Simultaneously Linking Entities and Extracting Relations from Biomedical Text Without Mention-level Supervision

Trapit Bansal† and **Pat Verga***‡ and **Neha Choudhary***† and **Andrew McCallum**†
†University of Massachusetts, Amherst
†{tbansal, nchoudhary, mccallum}@cs.umass.edu
‡Google Research
‡patverga@google.com

Abstract

Understanding the meaning of text often involves reasoning about entities and their relationships. This requires identifying textual mentions of entities, linking them to a canonical concept, and discerning their relationships. These tasks are nearly always viewed as separate components within a pipeline, each requiring a distinct model and training data. While relation extraction can often be trained with readily available weak or distant supervision, entity linkers typically require expensive mention-level supervision – which is not available in many domains. Instead, we propose a model which is trained to simultaneously produce entity linking and relation decisions while requiring no mention-level annotations. This approach avoids cascading errors that arise from pipelined methods and more accurately predicts entity relationships from text. We show that our model outperforms a state-of-the art entity linking and relation extraction pipeline on two biomedical datasets and can drastically improve the overall recall of the system.

Introduction

Making complex decisions in domains like biomedicine and clinical treatments requires access to information and facts in a form that can be easily viewed by experts and is computable by reasoning algorithms. The predominant paradigm for storing this type of data is in a knowledge graph. Much of these facts are populated from hand curation by human experts, inevitably leading to high levels of incompleteness (Bodenreider 2004; Bollacker et al. 2008). To address this, researchers have focused on automatically constructing knowledge bases by directly extracting information from text (Ji et al. 2010).

This procedure can be broken down into three major components; identifying mentions of entities in text (Ratinov and Roth 2009; Lample et al. 2016; Strubell et al. 2017), linking mentions of the same entity together into a single canonical concept (Cucerzan 2007; Gupta, Singh, and Roth 2017; Raiman and Raiman 2018), and identifying relationships occurring between those entities (Bunescu and Mooney 2007; Wang et al. 2016; Verga, Strubell, and McCallum 2018).

*work done while authors were at UMass Amherst

Copyright © 2020, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

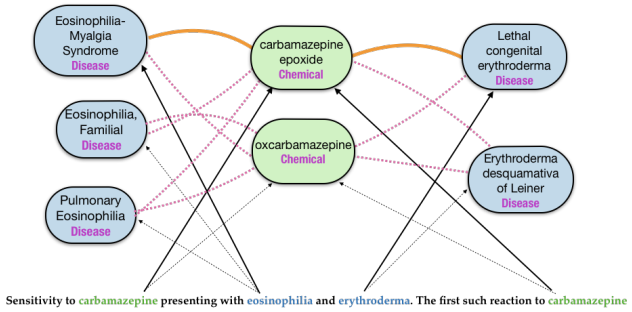

Figure 1: Overview of the graph extraction task. Given a document represented as a title and abstract. Text mentions are denoted with color and each can link to one of several possible entities. The model considers the full set of entity linking and relation edges (all lines) and predicts the graph of true entities and relations (solid lines) represented in the text. Dashed lines show possible (incorrect) edges and solid lines show the true edges.

These three stages are nearly always treated as separate serial components in an extraction pipeline and current state-of-the-art approaches train separate machine learning models for each component, each with their own distinct training data. More precisely, this data consists of mention-level supervision, that is individual instances of entities and relations which are identified and demarcated in text. This type of data can be prohibitively expensive to acquire, particularly in domains like biomedicine where expert knowledge is required to understand and annotate relevant information.

In contrast, forms of distant supervision are readily available as database entries in existing knowledge bases. This type of information encodes global properties about entities and their relationships without identifying specific textual instances of those facts. This form of distant supervision has been successfully applied to relation extraction models (Mintz et al. 2009; Surdeanu et al. 2012; Riedel et al. 2013). However, all of these methods still consume entity linking decisions as a preprocessing step, and unfortunately, accurate entity linkers and the mention-level supervision required to train them do not exist in many domains.

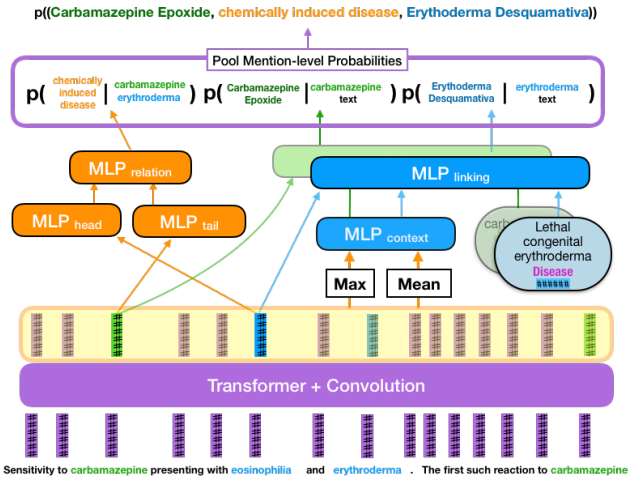

Figure 2: Architecture of the SNERL model. The text of the title and abstract are mapped to word embeddings which is then contextually encoded using a transformer architecture. The left side of the figure shows the procedure for scoring an individual relation mention using a separate head and tail MLP fed to a $\text{MLP}_{\text{relation}}$. The right side shows the entity linking component. The $\text{MLP}_{\text{linking}}$ model takes as input, an entity mention, a context representation derived from the mean and max over all contextualized token embeddings, and a candidate entity representation. These three probabilities (relation prediction and the two entity linking predictions) make up a single mention-level prediction. All mention-level predictions corresponding to the same entities are then pooled to make a final entity-level prediction.

