Bioinformatics, 32(18), 2016, 2839–2846
doi: 10.1093/bioinformatics/btw343
Advance Access Publication Date: 9 June 2016
Original Paper

Data and text mining

TaggerOne: joint named entity recognition and normalization with semi-Markov Models

Robert Leaman and Zhiyong Lu*

National Center for Biotechnology Information, 8600 Rockville Pike, Bethesda, MD 20894, USA

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on February 16, 2016; revised on May 2, 2016; accepted on May 26, 2016

Abstract

Motivation: Text mining is increasingly used to manage the accelerating pace of the biomedical literature. Many text mining applications depend on accurate named entity recognition (NER) and normalization (grounding). While high performing machine learning methods trainable for many entity types exist for NER, normalization methods are usually specialized to a single entity type. NER and normalization systems are also typically used in a serial pipeline, causing cascading errors and limiting the ability of the NER system to directly exploit the lexical information provided by the normalization.

Methods: We propose the first machine learning model for joint NER and normalization during both training and prediction. The model is trainable for arbitrary entity types and consists of a semi-Markov structured linear classifier, with a rich feature approach for NER and supervised semantic indexing for normalization. We also introduce TaggerOne, a Java implementation of our model as a general toolkit for joint NER and normalization. TaggerOne is not specific to any entity type, requiring only annotated training data and a corresponding lexicon, and has been optimized for high throughput.

Results: We validated TaggerOne with multiple gold-standard corpora containing both mention- and concept-level annotations. Benchmarking results show that TaggerOne achieves high performance on diseases (NCBI Disease corpus, NER f-score: 0.829, normalization f-score: 0.807) and chemicals (BioCreative 5 CDR corpus, NER f-score: 0.914, normalization f-score 0.895). These results compare favorably to the previous state of the art, notwithstanding the greater flexibility of the model. We conclude that jointly modeling NER and normalization greatly improves performance.

Availability and Implementation: The TaggerOne source code and an online demonstration are available at: http://www.ncbi.nlm.nih.gov/bionlp/taggerone

Contact: zhiyong.lu@nih.gov

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Many tasks in biomedical information extraction rely on accurate named entity recognition (NER), the identification of text spans mentioning a concept of a specific class, such as disease or chemical. Recent research has demonstrated that a particular NER approach—namely, conditional random fields with a rich feature set—consistently achieves high performance on a variety of NER tasks when provided with an appropriate training corpus and a relatively small investment in feature engineering. This approach has been used to identify a wide variety of entities, including genes and proteins (Leaman and Gonzalez, 2008; Wei *et al.*, 2015a), diseases (Chowdhury and Lavelli, 2010; Leaman *et al.*, 2013), chemicals. (Leaman *et al.*, 2015b; Rocktaschel *et al.*, 2012) and anatomic entities (Pyysalo and Ananiadou, 2014). However many end-user tasks

also require normalization (grounding), the identification of the concept mentioned within a controlled vocabulary or ontology, making the utility of NER on its own relatively low.

We recently demonstrated DNorm, the first machine learning based method for disease normalization (Leaman *et al.*, 2013). This method used supervised semantic indexing (Bai *et al.*, 2010), trained with pairwise learning to rank, to score the mentions returned by a conditional random field NER system, BANNER (Leaman and Gonzalez, 2008), against the disease names from a controlled vocabulary. The method focuses primarily on semantic term variation, such as when an author refers to the concept 'renal insufficiency' with the phrase 'decreased renal function.' Our experiments demonstrated the method to be highly effective for disease normalization.

Like many normalization systems, however, DNorm uses a pipeline architecture: the tasks of NER and normalization are performed serially, making errors cascading from one component to the next a common problem. Our error analysis of DNorm, for example, demonstrated that over half of the overall system errors were caused by NER errors that the normalization component could not recover.

One way to overcome cascading errors is to perform NER and normalization simultaneously. Dictionary systems do this by directly matching text to the names in a controlled vocabulary. Unfortunately, NER systems employing machine learning typically have higher performance. To the best of our knowledge, a machine learning method that trains a joint model of NER and normalization has not been previously proposed.

In this work, we propose a model that simultaneously performs NER and normalization—focusing on term variation—during both training and prediction. We evaluate our model on two corpora containing both mention and concept annotations; one contains disease entities, the other contains both disease and chemical entities. Figure 1 provides an example text with both disease and chemical annotations. We achieve state-of-the-art performance on both diseases and chemicals.

1.1 Related work

Named entity recognition (NER) and normalization have long been recognized as important tasks within biomedical text mining. Both tasks have been the subject of community challenges (Hirschman *et al.*, 2005; Kim *et al.*, 2009; Krallinger *et al.*, 2015a,b; Morgan *et al.*, 2008).

The development of NER and normalization systems for diseases lagged behind genes and proteins for some time, primarily due to the lack of annotated corpora. Jimeno *et al.* (2008) created a corpus of sentences that was expanded by Leaman *et al.* (2009); this was further expanded to become the NCBI Disease Corpus (Doğan *et al.*, 2014). Diseases were also included in the set of entities annotated in the CALBC silver standard corpus (Rebholz-Schuhmann *et al.*, 2010). Several rule or dictionary based systems have used these disease corpora for evaluation of NER (Campos *et al.*, 2013; Song *et al.*, 2015) or normalization (Kang *et al.*, 2012). Our previous work DNorm demonstrated significantly higher normalization performance when using a machine learning model (supervised semantic indexing) trained with pairwise learning to rank (Leaman *et al.*, 2013). Most recently, the Chemical Disease Relation task at the BioCreative V community challenge included disease normalization as a subtask (Li *et al.*, 2015; Wei *et al.*, 2015a,c).

