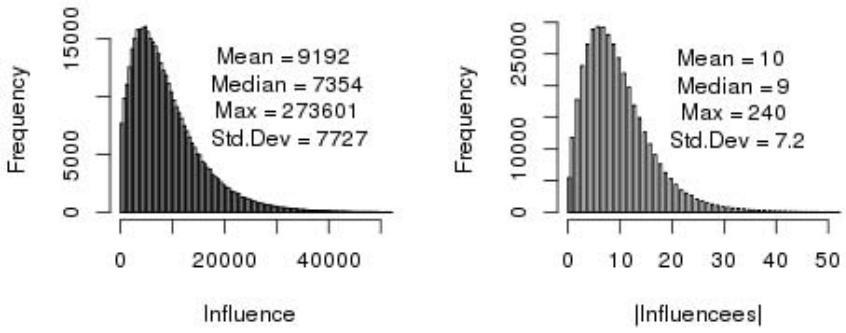


Fig. 3. Histogram of influence and number of influencees for BC2GM all-features graphs.

(because $K = 10$), the outgoing degree is 10 for all vertices. It follows that the number of edges are exactly 10 times the number of vertices in both graphs. The fact that any given vertex has exactly 10 nearest neighbors, however, does not mean that each vertex is the nearest neighbor to exactly 10 vertices.

The nearest neighbors of a vertex influence its label distribution in graph propagation. To formalize this, we can define Influencees$(v)$ as the set of vertices that $v$ influences, that is the set of vertices to which $v$ is a nearest neighbor. Then we can define Influence$(v)$ to be the sum of edge weights that connect $v$ to Influencees$(v)$:

$$\text{Influence}(v) = \sum_{k \in \text{Influencees}(v)} w_{k,v}$$

Using these definitions, we can use $|\text{Influnecees}(v)|$ and Influence$(v)$ as measures of a vertex's influence. Figure 3 shows the histogram of these measures over all vertices in the all-features BC2GM graph. As expected, the plots indicate that most vertices have low influences. We obtained similar histograms for AML all-features graph.

### *E. Qualitative performance differences*

GraphNER consistently improved precision of the purely supervised CRF models, both BANNER and BANNER-ChemDNER, when trained on either the BC2GM corpus or AML corpus. To evaluate this outcome qualitatively, we performed a manual review of the false positive and false negative calls when using the corpus as the gold standard. To reduce the time of this task, we randomly sampled 280 errors from the 5000 BC2GM corpus errors. Due to the small number of total errors in the AML corpus, we reviewed all of the 454 AML corpus errors.

We categorized each false positive or false negative entity into one of two categories, either gene-related or spurious. Gene-related entities included actual genes, gene families, or specific protein domains. For example, "E3 ubiquitin", a gene family, was a gene-related false positive in BANNER that was corrected by GraphNER. Spurious entities were entirely erroneous annotations that did not thematically relate to genes or proteins. For example, "Ann Arbor" was a spurious false positive in BANNER that was also corrected by GraphNER.

Figure 4 shows an UpSet plot [18] of the intersections of false positive calls in GraphNER versus the BANNER-ChemDNER in the AML corpus. UpSet plots visualize combinatorial set intersections with a bar plot. The intersecting set for each bar is represented by the ball and stick model on the x axis. A chi-square two-sample test for equality of proportions with continuity correction found no significant difference in the relative proportion of false positives in gene-related entities in the supervised CRF model and the semi-supervised GraphNER model when trained and tested with the AML corpus (p=0.56). The difference in AML corpus precision noted in Table II was due to a quantitative difference in total annotations, rather than a difference in quality of annotations. Conversely, a chi-square test of relative proportion of gene-related entities was significant when both tools were trained and tested on the BC2GM corpus (p=0.029). Figure 5 shows substantial quantitative and proportional decreases in the number of spurious false positive calls when using GraphNER compared to BANNER-ChemDNER. GraphNER corrected several spurious annotations from the supervised BANNER and BANNER-ChemDNER CRF models, resulting in proportionally fewer spurious entities when trained on the BC2GM data.

In comparison to the AML corpus training results, a significantly higher proportion of false positive and false negative annotations from the BC2GM corpus training were in fact caused by a higher proportion of incorrect annotations in the gold standard corpus. For example, GraphNER correctly tagged "GRK6" as a gene, but our testing protocol counted this as a false positive due to the lack of an annotation in the BC2GM gold standard. The discrepancy in proportion of false corpus annotations between the AML and BC2GM corpora was highly significant based on a chi-square test of

proportions ($p < 2.2 \times 10^{-16}$). These results support the conclusion that GraphNER is more robust to training on sets with a higher rate of annotator errors, and GraphNER corrected spurious BC2GM annotations in BANNER and BANNER-ChemDNER. This advantage was less apparent when training on the AML corpus, which had fewer actual errors in its ground truth annotations.