In this work, we instead develop a method to simultaneously link entities in the text and extract their relationships (see Fig. 1). Our proposed method, called SNERL (**S**imultaneous **N**eural **E**ntity-**R**elation **L**inker), can be trained by leveraging readily available resources from existing knowledge bases and *does not utilize any mention-level supervision*. In experiments performed on two different biomedical datasets, we show that our model is able to substantially outperform a state-of-the-art pipeline of entity linking and relation extraction by jointly training and testing the two tasks together.

Methodology

In this section, we describe the proposed model, Simultaneous Neural Entity-Relation Linker (SNERL), and how it's trained. The input to the model is the full title and abstract of an article and the output is the predicted graph of entities and relations represented in the text (see Fig. 1). This is done by first encoding the text using self-attention (Vaswani et al. 2017) to obtain a contextualized representation of each entity mention in the input. These contextualized representations are then used to predict both the distribution over entities at the mention-level and the distribution over relations at the mention-pair-level. These predicted probabilities are then combined for each mention-pair and pooled at the document-level to get a final probability for predicting the tuple (e_1, r, e_2) for the text (see Fig. 2).

Notations: Let $[N]$ denote the set of natural numbers $\{1, \ldots, N\}$. Each document consists of a set of words $\{x_i\}$ indexed by $i \in [V]$ where V is the vocabulary size. Entity mentions in the document are found using a named entity recognition (NER) system (Wei, Kao, and Lu 2013). Let $\{m_j\}$ for $j \in [M]$ be the set of mention start indices for the document, where M is the number of mentions in the document. For each mention string x_{m_i} we generate up to C candidate entities (see Candidate Generation for details). Let E be the set of all entities. Each document is annotated with the graph of entities and relations, given as a set of tuples $G_d = \{(e_k, r, e_l)\}$, where $e_k, e_l \in E$ and $r \in [R]$. This is obtained from a knowledge base under the strong distant supervision assumption (Mintz et al. 2009) (see Experiments section for details). Let $E_d \subset E$ be the set of entities in the annotations for the document d. $[a; b]$ denotes concatenation of vectors a and b.

Text Encoder

The initial input to our model is the full title and abstract of a biomedical article from PubMed. [1] The sequence is tokenized and each token is mapped to a n-dimensional word embedding. The sequence of word embeddings are the input to our text encoder. The text encoder is based on the Transformer architecture of Vaswani et al. (2017). The transformer applies multiple blocks of multi-head self-attention followed by width 1 convolutions. We follow Verga, Strubell, and McCallum (2018) and add additional width 5 convolutions between blocks. The reader is referred to the Supplementary for the specific details. The text encoder, after multiple blocks of transformations, generates position and context informed hidden representations for each word in the document. The output of the text encoder is an n-dimensional contextualized embedding h_i for each token x_i:

$$h_1, \ldots, h_N = \text{transformer}(x_1, \ldots, x_N)$$

From an efficiency perspective, we only encode the document once and use the contextualized token representations to predict both the entities and the relations.

Predicting entities

From the contextualized token representations $\{h_i\}$, we first obtain a document representation by concatenating the mean-pooled and max-pooled token representations and projecting it through a multi-layer perceptron (MLP).

$$\tilde{h} = W^2_{\text{doc}}(\text{ReLU}(W^1_{\text{doc}}[\text{mean}(\{h_i\}); \max(\{h_i\})]))$$

where $\text{mean}(\cdot)$ denotes an element-wise mean of a set of vectors and $\max(\cdot)$ denotes an element-wise max of a set of vectors. Now, for each mention, we generate candidates entities for the mention. Such a candidate generation step is often used in entity-linking models (Shen, Wang, and Han

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

2015) and in many domains, such as for Wikipedia entities, high quality candidates can be generated by using prior linking counts of mention surface forms to entities obtained from Wikipedia anchor texts (Ganea and Hofmann 2017; Raiman and Raiman 2018). However, such high quality candidate generation is not available in the biomedical domain and so we resort to an approximate string matching approach for generating candidate entities.

Candidate Generation: We followed procedures from previous work (Leaman and Lu 2016; Murty et al. 2018). Each mention was first normalized by removing all punctuation, lower-casing, and then stemming. Next, these strings were converted to tfidf vectors consisting of both word and character ngrams. We considered character ngrams of lengths two to five, and for words we considered unigrams and bigrams. The same procedure was also applied to convert all canonical string names and synonyms for entities in our knowledge base. Finally, candidates for each mention were generated according to their cosine similarity amongst all entities in the knowledge base.

For each candidate entity e_i with type t_i, we generate a n-dimensional entity embedding as $\tilde{e}_i = \hat{e}_i + t_i$, by adding an entity-specific embedding $\hat{e}_i$ and a n-dimensional entity type embedding t_i. The entity-specific embedding can be learned or it can be a pre-trained embedding obtained from another source such as entity descriptions (Ganea and Hofmann 2017; Xie et al. 2016) or by a graph embedding method (Yang et al. 2014). Now, for the i-th mention in the document, with starting index m_i, we consider h_{m_i} as a contextualized mention representation and define a score for predicting the candidate entity e for this mention using the candidate representation $\tilde{e}$, document representation $\tilde{h}$, and mention representation h_{m_i}. This is passed through a softmax function, normalizing over the set of candidates C_{m_i} for the mention to get a probability $p(e|m_i, \text{text})$ for linking the mention m_i to entity e.

$$\begin{aligned} l(e, m_i, \text{text}) &= W_l^2(\text{ReLU}(W_l^1[\tilde{e}; \tilde{h}; h_{m_i}]) \\ p(e|m_i, \text{text}) &= \underset{e \in C_{m_i}}{\text{softmax}}\left(l(e, m_i, \text{text})\right) \end{aligned} \tag{1}$$

We thus obtain a $(\text{M} \times \text{C})$ matrix of linking probabilities for the document, where M is the maximum number of entity mentions in the document and C is the maximum number of candidates per mention. *Note that there is no direct mention-level supervision available to train these probabilities*.