The development of chemical NER and normalization systems was initially enabled by rigorous standards for the chemical nomenclature. The OSCAR system normalizes many varieties of chemical mentions, and is intended for mining chemistry publications (Jessop *et al.*, 2011).

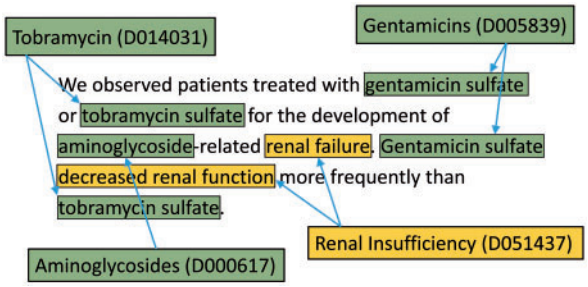

Fig. 1. Example text with chemical and disease entity annotations, adapted from PMID 7420681. The outer boxes specify the annotated term and MeSH identifier

Kolarik *et al.* (2008) created the SCAI corpus of chemical mentions, Klinger *et al.* (2008) used this to train and evaluate a machine learning approach for chemical NER. Rocktaschel *et al.* (2012) expanded the machine learning approach with extensive lexical resources. Chemicals were also included in the CALBC silver standard corpus (Rebholz-Schuhmann *et al.*, 2010). The CHEMDNER task at BioCreative IV addressed chemical NER, releasing a large corpus of chemical mentions in PubMed abstracts (Krallinger *et al.*, 2015a), where our submission tmChem achieved the highest performance out of 27 teams (Leaman *et al.*, 2015b). The CHEMDNER task at BioCreative V also addressed chemical NER, but changed the domain to patents (Krallinger *et al.*, 2015b). Two recent surveys of the field are Vazquez *et al.* (2011) and Eltyeb and Salim (2014).

Our method builds successfully on previous work in NER and normalization. Cohen and Sarawagi (2004) were the first to apply semi-Markov models to NER, motivated by a need to integrate soft-match dictionary features. Okanohara *et al.* (2006) later applied semi-Markov models to the biomedical domain. Tsuruoka *et al.* (2007) is a method for learning term variation, trained directly from a lexicon using similarity measures as features. DNorm instead learned the similarity between individual tokens directly from training data (Leaman *et al.*, 2013). The advantage of joint learning has been demonstrated for many tasks. For example, Finkel and Manning (2009) learned a joint model for parsing and NER in newswire text, while Durrett and Klein (2014) learned a model for joint coreference resolution, named entity classification and entity linking (disambiguation) when the named entity spans were provided as input. Recently, Le *et al.* (2015) proposed a model that performs joint NER and normalization for diseases in biomedical text during prediction, but not during training. Our system is the first, to our knowledge, that performs joint NER and normalization during both training and prediction. In addition, our system is open source, trainable for arbitrary entity types and optimized for high throughput.

2 Methods

In this section we describe our model for joint NER and normalization. We describe the preprocessing steps used and the lexicons employed. We detail our joint model, describing the features used, how it is trained and used for prediction. We also describe the disambiguation steps performed. An overview of the TaggerOne system is provided in Figure 2. Finally, we describe the state-of-the-art open source systems used for comparison.

2.1 Preprocessing

We use Ab3P to identify abbreviations within each document (Sohn *et al.*, 2008), and then replace each instance of the short form (e.g.

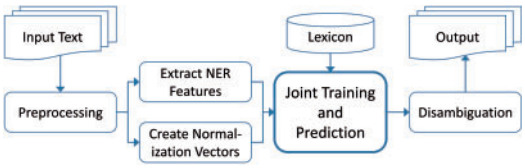

Fig. 2. Overview of the TaggerOne system. Joint modeling of NER and normalization is performed by scoring all text segments against each NER class and name in the lexicon

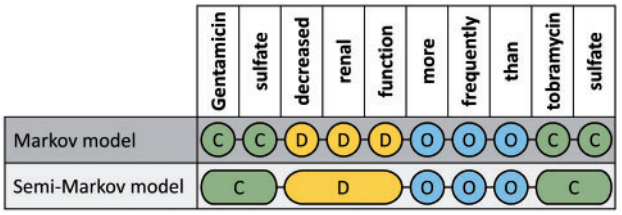

Fig. 3. Example text labeled with both Markov and semi-Markov models. C labels refer to chemicals, D labels to diseases and O labels to non-entities. Markov models assign labels to individual tokens; semi-Markov models separate the input into segments of one or more tokens and assign a label to each

'CT') with the corresponding long form ('copper toxicosis'). We use SimConcept to identify composite mentions (e.g. 'cleft lip/palate') and resolve them into their component parts ('cleft lip' and 'cleft palate') (Wei *et al.*, 2015a). We also segment text into sentences. We use two tokenization approaches. For diseases, we segment tokens at whitespace and separate punctuation characters into individual tokens. For chemicals, we also separate tokens at letter/digit boundaries and lowercase to uppercase boundaries. When jointly modeling chemicals and diseases, we use the same strategy as for chemicals.