## IV. Conclusion

We presented a new method, called GraphNER, for the semi-supervised learning of named entities, specifically, gene mentions in biomedical literature. Our tool uses a novel combination of conditional random fields (CRFs) for structured prediction of gene mentions, and graph propagation to combine labeled (for supervised learning) and unlabeled data (which, combined with the labeled data, leads to our semi-supervised learning approach). BANNER and BANNER-ChemDNER are the state of the art CRF models for supervised gene mention detection. We show that our semi-supervised learning approach improves upon their results. We benchmark this approach on two different biomedical text corpora: annotated abstracts of the BioCreative II gene mention shared task corpus, and annotated full text articles related to acute myeloid leukemia.

GraphNER outperformed the baselines regardless of the features used from the CRF models in the graph construction phase of our approach. The higher F-score produced by GraphNER was consistently (in many experiments) due to higher precision over the base model. This is expected given the low percentage of positively labelled vertices.

We used GraphNER in a transductive setting where all the unlabelled data came from the test set (we consider the test set unlabeled since we do not know the true labels until we do the evaluation). Given the semi-supervised nature of GraphNER, we expect even higher performance when the tool is provided abundant unlabelled data. Though this would present an algorithmic challenge, as the time complexity of graph construction grows rapidly with increased dataset size, and becomes prohibitive for resources as large as the complete PubMed database. However, once the graph is constructed, we have shown that the time cost of semi-supervised learning with GraphNER is low in comparison to the CRF purely supervised model.

We expect that the increased accuracy of the GraphNER algorithm and the fact that it can adapt to new types of gene mentions that are not in the labeled training set will lead to more successful automated knowledge discovery from the massive quantities of published papers that overwhelm biomedical researchers today.

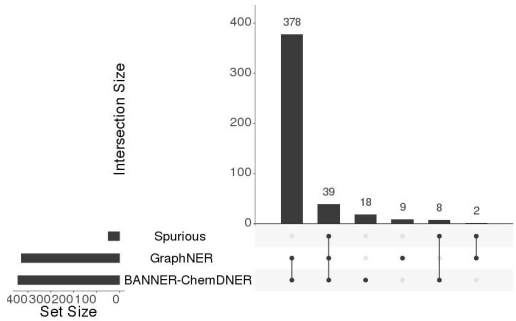


Fig. 4. Upset plot of qualitative false positive differences between GraphNER and BANNER-ChemDNER trained on the AML corpus.

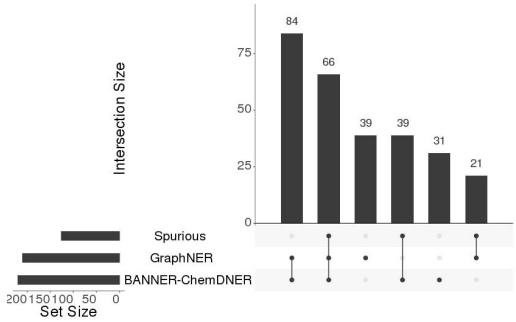


Fig. 5. Upset plot of qualitative false positive differences between GraphNER and BANNER-ChemDNER trained on the BC2GM corpus.

## References


[1] A. Alexandrescu and K. Kirchhoff, "Graph-based learning for statistical machine translation," in *NAACL 2009*, 2009.

[2] R. K. Ando, "BioCreative II gene mention tagging system at IBM Watson," in *Proceedings of the Second BioCreative Challenge Evaluation Workshop*, vol. 23. Centro Nacional de Investigaciones Oncologicas (CNIO) Madrid, Spain, 2007, pp. 101–103.

[3] P. F. Brown, P. V. Desouza, R. L. Mercer, V. J. D. Pietra, and J. C. Lai, "Class-based n-gram models of natural language," *Computational linguistics*, vol. 18, no. 4, pp. 467–479, 1992.

[4] D. Campos, S. Matos, and J. L. Oliveira, "Gimli: open source and high-performance biomedical name recognition," *BMC bioinformatics*, vol. 14, no. 1, p. 54, 2013.

[5] J. P. Chiu and E. Nichols, "Named entity recognition with bidirectional lstm-cnns," in *TACL 2016.*, 2016.

[6] H.-J. Dai, P.-T. Lai, Y.-C. Chang, and R. T.-H. Tsai, "Enhancing of chemical compound and drug name recognition using representative tag scheme and fine-grained tokenization," *Journal of cheminformatics*, vol. 7, no. S1, pp. 1–10, 2015.

[7] D. Das and S. Petrov, "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*. Association for Computational Linguistics, 2011, pp. 600–609.

[8] G. H. Gonzalez, T. Tahsin, B. C. Goodale, A. C. Greene, and C. S. Greene, "Recent advances and emerging applications in text and data mining for biomedical discovery," *Briefings in bioinformatics*, vol. 17, no. 1, pp. 33–42, 2016.