Predicting relations

Given the contextualized mention representation, we obtain a head and tail representation for each mention to serve as the head or tail entity of a relation tuple (e_i, r, e_j). This is done by using two MLP to project each mention representation.

$$\begin{aligned} e_{m_i}^{\text{head}} &= W_{\text{head}}^2(\text{ReLU}(W_{\text{head}}^1 h_{m_i})) \\ e_{m_j}^{\text{tail}} &= W_{\text{tail}}^2(\text{ReLU}(W_{\text{tail}}^1 h_{m_j})) \end{aligned}$$

The head and tail representations are then passed through an MLP to predict a score for every relation r for a pair of mentions m_i and m_j. We pass this score vector through a sigmoid function to get a probability of predicting the relation from the mention-pair.

$$\begin{aligned} s(r, m_i, m_j) &= W_r^2(\text{ReLU}(W_r^1[e_{m_i}^{\text{head}}; e_{m_i}^{\text{tail}}])) \\ p(r|m_i, m_j) &= \sigma(s(r, m_i, m_j)) \end{aligned} \tag{2}$$

We thus obtain a $(\text{M} \times \text{M} \times \text{R})$ matrix of probabilities for predicting all relations, where R is the maximum number of relations, from all pairs of entity mentions.

Combining entity and relation predictions

To predict the graph of entities and relations from the document, we need to assign a probability to every possible relation tuple (e_k, r, e_l). We first obtain the probability of predicting a tuple (e_k, r, e_l) from a mention-pair (m_i, m_j) by combining the probability for predicting the candidates for each of the mentions (1) and the relation prediction probability (2). If an entity is not a candidate for a mention then it's entity prediction probability is zero for that mention.

$$\begin{aligned} &p\left((e_k, r, e_l)|m_i, m_j, \text{text}\right) = \\ &p(e_k|m_i, \text{text})p(r|m_i, m_j)p(e_l|m_j, \text{text}) \end{aligned} \tag{3}$$

Then, the probability of extracting the tuple (e_k, r, e_l) from the entire document can be obtained by pooling over all mention pairs (m_i, m_j). For example, we can use max-pooling, which corresponds to the inductive bias that in order to extract a tuple we must find at least one mention pair for the corresponding entities in the document that is evidence for the tuple.

$$p\left((e_k, r, e_l)|\text{text}\right) = \max_{i,j} p\left((e_k, r, e_l)|m_i, m_j, \text{text}\right) \tag{4}$$

Soft maximum pooling: It has been observed previously (Verga, Neelakantan, and McCallum 2017; Das et al. 2017) that the hard max operation is not ideal for pooling evidence as it leads to very sparse gradients. Recent methods, thus use the logsumexp function for pooling over *logits*, which allows for more dense gradient updates. However, we cannot use the logsumexp function in our case to pool over the probabilities (3) as the result of logsumexp over independent probabilities is not guaranteed to be a probability (in $[0, 1]$). Thus, we use a different operator that is considered a smooth relaxation of the maximum (Bansal, Das, and Bhattacharyya 2015). Given a set of elements $\{a_i\}$, the smooth-maximum (smax) with temperature τ is defined as:

$$w_i = \underset{i}{\text{softmax}}\left(\frac{a_i}{\tau}\right); \quad \textit{smax}(\{a_i\}) = \sum_i w_i a_i$$

Note that for $\tau \to 0$ the result of *smax* tends to the maximum of the set and for $\tau \to \infty$ the result is the average of the set. Thus, *smax* can smoothly interpolate between these extremes. We use this *smax* pooling over probabilities in (4) with a learned temperature τ.

Training

We are given ground-truth annotation for the set of tuples in the document, $G_d = \{(e_k, r, e_l)\}$. We train based on the cross-entropy loss from predicted tuple probabilities (4).

Since we only have a subset of positive annotations, there is uncertainty in the set of negatives, and we deal with this by weighting the positive annotations by a weight w_t in the cross-entropy loss. Let $y_{krl} = 1$ if document is annotated with the relation tuple (e_k, r, e_l) and 0 otherwise, and p_{krl} be its predicted probability in (4), then we maximize $\log p(G_d|text)$:

$$\frac{1}{|G_d|} \sum_{k,r,l} w_t y_{krl} \log p_{krl} + (1 - y_{krl}) \log(1 - p_{krl})$$

In addition, since we can obtain *document-level* entity annotations from the set of annotated relation tuples, we can provide an additional document-level entity supervision to better train our entity linking probabilities. To do this, we perform max-pooling over all mentions for each candidate entity for the document in (1), to obtain a document-level entity prediction score $p(e|\text{text}) = \max_m p(e|m, \text{text})$. We compute a weighted cross-entropy for these document-level predictions, again up-weighting the positive entities with a weight w_e. In summary, we combine graph prediction and document-level entity prediction objectives similar to multi-task learning (Caruana 1993), so if E_d is the set of entities in annotation, we maximize:

$$\log p(G_d|text) + \alpha \log p(E_d|text) \quad (5)$$

Note that since we only have some positive annotations, there could be many mentions in the document for which the correct entity is not annotated. Thus, we down-weight the document-entity prediction term by α in the objective.

Technical Details: Since the size of G_d can be very large, in order to improve training efficiency we subsample the set of unannotated entities as the negative entities to a maximum of n^- per document. Pooling over the joint mention-level probability (4) requires an intermediate $(\text{L} \times \text{L} \times \text{M} \times \text{M} \times \text{R})$ tensor, where L is the total number of *candidate* entities for the document. Since this can be computationally prohibitive, we compute the top-k mentions per candidate entity based on the predicted probabilities (1) and only backpropagate the gradients through the top-k. We consider k as a hyperparameter and tune it on the validation set.