2.2 Joint modeling of NER and normalization

NER is often handled as a sequence labeling problem and frequently addressed with Markov models. These models derive their name from the Markov property, which asserts that the current label in the output is independent of all other labels except the one preceding. Markov models assign a label to each token in the input sequence; an example text is shown in Figure 3.

In this work, we approach joint NER and normalization using semi-Markov models. These models assign labels to contiguous subsequences (segments) of variable length, as shown in Figure 3. Like Markov models, semi-Markov models obey the Markov property between transitions, but—unlike Markov models—do not require a transition for each token. Because segmentation is part of the model, semi-Markov models enable features that integrate information across all tokens in the segment. We exploit this ability to simultaneously learn a normalization scoring function, enabling the creation of a practical model for joint NER and normalization.

2.2.1 Problem statement

After preprocessing, our input consists of a sequence of tokens. The objective of our model is to divide this sequence into segments, each consisting of one or more tokens and assign a class to each. Since we are performing NER and normalization simultaneously, the class must indicate both the NER and normalization. Each segment must therefore specify the NER label (such as *Disease*) and both the name and entity mentioned by the text.

We extend the formal problem statement of Cohen and Sarawagi (2004) describing semi-Markov models for NER to our task of joint NER and normalization. Specifically, let $X = (X_1, \ldots, X_{|X|})$ represent an input text as a sequence of tokens. Let $\mathcal{L}$ be the set of NER labels (including a special non-entity label, *Other*). Let $\mathcal{N}_\ell$ and $\mathcal{E}_\ell$ be respectively the set of names and entities in the lexicon for label $\ell \in \mathcal{L}$. Let $entity : \mathcal{N} \rightarrow \mathcal{E}$ be the mapping defined by the lexicon from names to entities, which we assume to associate each name $n \in \mathcal{N}_\ell$ with exactly one entity $e \in \mathcal{E}_\ell$. Let $Y = \langle Y_1, \ldots, Y_{|Y|} \rangle$ be a *segmentation* of X. Each segment $Y_j \in Y$ is a 5-tuple consisting of:

1. The index of the first token in the segment, a_j: $0 \leq a_j \leq |X|$
2. The (exclusive) index of the last token, b_j: $a_j < b_j \leq |X| + 1$
3. The NER label, $\ell_j \in \mathcal{L}$
4. The lexicon name, $n_j \in \mathcal{N}_{\ell_j}$; if $\ell_j = Other$ then $n_j = \varnothing$
5. The entity, $e_j = entity(n_j)$; if $\ell_j = Other$ then $e_j = \varnothing$

Note that segmentations which have the same NER information (segment indices and NER labels) but differ in any of the normalization information (lexicon name or entity) are not equivalent.

A segmentation is valid if all tokens from X are used exactly once, in order, and if the length of all segments with label *Other* is exactly one token. Let $\mathcal{Y}(X)$ be the set of all valid segmentations of X. According to our definitions, the segmentation for the example text shown in Figure 3 would be:

$$Y = (0,\ 2,\ \text{Chemical},\ \text{"Gentamicins"},\ D005839),$$
$$(2,\ 5,\ \text{Disease},\ \text{"Renal Insufficiency"},\ D051437),$$
$$(5, 6, \text{Other}, \varnothing, \varnothing), (6, 7, \text{Other}, \varnothing, \varnothing),\ (7, 8, \text{Other}, \varnothing, \varnothing),$$
$$(8,\ 10,\ \text{Chemical},\ \text{"Tobramycin"},\ \text{D014031})$$

2.2.2 Model description

We define a scoring function over the set of valid segmentations $\mathcal{Y}(X)$, so that the task of prediction becomes finding the segmentation $Y \in \mathcal{Y}(X)$ with the highest score:

$$f(X) = \underset{Y \in \mathcal{Y}(X)}{\operatorname{argmax}}\, score(Y; s, t, W)$$

where s, t and W are the parameter weights of the model, to be defined. We define the score for a segmentation Y as the sum of the scores for each segment:

$$score(Y; s, t, W) = \sum_{j=0}^{|Y|} score(Y_j; s, t, W)$$

Under this formulation, the highest-scoring segmentation can be found efficiently using a modification of the Viterbi algorithm (fully described in the supplemental material). We perform NER and normalization simultaneously by defining the score for each segment Y_j to be the sum of its NER and normalization scores:

$$score(Y_j; s, t, W) = score_{NER}(Y_j; s) + score_{Norm}(Y_j; t, W)$$

We model the NER scoring function as a structured classification problem using a multi-class linear classifier, similar to previous work using structured perceptrons or support vector machines (Altun *et al.*, 2007; Crammer and Singer, 2001; Taskar *et al.*, 2004) with a rich feature approach. This approach learns one weight vector per label $\ell_j \in \mathcal{L}$, constrained so that the correct label for any given segment will

be the one with the highest score. Our rich feature approach for preparing the NER feature vectors is detailed in Section 2.2.3. If we let r_j be the NER feature vector for segment Y_j and let s_{ℓ_j} be the NER weight vector for ℓ_j, then the NER score for Y_j is their dot product:

$$score_{NER}(Y_j; s) = r_j^T s_{\ell_j}$$

Normalization is more difficult, however, due to the significantly greater number of categories (one per name $n_j \in \mathcal{N}_{\ell_j}$). We use a supervised semantic indexing approach (Bai *et al.*, 2010; Leaman *et al.*, 2013), which converts both the segments Y_j and names $n_j \in \mathcal{N}_{\ell_j}$ into vectors and then uses a weight matrix (W_ℓ) to score pairs of vectors. We describe the creation of the normalization vectors in Section 2.2.4. In this work we introduce an additional term for the cosine similarity, t_ℓ. If we let the normalization vector for Y_j be u_j and the normalization vector for name n_j be v_j, then the normalization score for Y_j is:

$$score_{Norm}(Y_j; t, W) = t_{\ell_j}\left(u_j^T v_j\right) + u_j^T W_{\ell_j} v_j$$