[9] K. Hakala, S. Van Landeghem, T. Salakoski, Y. Van de Peer, and F. Ginter, "Application of the evex resource to event extraction and network construction: Shared task entry and result analysis," *BMC bioinformatics*, vol. 16, no. Suppl 16, p. S3, 2015.

[10] S. J. Hebbring, M. Rastegar-Mojarad, Z. Ye, J. Mayer, C. Jacobson, and S. Lin, "Application of clinical text data for phenome-wide association studies (PheWASs)," *Bioinformatics*, vol. 31, no. 12, pp. 1981–1987, 2015.

[11] M. Krallinger, O. Rabal, F. Leitner, M. Vazquez, D. Salgado, Z. Lu, R. Leaman, Y. Lu, D. Ji, D. M. Lowe *et al.*, "The chemdner corpus of chemicals and drugs and its annotation principles," *Journal of cheminformatics*, vol. 7, no. S1, pp. 1–17, 2015.

[12] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *ICML*, 2001.

[13] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, June 2016, pp. 260–270. [Online]. Available: http://www.aclweb.org/anthology/N16-1030

[14] R. Leaman, R. I. Doğan, and Z. Lu, "Dnorm: disease name normalization with pairwise learning to rank," *Bioinformatics*, vol. 29, no. 22, pp. 2909–2917, 2013.

[15] R. Leaman, G. Gonzalez *et al.*, "Banner: an executable survey of advances in biomedical named entity recognition." in *Pacific Symposium on Biocomputing*, vol. 13. Citeseer, 2008, pp. 652–663.

[16] R. Leaman, C.-H. Wei, and Z. Lu, "tmchem: a high performance approach for chemical named entity recognition and normalization." *J. Cheminformatics*, vol. 7, no. S-1, p. S3, 2015.

[17] H.-J. Lee, T. C. Dang, H. Lee, and J. C. Park, "Oncosearch: cancer gene search engine with literature evidence," *Nucleic acids research*, p. gku368, 2014.

[18] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister, "Upset: visualization of intersecting sets," *IEEE transactions on visualization and computer graphics*, vol. 20, no. 12, pp. 1983–1992, 2014.

[19] G. Li, K. E. Ross, C. N. Arighi, Y. Peng, C. H. Wu, and K. Vijay-Shanker, "mirtex: A text mining system for mirna-gene relation extraction," *PLoS Comput Biol*, vol. 11, no. 9, p. e1004391, 2015.

[20] S. Liu, C.-H. Li, M. Li, and M. Zhou, "Learning translation consensus with structured label propagation," in *ACL 2012*, 2012.

[21] Y. Luo, Ö. Uzuner, and P. Szolovits, "Bridging semantics and syntax with graph algorithmsstate-of-the-art of extracting biomedical relations," *Briefings in bioinformatics*, p. bbw001, 2016.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] T. Munkhdalai, M. Li, K. Batsuren, H. Park, N. Choi, and K. H. Ryu, "Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations." *J. Cheminformatics*, vol. 7, no. S-1, p. S9, 2015.

[24] S. Padó, *User's guide to* `sigf`*: Significance testing by approximate randomisation*, 2006.

[25] S. Pyysalo, T. Ohta, R. Rak, A. Rowley, H.-W. Chun, S.-J. Jung, S.-P. Choi, J. Tsujii, and S. Ananiadou, "Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013," *BMC bioinformatics*, vol. 16, no. Suppl 10, p. S2, 2015.

[26] M. Rei, G. K. Crichton, and S. Pyysalo, "Attending to characters in neural sequence labeling models," *arXiv preprint arXiv:1611.04361*, 2016.

[27] A. Saluja, H. Hassan, K. Toutanova, and C. Quirk, "Graph-based semi-supervised learning of translation models from monolingual data," in *ACL 2014*, 2014.

[28] G. Sheikhshab, E. Starks, A. Karsan, A. Sarkar, and I. Birol, "Graph-based semi-supervised gene mention tagging," in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing*, 2016, pp. 27–35.

[29] A. Subramanya and J. A. Bilmes, "Entropic graph regularization in non-parametric semi-supervised classification," in *Advances in Neural Information Processing Systems*, 2009, pp. 1803–1811.

[30] A. Subramanya, S. Petrov, and F. Pereira, "Efficient graph-based semi-supervised learning of structured tagging models," in *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2010, pp. 167–176.

[31] P. P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira, "Weakly-supervised acquisition of labeled class instances using graph random walks," in *EMNLP 2008*, 2008.

[32] A. Tamura, T. Watanabe, and E. Sumita, "Bilingual lexicon extraction from comparable corpora using label propagation," in *EMNLP-CoNLL 2012*, 2012.

[33] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, 1967.

[34] A. Yeh, "More accurate tests for the statistical significance of result differences," in *Proceedings of the 18th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, 2000, pp. 947–953.

[35] X. Zhu, Z. Ghahramani, J. Lafferty *et al.*, "Semi-supervised learning using gaussian fields and harmonic functions," in *ICML*, vol. 3, 2003, pp. 912–919.