Experiments

Our experimental setting is that, for each test document (title and abstract), the model should predict the full graph of entity-relationships expressed in that document (a single example is depicted in Fig. 1). Thus, we evaluate on micro-averaged precision, recall and F1 for predicting *the entire set of annotated relation tuples* across documents. Our results show significant improvement in F1 over a state-of-the-art pipelined approach (Verga, Strubell, and McCallum 2018). Hyperparameters are in the Supplementary.

Baselines

All of our models use the same neural architecture described earlier, consume the same predicted entity mentions from an external NER model (Wei, Kao, and Lu 2013), and *differ* in how they produce entity linking decisions. The first two baselines take hard entity linking decisions as inputs and do not do any internal entity linking inference. Both these baselines are equivalent to the BRAN model from Verga, Strubell, and McCallum (2018) with two different ways of obtaining entity links for that model. This is a state-of-the-art *pipelined approach* to entity-relation extraction. We used an MLP as the relation scoring function for BRAN (similar to the SNERL model) as it performed better in experiments compared to the biaffine function used in the original paper.

BRAN (Top Candidate) produces entity linking decisions based on the highest scoring candidate entity (as described in 'Candidate Generation' section).

BRAN (Linker) produces entity linking decisions from a trained state-of-the-art entity linker. We followed BRAN and obtained entity links from Wei, Kao, and Lu (2013).

SNERL is our proposed model that does not take in any hard entity linking decisions as input and instead jointly predicts the full set of entities and relations within the text. For this model we considered 25 candidates per mention.

	Tuple Recall
Top 1 Candidates	67.0%
Top 25 Candidates	**80.0%**
Entity Linker	60.4%

Table 1: Oracle recall for predicting entity-relation tuples under various models for selecting entity prediction, on the development set of CTD dataset. The oracle assumes perfect relation extraction recall. Note that to correctly extract a given entity-relation tuple, both head and tail entities in that relation need to predicted correctly. Since SNERL does not take entity links as input and has access to the top 25 candidates to make it's entity prediction, it can provide significantly higher recall.

CTD Dataset

Our first set of experiments are on the CTD dataset first introduced in Verga, Strubell, and McCallum (2018). The data is derived from annotations in the Chemical Toxicology Database (Davis et al. 2018), a curated knowledge base containing relationships between chemicals, diseases, and genes. Each fact additionally contains a reference to the document (a scientific publication) where the annotator identified the relationship. Thus, these are used to obtain annotator identified entity-relationships in a given scientific publication. This type of document annotation is fairly common in biomedical knowledge bases, further motivating this work. This allows us to treat these annotations as a form of strong distant supervision (Mintz et al. 2009). Here annotations are at the document-level rather than the mention-level (as in typical supervised learning) or corpus-level (as in standard distant supervision).

An aspect of the document-level supervision is that the original facts were annotated over complete documents. However, due to paywalls we often only have access to titles and abstracts of papers. Therefore, there is no guarantee that the relationship is actually expressed in the title or abstract. Verga, Strubell, and McCallum (2018), thus, filtered the CTD dataset to only consider those entity-relation tuples

Model	Precision	Recall	F1
BRAN (Top Candidate)	30.5	29.5	30.0
BRAN (Linker)	33.2	28.1	30.5
SNERL	**41.1**	**43.4**	**42.2**

Table 2: Precision, Recall, and F1 for the full CTD test set. Bold values are statistically significant (p-value < 0.05 using Wilcoxon signed-rank test) over the non-bold values in the same column.

where both entities are found in the text, for some mention, by the external entity linker. This ensures that all filtered tuples can be predicted by the model. However, this removes many correct entity-relationships that were indeed present but were filtered because those entities cannot be predicted by the entity linking model. We remedy this and create a more challenging train/development/test split from the entire CTD annotations, where we keep all entity-relationships in which the participating entities are a *candidate* for some mention in the document. That is, for each annotated tuple (e_1, r, e_2) between entities e_1 and e_2 in document D, we consider that tuple if both e_1 and e_2 are candidates for some mention in D. Note that we used the candidate-generation approach described previously for generating the candidates for the mentions[2]. Dataset statistics are in the Supplementary. We consider this as the Full CTD Dataset as it does not give an advantage to any particular entity linking model, but for completeness also evaluate on the subset filtered according to the original paper (we refer to this as BRAN-filtered).

To illustrate how the cascading errors of a pipelined approach of first predicting entity links and then predicting relations can degrade performance, we computed an oracle recall for the tuple prediction task on the development set of CTD. For this, we consider perfect accuracy on relation prediction, so the recall on tuple extraction is limited only by the entity linking accuracy. We consider three methods for entity linking: predicting the top candidate (based on the string similarity score from candidate generation), an oracle which can select the correct entity (if present) from the top 25 candidates and the trained entity liking model used by BRAN. Table 1 shows the results. We can see that errors from the entity linking step significantly restrict the models performance in pipelined approaches. On the other hand, if the model can infer the entity links (from top 25 candidates) jointly with the relations, it ameliorates this problem of cascading errors, potentilally leading to much higher recall.

Results on Full CTD data: In table 2, we can see that the SNERL model that jointly considers both entity and relations together drastically outperforms the models that take hard linking decisions from an external model. This is primarily due to huge drop in recall caused by cascading errors.

Results on BRAN Filtered CTD data: We also report results using the original filtering approach of Verga, Strubell, and McCallum (2018). Importantly, this approach gives a substantial advantage to the BRAN (Linker) baseline as the data is filtered to only consider the relationships for which it could potentially make a prediction. In table 3, we can see that in spite of this disadvantage, the SNERL model is able to perform comparably to the BRAN (Linker) baseline.