Element $\langle i, j \rangle$ in matrix W_ℓ can be interpreted as the correlation between token t_i appearing in a text segment with NER label ℓ and token t_j appearing in any concept name for ℓ from the lexicon. The model can thus learn a variety of relationships between tokens in text and names from the lexicon, including both synonymy and contrast. While the diagonal elements of W_ℓ already model the same values represented by the cosine similarity parameter t_ℓ, it represents the similarity between *any* token appearing in a text segment with NER label ℓ and the *same* token in the lexicon. The term can therefore be considered a 'base value' for all of the diagonal elements; it is also the only trained normalization parameter used for tokens not seen during training.

We could also add an element to the scoring function that models the dependency of the current label (ℓ_j) on the previous label (ℓ_{j-1}), as specified by the Markov property. The number of previous labels included (the *order*) can also be varied; order 1 and order 2 are common choices. We found, however, that conditioning the classification on any number of previous labels reduced performance. We use a scoring function that is independent of all other labels, making our model an order 0 semi-Markov model.

2.2.3 NER features

The NER features are prepared using a rich feature approach, with feature templates defined for either individual tokens or segments as needed. Token-level feature templates are similar to previous work in biomedical NER (Leaman and Gonzalez, 2008; Leaman *et al.*, 2015a), including:

- Token text, token stem
- Part of speech
- Character 2, 3 and 4—grams
- Patterns to recognize numbers and common variations in capitalization

Feature templates defined at the segment level include:

- The number of tokens in the segment
- The characters and tokens surrounding the segment
- The first and last token in the segment
- Whether the segment contains unbalanced parenthesis
- Patterns to recognize common representations of Greek letters, partial chemical formulas and amino acids.
- Whether the start or end token is a member of a closed class in English

The NER feature vector for each segment is equal to the segment level feature values summed with each of the token level features for each token within the segment.

2.2.4 Normalization vector space

The normalization vector space is prepared similar to our previous work with the tokens from the lexicon (Leaman *et al.*, 2013), but now also contains all tokens in the training data. To create the set of tokens within the space, we process the names in the lexicon and all segments in the training data as follows:

- Conversion to lower case.
- Punctuation removal.
- Stop word removal; we use the same set of stop words as DNorm (Leaman *et al.*, 2013).
- Stemming: diseases use the Porter stemmer (Porter, 1980) while chemicals only remove plurals (Hartman, 1991).

We then define a corresponding vector space and create vectors within that space for each segment in the input data and each name in the lexicon. We use tf-idf weighting, modified so that the set of documents used for the idf calculation is the set of names in the lexicon. Tokens not present in the vector space (i.e. present in the evaluation set but not the training set) are represented as a unique 'unknown' token so that normalization scores reflect the reduced quality of the match.

All normalization vectors are scaled to unit length, making the normalization score independent of the number of tokens in the text segment or lexicon name. This scaling requires information to be integrated across the text segment, and is therefore enabled by our use of semi-Markov models.

2.3 Training

We train our model using the margin-infused relaxed algorithm (MIRA) (Crammer and Singer, 2003). Similar to the perceptron, MIRA is an online algorithm that performs no update if the instance is already correctly classified. Unlike the perceptron, the update does not use a fixed step size. Instead, MIRA determines the minimum change to the weights that would score the (correct) annotated segmentation higher than the (incorrect) segmentation currently given the highest score by the model by at least as much as the loss.

If we use s', x' and W' to respectively describe s, x and W after the update, then the size of the update (α) is the length of the difference of all weights in Euclidean space:

$$\alpha = \sqrt{\|s' - s\|^2 + (t' - t)^2 + \|W' - W_\ell\|^2}$$

The goal of the MIRA update is to find the smallest update, subject to the constraint of correctly classifying the instance after the update:

$$update = \underset{s', x', W'}{\operatorname{argmin}} \alpha + c \sum_n \xi_k$$

where ξ_k are slack variables ($\xi_k \geq 0$) to ensure separability, the c parameter controls the size of the updates, and n is the number of constraints. We use the hinge loss and constrain the update so that the score for the annotated segmentation (Y^+) will be higher than the score for the segmentation that currently has the highest score ($Y^- = f(X)$) by at least as much as the loss:

$$score(Y^+; s', t', W') - score(Y^-; s', t', W') + \xi_0$$

$$\geq max(0, score(Y^-; s, t, W) - score(Y^+; s, t, W))$$

We found it useful to also add constraints focusing on the normalization. For each segment Y_j^+ in the annotated segmentation whose label is not *Other*, we add a constraint that the normalization with the highest score for that segment should be the one annotated:

$$score_{Norm}\left(Y_j^+; t', W'\right) - score_{Norm}\left(Y_j^-; t', W'\right) + \xi_k$$

$$\geq \max(0, score_{NEN}(Y^-; t, W) - score_{Norm}(Y^+; t, W))$$

When the entity for the annotated segment Y_j^+ has multiple synonyms, we let the model determine which name should be used by selecting the name with the highest score according to the current model weights.