[2]we consider up to 250 candidates entities per mention for the data filtering

Model	Precision	Recall	F1
BRAN (Top Candidate)	43.0	49.0	45.8
BRAN (Linker)	**45.7**	53.8	**49.4**
SNERL	**45.2**	**55.2**	**49.7**

Table 3: Precision, Recall, and F1 for the BRAN-filtered CTD test data (i.e. filtered to tuples where BRAN can make a prediction). Bold values are statistically significant (p-value < 0.05 using Wilcoxon signed-rank test) over the non-bold values in the same column, and the difference between multiple bold values in the same column is not statistically significant.

CDR Entity Linking Performance

In order to evaluate how much of the success of the SNERL model can be attributed to the entity linking component (1), we evaluated its performance on the BioCreative V Chemical Disease Relation dataset (CDR) introduced in Wei et al. (2015). Similar to the CTD dataset, CDR was also originally derived from the Chemical Toxicology Database. Expert annotators chose 1,500 of those documents and exhaustively annotated all *mentions* of chemicals and diseases in the text. Additionally, each mention was assigned its appropriate entity linking decision. We use this dataset as a gold standard to *validate* our entity linking models. *Note that we do not use this data for training, but only for evaluation*.

We use the model that was trained on the CTD data and make it predict entities for every mention on the test set of CDR. We follow the standard practice of using the gold mention boundaries for evaluation only, to not confound the entity linking performance with mention-detection performance. In Table 4, we see that our SNERL does learn to link entities better than the top candidate. As is common when evaluating on this data, we consider document-level rather than mention-level entity linking evaluation (Leaman and Lu 2016), that is, how does the set of predicted entities compare to the gold set annotated in the document. Note that the SNERL model additionally benefits from jointly predict entities and relations. Breakdown of the results into Chemical and Disease prediction performance can be found in Supplementary.

Model	Precision	Recall	F1
Top Candidate	79.0	86.8	82.7
SNERL	**83.3**	**90.2**	**86.6**

Table 4: Results for entity linking on the CDR dataset. Bold values are statistically significant (p-value < 0.05 using Wilcoxon signed-rank test).

Disease-Phenotype Relations

To further probe the performance of our model we created a dataset of disease / phenotype (aka symptom) relations. The goal here is to identify specific symptoms caused by a disease. This type of information is particularly important in clinical treatments as it can lead to earlier diagnosis of rare diseases, faster application of appropriate interventions, and better overall outcomes for patients. This task also serves to further motivate our methods as accurate entity linking models for phenotypes are not readily available, nor is sufficient mention-level training data to build a supervised classifier.

Relation Annotations: We created this dataset with a similar technique to the construction of the CTD dataset. We started from the relations in the Human Phenotype Ontology (Köhler et al. 2018) that were annotated with a document containing that relationship.

Mention Detection: For disease mention detection we followed the same procedure as CTD dataset and used the annotated mentions from Wei, Kao, and Lu (2013). Because there is not a readily available phenotype tagger, we trained our own model to identify mentions of phenotypes in text. We trained an iterated dilated convolution model Strubell et al. (2017).[3] Our training data came from Groza et al. (2015), which we split into train, dev, and test sets (see Supplementary). Our NER model achieved a micro F1 score of 72.57.

We observed that disease and phenotype entity spans are often overlapping and nested. We thus over-generate the set of mentions by taking the predictions from both the taggers and adding them to the set of all mentions for the document, since our model is able to pool over all theses mentions even if they overlap.

Entity Linking: We followed a similar procedure as described in section to generate phenotype entity linking candidates. Using the small set of gold entity linked text mentions from Groza et al. (2015) we were able to estimate our candidate's entity linking accuracy. Our top candidate achieved an accuracy of 46.8% while the recall for 100 candidates was 76.5%. This demonstrates the additional difficulty of the disease-phenotype dataset as these candidate accuracies are much lower than the results for CTD data. See Supplementary Figure 1 for recall of the candidate set at different values of K.

Köhler et al. (2018) annotations make use of several disease vocabularies from OMIM (Hamosh et al. 2005), ORPHANET (Pavan et al. 2017) and DECIPHER (Bragin et al. 2013) databases. For generating disease candidates, we use disease name strings from all of these. The external entity linker that we used from Wei, Kao, and Lu (2013) links diseases to the MeSH disease vocabulary. To align these with our disease-phenotype relation annotations, we use the MEDIC database (Davis et al. 2012) for mapping OMIM disease terms into the MeSH vocabulary.

The final dataset annotations were selected by filtering based on entities that can be found in document when considering up to 250 candidates per mention. See Supplementary for dataset statistics.

[3]https://github.com/iesl/dilated-cnn-ner

Model	Precision	Recall	F1
BRAN (Top Candidate)	8.9	5.3	6.6
BRAN (Linker)	11.3	6.6	8.3
SNERL	**12.8**	**10.9**	**11.8**

Table 5: Results on the disease phenotype dataset . Bold values are statistically significant (p-value < 0.05 using Wilcoxon signed-rank test) over non-bold values.

Pre-training Entity Embedding Since the dataset has many unseen entities at test time, we need a method to address these unseen entities as generating the linking probabilities in (1) requires an entity embedding. For this, we obtained entity descriptions for the phenotypes and encoded them using pre-trained sentence embedding from BioSentVec (Chen, Peng, and Lu 2018). However, not all test entities have descriptions. So, in addition to the descriptions we trained a graph embedding model, DistMult (Yang et al. 2014), on the graph obtained from the set of all annotations in Human Phenotype Ontology excluding the dev/test annotations. We project both these pre-trained embeddings using a learned linear transformation and sum the description and graph embedding to obtain the entity-specific embedding $\hat{e}$.

Baselines We use the same baselines as before. For BRAN (Linker), the disease entity links come Wei, Kao, and Lu (2013) and since we don't have access to an accurate pre-trained phenotype entity linking model, this model also uses the top phenotype candidate as a hard phenotype entity linking decision.

Results Our disease-phenotype results show a similar trend to those from the CTD experiments. Overall, the BRAN (Top Candidate) model performs the worst and the SNERL model outperforms both models that use hard entity linking decisions.

Overall, our results indicate that this particular task is extremely challenging. This is likely the combination of several difficulties. The first is that the candidate set itself is not as accurate as the ones from the CTD experiment which we can see from comparing the top candidate accuracy of 46.8% with the Top Candidate results in Table 4. Since we rely on the candidate set to filter the annotations for the documents, we might end up with significant annotations that are not present in the title and abstract. Secondly, the amount of training data is significantly less (see Supplementary) than in the CTD experiments, requiring research into unsupervised approaches (Devlin et al. 2018) for this data. Lastly, dealing with out-of-vocabulary entities at test time required additional pre-training, and our analysis indicated that these are not highly predictive for mention-level disambiguation due to the sparsity of the graph training data. Looking into more sophisticated embedding methods (Xie et al. 2016; Gupta, Singh, and Roth 2017; Bansal et al. 2019) and methods for dealing with unseen entities would be an important problem for future work.

Related Work

Extracting entities and relations from text has been widely studied over the past few decades. In the biomedical domain specifically, there has been substantial progress on entity mention detection (Greenberg et al. 2018; Wei, Kao, and Lu 2015) and entity linking (often referred to as normalization in the bio NLP community) (Leaman and Gonzalez 2008; Leaman, Islamaj Doğan, and Lu 2013; Leaman, Wei, and Lu 2015; Leaman and Lu 2016), and relation extraction (Wei et al. 2016; Krallinger et al. 2017). There have also been numerous works that have identified both entity mentions and relationships from text in both the general domain (Miwa and Bansal 2016) and in the biomedical domain (Li et al. 2017; Ammar et al. 2017; Verga, Strubell, and McCallum 2018). Leaman and Lu (2016) showed that jointly considering named entity recognition (NER) and linking led to improved performance.

A few works have shown that jointly modeling relations and entity linking can improve performance. Le and Titov (2018) improved entity linking performance by modeling latent relations between entities. This is similar to coherence models (Ganea and Hofmann 2017) in entity linking which consider the joint assignment of all linking decisions, but is more tractable as it focuses on only pairs of entities in a short context. Luan et al. (2018) created a multi-task learning model for predicting entities, relations, and coreference in scientific documents. This model required supervision for all three tasks and predictions amongst the different tasks were made independently rather than jointly. To the best of our knowledge, SNERL is the first model that simultaneously links entities and predicts relations without requiring expensive mention-level annotation.

Conclusion

In this paper, we presented a novel method, SNERL, to simultaneously predict entity linking and entity relation decisions. SNERL can be trained without any mention-level supervision for entities or relations, and instead relies solely on weak and distant supervision at the document-level, readily available in many biomedical knowledge bases. The proposed model performs favorably as compared to a state-of-the-art pipeline approach to relation extraction by avoiding cascading errors, while requiring less expensive annotation, opening possibilities for knowledge extraction in low-resource and expensive to annotate domains.

Acknowledgments

We thank Andrew Su and Haw-Shiuan Chang for early discussions on the disease-phenotype task. This work was supported in part by the UMass Amherst Center for Data Science and the Center for Intelligent Information Retrieval, in part by the Chan Zuckerberg Initiative under the project Scientific Knowledge Base Construction, and in part by the National Science Foundation under Grant No. IIS-1514053. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Ammar, W.; Peters, M.; Bhagavatula, C.; and Power, R. 2017. The ai2 system at semeval-2017 task 10 (scienceie): semi-supervised end-to-end entity and relation extraction. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, 592–596.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bansal, T.; Juan, D.-C.; Ravi, S.; and McCallum, A. 2019. A2n: Attending to neighbors for knowledge graph inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4387–4392.

Bansal, T.; Das, M.; and Bhattacharyya, C. 2015. Content driven user profiling for comment-worthy recommendations of news and blog articles. In *Proceedings of the 9th ACM Conference on Recommender Systems*, 195–202. ACM.

Bodenreider, O. 2004. The unified medical language system (umls): integrating biomedical terminology. *Nucleic acids research* 32(suppl_1):D267–D270.

Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, 1247–1250. AcM.

Bragin, E.; Chatzimichali, E. A.; Wright, C. F.; Hurles, M. E.; Firth, H. V.; Bevan, A. P.; and Swaminathan, G. J. 2013. Decipher: database for the interpretation of phenotype-linked plausibly pathogenic sequence and copy-number variation. *Nucleic acids research* 42(D1):D993–D1000.

Bunescu, R., and Mooney, R. 2007. Learning to extract relations from the web using minimal supervision. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, 576–583.

Caruana, R. 1993. Multitask learning: A knowledge-based source of inductive bias. *Proceedings of the Tenth International Conference on International Conference on Machine Learning (ICML)* 41–48.

Chen, Q.; Peng, Y.; and Lu, Z. 2018. Biosentvec: creating sentence embeddings for biomedical texts. *arXiv preprint arXiv:1810.09302*.

Cucerzan, S. 2007. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Das, R.; Neelakantan, A.; Belanger, D.; and McCallum, A. 2017. Chains of reasoning over entities, relations, and text using recurrent neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 1, 132–141.

Davis, A. P.; Wiegers, T. C.; Rosenstein, M. C.; and Mattingly, C. J. 2012. Medic: a practical disease vocabulary used at the comparative toxicogenomics database. *Database* 2012.

Davis, A. P.; Grondin, C. J.; Johnson, R. J.; Sciaky, D.; McMorran, R.; Wiegers, J.; Wiegers, T. C.; and Mattingly, C. J. 2018. The comparative toxicogenomics database: update 2019. *Nucleic acids research* 47(D1):D948–D954.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ganea, O.-E., and Hofmann, T. 2017. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*.