Determining the smallest update that satisfies the constraints is a numerical optimization problem, specifically a quadratic program. While it has an exact solution, it contains more than one constraint and therefore must be solved numerically. We use an open source numerical optimizer (ojAlgo: http://ojalgo.org) to solve for the update.

To keep a single instance from making large changes to the weights, we limit the change (λ) to be at most m: $\lambda = min(m, \ldots)$. We empirically determine the value of c and m by performing a grid search using a randomly selected subset of the training data (100 documents).

We iterate through all training instances in random order on each iteration. All weights are initialized to 0 at the start of training. To reduce overtraining, we use model averaging and also evaluate the performance on a holdout set after each training iteration. We use the harmonic mean of the NER and normalization f-scores (as described in Section 3) as the holdout performance measure. We output the current model if performance has improved over the previous iteration, and stop training when $n = 10$ iterations have elapsed without a performance improvement. We then consider the last model output as the final model.

2.4 Disambiguation

Though our primarily normalization focus is term variation, if the highest-scoring name vector is the name for two or more entities then we perform two steps to disambiguate. First, if the name is marked as a synonym for one entity and the primary name for the parent of that entity, we prefer the parent. Second, we prefer the entities that appear more frequently in the training data.

2.5 Lexical resources

In this work, the goal is to perform NER and normalization by learning a mapping to a specific lexicon, rather than maximizing performance by expanding the lexicon. We therefore exclusively use the disease and chemical vocabularies distributed by the Comparative Toxicogenomics Database project (CTD, http://ctdbase.org). The CTD vocabulary for diseases, MEDIC, is derived from a combination of OMIM (http://www.omim.org) and the disease branch of MeSH (https://www.nlm.nih.gov/mesh) and lists 11 885 disease entities and 76 685 names. The CTD chemical vocabulary contains concepts from the MeSH chemical branch. We augmented this vocabulary slightly to ensure it included all chemical element names and symbols up to atomic number 103, resulting in a total of 158 721 chemical entities and 414 246 names.

2.6 Comparison systems

We employ two open source systems with state-of-the-art performance for NER and normalization as comparison benchmarks. We use DNorm (Leaman *et al.*, 2013) for diseases; it has the highest published performance on the NCBI Disease Corpus and also achieved the highest performance in a previous disease challenge task (Leaman *et al.*, 2015a; Pradhan *et al.*, 2015). We use tmChem (Leaman *et al.*, 2015b) for chemicals; it is an ensemble of two chemical NER/normalization systems and achieved the highest performance in the recent CHEMDNER challenge task for chemical NER at BioCreative IV (Krallinger *et al.*, 2015a). In this work we exclusively use Model 1, which is an adaptation of BANNER (Leaman and Gonzalez, 2008) to recognize chemical mentions, combined with a dictionary approach for normalization.

3 Results

We validate TaggerOne by applying it to two corpora containing both mention- and concept-level annotations: the NCBI Disease corpus (Doğan *et al.*, 2014) and the BioCreative V Chemical Disease Relation task corpus (Li *et al.*, 2015). Overall statistics for each dataset are provided in Table 1. The NCBI Disease corpus consists of 793 PubMed abstracts separated into training (593), development (100) and test (100) subsets. The NCBI Disease corpus is annotated with disease mentions, using concept identifiers from either MeSH or OMIM. The BioCreative V Chemical Disease Relation (BC5CDR) corpus consists of 1500 PubMed abstracts, separated into training (1000) and test (500) sets. We created a holdout set by separating the sample set (50 abstracts) from the remainder of the training set. The BC5CDR corpus enables experiments simultaneously modeling multiple entity types; it is annotated with concept identifiers from MeSH for both chemical and disease mentions.

We use two evaluation measures since our model performs both NER and normalization. The NER measure is at the mention level; we require the predicted span and entity type to exactly match the annotated span and entity type. The normalization measure is at the abstract level, comparing the set of concepts predicted for the document to the set annotated, independent of their location within the text. We report both measures in terms of micro-averaged precision, recall and f-score.

We perform two sets of experiments. The first set of experiments evaluates the ability of the model to generalize to unseen text and whether joint NER and normalization improves performance over performing NER separately. This set of experiments models diseases and chemicals separately. The second set of experiments evaluates the ability of the model to simultaneously handle multiple entity types (both diseases and chemicals).

3.1 Results for single-entity models

The results for training and evaluating TaggerOne on a single entity type can be found in Table 2 for NER and Table 3 for normalization. For each corpus, the model was trained on the training set, using the development (or sample) set as a holdout set, and evaluated on the official test set.

The NER f-score is higher for the joint NER+ normalization model than for the NER-only model for all entity types and corpora. Specifically, the error rate for NCBI Disease is reduced by 8%, for BC5CDR (disease) by 15% and for BC5CDR (chemical) by 26%. In all cases the NER f-score is also higher for the joint NER+ normalization model of TaggerOne than for the comparison systems. Finally, we note that the normalization performance has increased

over the comparison systems; specifically the error rate for NCBI Disease is 11% lower, BC5CDR (disease) is 16% lower and BC5CDR (chemical) is 17% lower.