Greenberg, N.; Bansal, T.; Verga, P.; and McCallum, A. 2018. Marginal likelihood training of bilstm-crf for biomedical named entity recognition from disjoint label sets. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2824–2829.

Groza, T.; Köhler, S.; Doelken, S.; Collier, N.; Oellrich, A.; Smedley, D.; Couto, F. M.; Baynam, G.; Zankl, A.; and Robinson, P. N. 2015. Automatic concept recognition using the human phenotype ontology reference and test suite corpora. *Database* 2015.

Gupta, N.; Singh, S.; and Roth, D. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2681–2690.

Hamosh, A.; Scott, A. F.; Amberger, J. S.; Bocchini, C. A.; and McKusick, V. A. 2005. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic acids research* 33(suppl_1):D514–D517.

Ji, H.; Grishman, R.; Dang, H. T.; Griffitt, K.; and Ellis, J. 2010. Overview of the tac 2010 knowledge base population track. In *Third Text Analysis Conference (TAC 2010)*, volume 3, 3–3.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Köhler, S.; Carmody, L.; Vasilevsky, N.; Jacobsen, J. O. B.; Danis, D.; Gourdine, J.-P.; Gargano, M.; Harris, N. L.; Matentzoglu, N.; McMurry, J. A.; et al. 2018. Expansion of the human phenotype ontology (hpo) knowledge base and resources. *Nucleic acids research* 47(D1):D1018–D1027.

Krallinger, M.; Rabal, O.; Akhondi, S. A.; et al. 2017. Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, 141–146.

Kulick, S.; Bies, A.; Liberman, M.; Mandel, M.; McDonald, R.; Palmer, M.; Schein, A.; Ungar, L.; Winters, S.; and White, P. 2004. Integrated annotation for biomedical information extraction. In *HLT-NAACL 2004 Workshop: Linking Biological Literature, Ontologies and Databases*.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Le, P., and Titov, I. 2018. Improving entity linking by modeling latent relations between mentions. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1595–1604.

Leaman, R., and Gonzalez, G. 2008. Banner: an executable survey of advances in biomedical named entity recognition. In *Biocomputing 2008*. World Scientific. 652–663.

Leaman, R., and Lu, Z. 2016. Taggerone: joint named entity recognition and normalization with semi-markov models. *Bioinformatics* 32(18):2839–2846.

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

Leaman, R.; Wei, C.-H.; and Lu, Z. 2015. tmchem: a high performance approach for chemical named entity recognition and normalization. *Journal of cheminformatics* 7(1):S3.

Li, F.; Zhang, M.; Fu, G.; and Ji, D. 2017. A neural joint model for entity and relation extraction from biomedical text. *BMC bioinformatics* 18(1):198.

Luan, Y.; He, L.; Ostendorf, M.; and Hajishirzi, H. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 3219–3232.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, 1003–1011. Association for Computational Linguistics.

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1105–1116.

Murty, S.; Verga, P.; Vilnis, L.; Radovanovic, I.; and McCallum, A. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 1, 97–109.

Pavan, S.; Rommel, K.; Marquina, M. E. M.; Höhn, S.; Lanneau, V.; and Rath, A. 2017. Clinical practice guidelines for rare diseases: the orphanet database. *PloS one* 12(1):e0170365.

Raiman, J. R., and Raiman, O. M. 2018. Deeptype: multilingual entity linking by neural type system evolution. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ratinov, L., and Roth, D. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the thirteenth conference on computational natural language learning*, 147–155. Association for Computational Linguistics.

Riedel, S.; Yao, L.; McCallum, A.; and Marlin, B. M. 2013. Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 74–84.

Shen, W.; Wang, J.; and Han, J. 2015. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering* 27(2):443–460.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Strubell, E.; Verga, P.; Belanger, D.; and McCallum, A. 2017. Fast and accurate entity recognition with iterated dilated convolutions. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2670–2680.

Surdeanu, M.; Tibshirani, J.; Nallapati, R.; and Manning, C. D. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, 455–465. Association for Computational Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 5998–6008.

Verga, P.; Neelakantan, A.; and McCallum, A. 2017. Generalizing to unseen entities and entity pairs with row-less universal schema. In *Proceedings of the 15th Conference of the European Chapter*

of the Association for Computational Linguistics, volume 1, 613–622.

Verga, P.; Strubell, E.; and McCallum, A. 2018. Simultaneously Self-attending to All Mentions for Full-Abstract Biological Relation Extraction. In *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT)*.

Wang, L.; Cao, Z.; de Melo, G.; and Liu, Z. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, volume 1, 1298–1307.

Wei, C.-H.; Peng, Y.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Li, J.; Wiegers, T. C.; and Lu, Z. 2015. Overview of the biocreative v chemical disease relation (cdr) task. In *Proceedings of the fifth BioCreative challenge evaluation workshop*, 154–166.

Wei, C.-H.; Peng, Y.; Leaman, R.; Davis, A. P.; Mattingly, C. J.; Li, J.; Wiegers, T. C.; and Lu, Z. 2016. Assessing the state of the art in biomedical relation extraction: overview of the biocreative v chemical-disease relation (cdr) task. *Database* 2016.

Wei, C.-H.; Kao, H.-Y.; and Lu, Z. 2013. Pubtator: a web-based text mining tool for assisting biocuration. *Nucleic acids research* 41(W1):W518–W522.

Wei, C.-H.; Kao, H.-Y.; and Lu, Z. 2015. Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international* 2015.

Xie, R.; Liu, Z.; Jia, J.; Luan, H.; and Sun, M. 2016. Representation learning of knowledge graphs with entity descriptions. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

SUPPLEMENTARY MATERIAL

Transformer Text Encoder

The model takes a sequence of N word embeddings as input, $\{x_1, \ldots, x_N\}$. Since the Transformer has no innate notion of position, the model relies on positional embeddings which are added to the input token embeddings. The positional embeddings are also learned parameters of the model. Thus, we get the token representations:

$$s_i = x_i + p_i$$

Transformer (Vaswani et al. 2017) is made up of B blocks. Each Transformer block, denoted transformer_k, has its own set of parameters and consists of two components: multi-head attention followed by a series of convolutions. The output for token i of block k, b_i^k, is connected to its input b_i^{k-1} with a residual connection:

$$b_i^k = b_i^{k-1} + \text{transformer}_k(b_i^{k-1}) \quad (6)$$

In each block, multi-head attention (Vaswani et al. 2017) applies self-attention multiple times over the same inputs using separately normalized parameters (attention heads) and combines the results. Each head in multi-head self-attention updates its input b_i^{k-1} by performing a weighted sum over all tokens in the sequence, weighted by their importance for modeling token i. Refer to Vaswani et al. (2017) for details of multi-head attention. The outputs of the individual attention heads are concatenated, to give the output of multi-head attention at the i-th token, $o_i = [o_{i1} \ \ldots ; o_{in_h}]$. This is followed by layer normalization (Ba, Kiros, and Hinton 2016), and two width-1 convolutions. Following (Verga, Strubell, and McCallum 2018), we add a third layer with kernel width 5 convolutions, which allows explicit n-gram modeling useful for relation extraction. This gives the output at the i-th token for the k-th transformer block in (6).

The sequence of representations at each token obtained after B blocks of processing, described above, is the final output of the transformer text encoder:

$$\text{transformer}(x_1, \ldots, x_N) = h_1, \ldots, h_N = b_1^B, \ldots, b_N^B$$

CTD Dataset

The number of documents in the splits of CTD dataset are given in Table 6. There are 19933 entities and 14 relation types in this data. The number of entity-relationship tuples in train/dev/test are given in Table 7.

Data	Train	Dev	Test
Full CTD	52,003	8,177	8,284
BRAN-filtered CTD	45,586	5,857	5,804

Table 6: Number of documents in CTD dataset

Data	Train	Dev	Test
Full CTD	140,121	34,213	36,656
BRAN-filtered CTD	115,319	14,141	14,372

Table 7: Number of entity-relationship tuples in CTD dataset

Entity Linking on CDR

Table 8 shows the entity linking performance for diseases on the CDR dataset. Table 9 shows the entity linking performance for diseases on the CDR dataset.

Model	Precision	Recall	F1
Top Candidate	77.3	80.8	79.0
SNERL	**83.6**	**86.0**	**84.8**

Table 8: Disease entity linking on the CDR dataset

Implementation Details

All word embeddings are randomly initialized. Text is tokenized using the Genia tokenizer (Kulick et al. 2004). We used dropout (Srivastava et al. 2014) at the input word embeddings (p_i), on attention weights in the transformer (Vaswani et al. 2017) (p_t), after head and tail projection MLP (p_s), and after the first layers of the relation (p_s) and linking MLP (p_s). We also apply dropout to the input words replacing words with a special UNK token (p_w). Note that the values of dropout p_* reported here are keep probabilities. We used Adam (Kingma and Ba 2014) for optimization with a learning rate of 0.001. We tuned the dropout rates, weights

Model	Precision	Recall	F1
Top Candidate	81.4	95.6	87.9
SNERL	**83.0**	**96.5**	**89.3**

Table 9: Chemical entity linking on the CDR dataset

Split	Docs	Relations
Train	401	1631
Dev	86	303
Test	86	455

Table 10: Statistics for disease phenotype relation data

for the cross-entropy term w_t, w_e, number of blocks B of transformer, number of heads n_h, number of negative samples n^-, k for the number of top mentions, and the weight α for the objective.

On CTD, full dataset, the best hyperparameters were: $p_i = 0.25$, $p_t = 0.25$, $p_s = .15$, $p_w = 0.2$, $w_t = 5.0$, $w_e = 2.0$, $n^- = 100$, $B = 4$, $n_h = 2$ $k = 15$, $\alpha = 0.1$. We used embedding dimension $n = 128$.

On CTD, BRAN-filtered dataset, the best hyperparameters were: $p_i = 0.2$, $p_t = 0.$, $p_s = 0.$, $p_w = 0.3$, $w_t = 5.0$, $w_e = 0.$, $n^- = 200$, $B = 2$, $n_h = 8$ $k = 10$, $\alpha = 0.$. We used embedding dimension $n = 128$.

On the Disease-Phenotype dataset, the best hyperparameters were: $p_i = 0.4$, $p_t = 0.4$, $p_s = 0.15$, $p_w = 0.15$, $w_t = 5.0$, $w_e = 0.$, $n^- = 400$, $B = 4$, $n_h = 4$ $k = 10$, $\alpha = 0.$. We used embedding dimension $n = 64$.

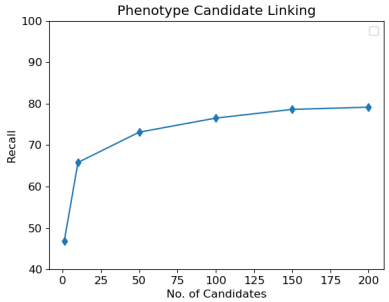

Figure 3: Recall for different numbers of candidates for phenotype entity linking

Disease-Phenotype Dataset

Our final NER model achieved a micro F1 score of 75.00 on the development set and 72.57. The train/dev/test split consisted of 173/23/23 documents with 1294/118/160 metions respectively.

Table 10 shows the train/dev/test splits for the disease-phenotype relation extraction dataset.

Figure 3 shows the recall@k for phenotype candidate set.