3.2 Results for disease+chemical

The results of training and evaluating TaggerOne on two entity types simultaneously are described in Table 4. For this experiment we trained a single model on the BC5CDR corpus, simultaneously modeling both diseases and chemicals. We note that jointly modeling chemicals and diseases produces the same NER performance and very similar normalization performance.

4 Discussion

The single-entity performance demonstrates both that our model is effective and that jointly modeling NER and normalization improves performance. Our results significantly improve on DNorm for diseases and on tmChem for chemicals. Analyzing the DNorm and TaggerOne results provides insight into the advantage of joint prediction: DNorm often misses phrases that require term variation to be resolved for the phrase to be recognized as an entity, such as 'abnormal involuntary motor movements,' annotated as MeSH identifier D004409: Drug-induced Dyskinesia.

The experiment jointly modeling chemicals and diseases demonstrates that the model maintains high performance while modeling multiple entity types. Modeling multiple entity types simultaneously may be advantageous when the entity types are more difficult to distinguish, such as with anatomical types (Pyysalo and Ananiadou, 2014).

Our results on the NCBI Disease corpus are the highest of which we are aware. The only normalization system with published results on the NCBI Disease corpus besides DNorm is the sieve-based system of D'Souza and Ng (2015). Their evaluation measure calculates the proportion of mentions correctly normalized given perfect NER. Using this measure, their system scored 0.847; TaggerOne scores 0.888.

The recent disease subtask at the BioCreative V chemical disease relation task provides an excellent comparison for our system (Wei *et al.*, 2015c). The UET-CAM system (Le *et al.*, 2015) performs joint NER and normalization for prediction but unlike TaggerOne does not perform joint training; it achieved an f-score of 0.764. The highest performing system at the BC5CDR disease subtask achieved 0.896 precision, 0.835 recall, for 0.865 f-score (Lee *et al.*, 2015). We note that expanding the lexicon was a significant feature in most participating systems; in this manuscript our goal is to automatically learn the best mapping to an existing lexicon. These two approaches are complementary, however. We are not aware of any previous performance evaluations on the chemical entities of the BC5CDR corpus.

We originally trained our model using an averaged perceptron; NER performance was similar but normalization performance was several percent lower (data not shown). We believe this was due to using the same update size for both the NER and normalization weights. Our use of semi-Markov models allows us to scale the normalization vectors for the mentions to unit length. Performance degrades significantly when this scaling is not performed (data not shown).

4.1 Implementation

TaggerOne was implemented in Java as a general toolkit for biomedical NER and normalization. TaggerOne is not specific to any entity type, and is designed to simultaneously handle multiple entity types and lexical resources. The current implementation has an average throughput of 8.5 abstracts per second for diseases, compared to 3.5 for our previous work DNorm (using a single 2.80 Ghz 64-bit Xeon processor limited to 20 Gb memory). The supplemental

Table 1. Statistics for the corpora used for training and evaluation, differentiated by entity type. 'Unique Mentions' and 'Unique Concepts' respectively refer to the number of annotations with unique text or unique identifiers

Corpus and entity type	Annotations	Unique mentions	Unique concepts
NCBI Disease	6892	2135	753
BC5CDR (Disease)	12864	3271	1082
BC5CDR (Chemical)	15933	2630	1274

Table 2. NER results for the NCBI Disease and BC5CDR corpora. DNorm is the benchmark system for disease entities, tmChem model 1 the benchmark system for chemicals. Precision, recall and f-score are respectively abbreviated as P, R and F. The highest value for each is bolded

System	NCBI disease			BC5CDR (disease)			BC5CDR (chemical)		
	P	R	F	P	R	F	P	R	F
Benchmark system	0.822	0.775	0.798	0.820	0.795	0.807	0.932	0.840	0.884
TaggerOne (NER Only)	0.835	0.796	0.815	0.831	0.764	0.796	0.924	0.847	0.884
TaggerOne	**0.851**	**0.808**	**0.829**	**0.852**	**0.802**	**0.826**	**0.942**	**0.888**	**0.914**

Table 3. Normalization results for the NCBI Disease and BC5CDR corpora. DNorm is the benchmark system for disease entities, tmChem model 1 the benchmark system for chemicals. Precision, recall and f-score are respectively abbreviated as P, R and F. The highest value for each is bolded

System	NCBI disease			BC5CDR (disease)			BC5CDR (chemical)		
	P	R	F	P	R	F	P	R	F
Benchmark system	0.803	0.763	0.782	0.812	0.801	0.806	**0.950**	0.808	0.873
TaggerOne	**0.822**	**0.792**	**0.807**	**0.846**	**0.827**	**0.837**	0.888	**0.903**	**0.895**

Table 4. TaggerOne results on the BC5CDR corpus when both disease and chemical mentions are trained within a single model. Precision, recall and f-score are micro-averaged and respectively abbreviated as P, R and F

Entity	NER			Normalization		
	P	R	F	P	R	F
Disease	0.847	0.810	0.828	0.838	0.829	0.833
Chemical	0.938	0.888	0.912	0.879	0.905	0.892

material describes optimizations critical for reducing the considerable computational cost of joint NER and normalization.

4.2 Error analysis and limitations

We manually analyzed a random sample of both corpora for errors and describe the trends observed. False positives and negatives remain a significant source of error. Other entity types—particularly gene names (e.g. 'GAP 43')—are frequently confused with both diseases and chemicals. Diseases are particularly prone to error because of the high similarity to the general biomedical vocabulary (e.g. 'nephrostomy tube'), because individual tokens can change the meaning significantly (e.g. 'coproporphyrinogen oxidase' was identified as the disease 'coproporphyrinogen oxidase deficiency'), and because the model does not identify states considered desirable in context ('analgesia').

Coordination ellipsis and noun compounds also remain a significant source of error. This is an especially difficult problem for chemicals, since it can be difficult to distinguish the number of entities present within a text snippet (e.g. 'copper/zinc superoxide').

We found that our model tends to rely more on the lexicon when the vocabulary is previously unseen. Consistency with the lexicon sometimes comes at the expense of consistency with the annotated data, however. For example, the model identified 'familial renal amyloidosis' though the corpus only contains an annotation for the less specific 'amyloidosis.'

Alternatively, segments are sometimes annotated to include tokens not found in the concept name. For example, the phrase 'isolated unilateral retinoblastoma' was annotated as a whole to 'retinoblastoma.' The model correctly found 'retinoblastoma' and included 'unilateral,' but missed 'isolated.' While primarily an NER issue, these sometimes cause difficulties with normalization (e.g. 'GI toxicity' was normalized to 'gastrointestinal disorder' instead of 'toxicity').

5 Conclusion

We conclude that jointly modeling named entity recognition and normalization results in improved performance for both tasks. Our model is not entity-specific and we expect it to generalize to arbitrary NER and normalization problems in biomedicine. In this work we have demonstrated this capability for both diseases and chemicals. In future work, we intend to integrate a more robust disambiguation method to allow entity types such as genes and proteins to be addressed. We are also interested in investigation its application to the general domain.

While our goal has been to learn the best mapping to an existing lexicon, expanding the lexicon is a complementary approach used by many normalization systems (Wei *et al.*, 2015a,b,c). We anticipate that applying our method to an expanded lexicon would further increase performance (Blair *et al.*, 2014).

An interesting research direction enabled by this work is the possibility of using data not annotated jointly (Finkel and Manning, 2010). Sources of annotations at the document-level are significantly more abundant than annotations at the mention level (Usami *et al.*, 2011). We anticipate our model may enable entity-level distant supervision by providing a joint model of both NER and normalization that handles term variation.

Acknowledgements

We thank the anonymous reviewers for their comments and suggestions.

Funding

This research was supported by the National Institutes of Health Intramural Research Program, National Library of Medicine.

Conflict of Interest: none declared.

References

Altun,Y. *et al.* (2007) Support vector machine learning for independent and structured output spaces. In: Bakir, G. *et al.* (eds) *Predicting Structured Data*. The MIT Press, Cambridge, Massachusetts, USA.

Bai,B. *et al.* (2010) Learning to rank with (a lot of) word features. *Inf. Retrieval*, **13**, 291–314.

Blair,D.R. *et al.* (2014) Quantifying the impact and extent of undocumented biomedical synonymy. *PLoS Comput. Biol.*, **10**, e1003799.

Campos,D. *et al.* (2013) A modular framework for biomedical concept recognition. *BMC Bioinformatics*, **14**, 281.

Chowdhury,F.M. and Lavelli,A. (2010) Disease mention recognition with specific features. *BioNLP Workshop*. Uppsala, Sweden, pp. 83–90.

Cohen,W.W. and Sarawagi,S. (2004) *Exploiting Dictionaries in Named Entity Extraction: Combining Semi-Markov Extractions Processes and Data Integration Methods. 10th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining. ACM*, Seattle, Washington, USA, pp. 89–98.

Crammer,K. and Singer,Y. (2001) On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, **2**, 265–292.

Crammer,K. and Singer,Y. (2003) Ultraconservative online algorithms for multiclass problems. *J. Mach. Learn. Res.*, **3**, 951–991.

D'Souza,J. and Ng,V. (2015) Sieve-Based Entity Linking for the Biomedical Domain. In: *53rd ACL and 7th IJCNLP*. Beijing, China, pp. 297–302.

Doğan,R.I. *et al.* (2014) NCBI disease corpus: A resource for disease name recognition and concept normalization. *J. Biomed. Inf.*, **47**, 1–10.

Durrett,G. and Klein,D. (2014) A joint model for entity analysis: coreference, typing and linking. *Trans. Assoc. Comput. Linguist.*, **2**, 477–490.

Eltyeb,S. and Salim,N. (2014) Chemical named entities recognition: a review on approaches and applications. *J. Cheminf.*, **6**, 17.

Finkel,J.R. and Manning,C.D. (2009) *Joint Parsing and Named Entity Recognition. NAACL/HLT*. Boulder, Colorado, pp. 326–334.

Finkel,J.R. and Manning,C.D. (2010) *Hierarchical Joint Learning: Improving Joint Parsing and Named Entity Recognition with Non-Jointly Labeled Data. 48th ACL*. Uppsala, Sweden, pp. 720–728.

Hartman,D. (1991) How effective is suffixing? *J. Am. Soc. Inf. Sci. Technol.*, **42**, 7–15.

Hirschman,L. *et al.* (2005) Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, **6**, S1.

Jessop,D.M. *et al.* (2011) OSCAR4: a flexible architecture for chemical text-mining. *J. Cheminf.*, **3**, 41.

Jimeno,A. *et al.* (2008) Assessment of disease named entity recognition on a corpus of annotated sentences. *BMC Bioinformatics*, **9**, S3.

Kang,N. *et al.* (2012) Using rule-based natural language processing to improve disease normalization in biomedical text. *J. Am. Med. Inf. Assoc.*, **20**, 876–881.

Kim,J.D. *et al*. (2009) Overview of BioNLP'09 shared task on event extraction. *BioNLP Workshop*, pp. 1–9.

Klinger,R. *et al*. (2008) Detection of IUPAC and IUPAC-like chemical names. *Bioinformatics*, **24**, i268–i276.

Kolarik,C. *et al*. (2008) Chemical names: terminological resources and corpora annotation. *LREC Workshop on Building and Evaluating Resources for Bbiomedical Text Mining*.

Krallinger,M. *et al*. (2015a) CHEMDNER: The drugs and chemical names extraction challenge. *J. Cheminf*., **7**, S1.

Krallinger,M. *et al*. (2015b) *Overview of the CHEMDNER Patents Task. Fifth BioCreative Challenge Evaluation Workshop*. Seville, Spain, pp. 63–75.

Le,H.-Q. *et al*. (2015) *The UET-CAM System in the BioCreAtIvE V CDR Task. BioCreative Workshop*. Seville, Spain, pp. 208–213.

Leaman,R. *et al*. (2013) DNorm: Disease name normalization with pairwise learning-to-rank. *Bioinformatics*, **29**, 2909–2917.

Leaman,R. and Gonzalez,G. (2008) BANNER: an executable survey of advances in biomedical named entity recognition. *Pac. Symp. Biocomput*., 652–663.

Leaman,R. *et al*. (2009) Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. *Proc Symp on Languages in Biology and Medicine*, **13**, pp. 82–89.

Leaman,R. *et al*. (2015a) Challenges in clinical natural language processing for automated disorder normalization. *J. Biomed. Inf*, **57**, pp. 28–37.

Leaman,R. *et al*. (2015b) tmChem: a high performance approach for chemical named entity recognition and normalization. *J. Cheminf*., **7**, S3.

Lee,H.C. *et al*. (2015) *An Enhanced CRF-Based System for Disease Name Entity Recognition and Normalization on BioCreative V DNER Task. Proc BioCreative Workshop*. Sevilla, Spain, pp. 226–233.

Li,J. *et al*. (2015) Annotating chemicals, diseases and their interactions in biomedical literature. *Proc BioCreative Workshop*. Seville, Spain, pp. 173–182.

Morgan,A.A. *et al*. (2008) Overview of BioCreative II gene normalization. *Genome Biol*., **9**, S3.

Okanohara,D. *et al*. (2006) *Improving the scalability of semi-markov conditional random fields for named entity recognition. 21st Int Conf on Comp Ling and 44th ACL. Association for Computational Linguistics*, pp. 465–472.

Porter,M.F. (1980) An algorithm for suffix stripping. *Program*, **14**, 130–137.

Pradhan,S. *et al*. (2015) Evaluating the state of the art in disorder recognition and normalization of the clinical narrative. *J. Am. Med. Inf. Assoc*., **22**, 143–154.

Pyysalo,S. and Ananiadou,S. (2014) Anatomical entity mention recognition at literature scale. *Bioinformatics*, **30**, 868–875.

Rebholz-Schuhmann,D. *et al*. (2010) CALBC silver standard corpus. *J. Bioinf. Comput. Biol*., **8**, 163–179.

Rocktaschel,T. *et al*. (2012) ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics*, **28**, 1633–1640.

Sohn,S. *et al*. (2008) Abbreviation definition identification based on automatic precision estimates. *BMC Bioinformatics*, **9**, 402.

Song,M. *et al*. (2015) PKDE4J: Entity and relation extraction for public knowledge discovery. *J. Biomed. Inf*, **57**, 320–332.

Taskar,B. *et al*. (2004) Max-margin Markov networks. In: Thrun,S. *et al*. (eds) *Adv Neural Inf Process Syst*. MIT Press, Cambridge, Massachusetts, USA.

Tsuruoka,Y. *et al*. (2007) Learning string similarity measures for gene/protein name dictionary look-up using logistic regression. *Bioinformatics*, **23**, 2768–2774.

Usami,Y. *et al*. (2011) Automatic acquisition of huge training data for biomedical named entity recognition. *BioNLP Workshop*. Portland, Oregon, pp. 65–73.

Vazquez,M. *et al*. (2011) Text mining for drugs and chemical compounds: methods, tools and applications. *Mol. Inf*., **30**, 506–519.

Wei,C.H. *et al*. (2015a) GNormPlus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed. Res. Int*., **2015**, 7.

Wei,C.H. *et al*. (2015b) SimConcept: a hybrid approach for simplifying composite named entities in biomedical text. *IEEE J. Biomed. Health Inf*., **19**, 1385–1391.

Wei,C.H. *et al*. (2015c) Overview of the BioCreative V Chemical Disease Relation (CDR) Task. *BioCreative Workshop*. pp. 154–166.