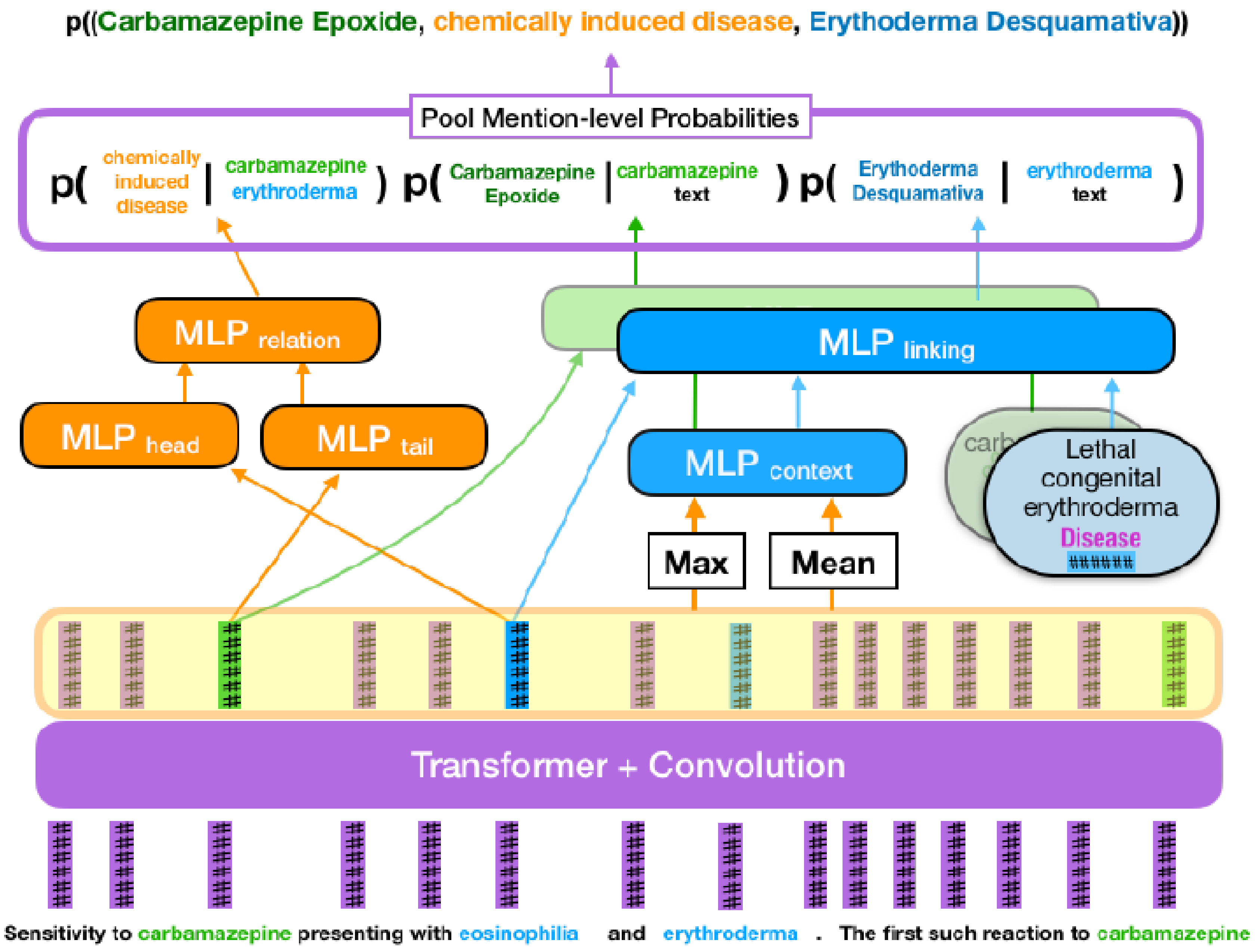
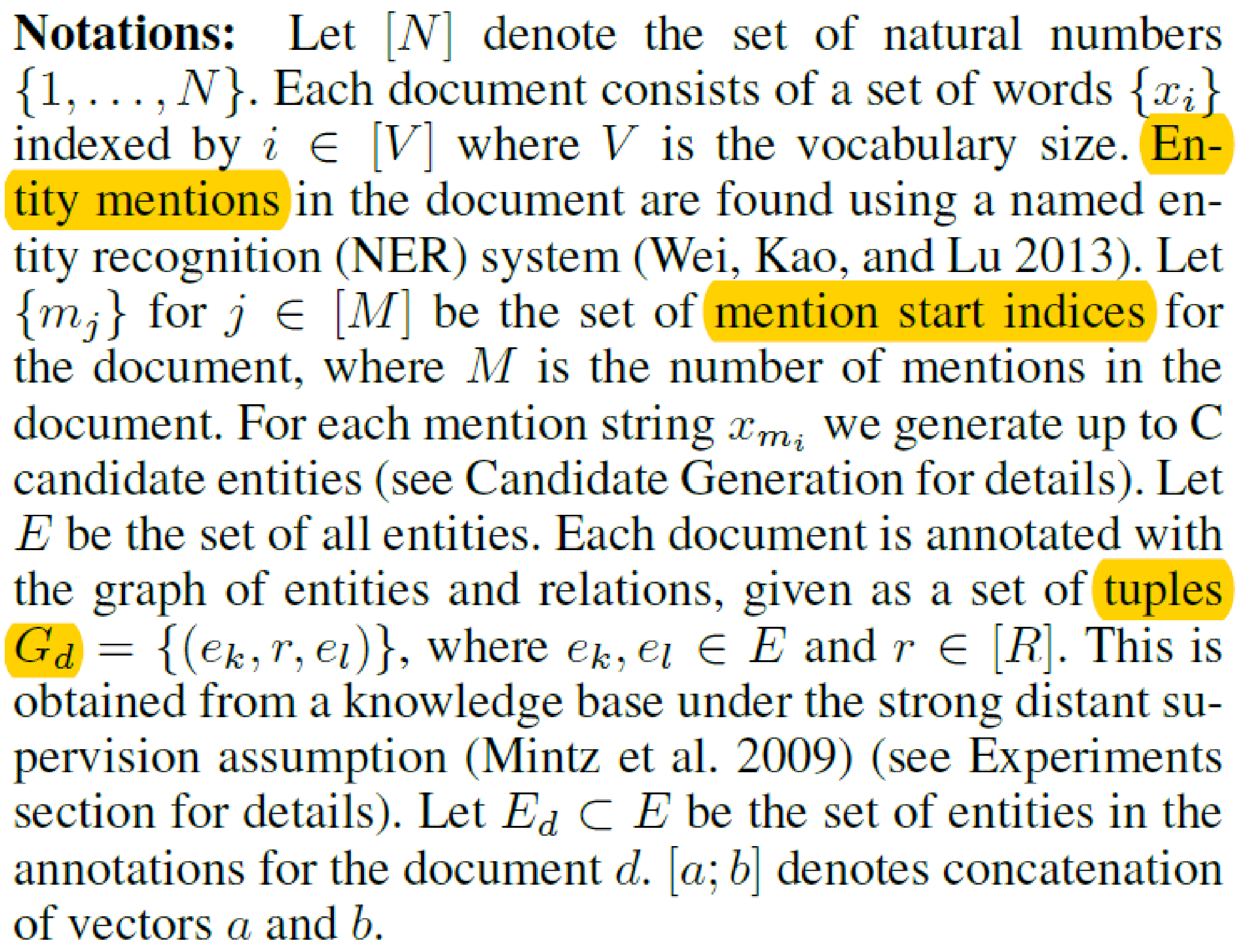
Simultaneously Linking Entities and Extracting Relations from Biomedical Text Without Mention-level Supervision

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

Summary

- Simultaneously link entities in the text and extract their relationships
- Our proposed method, called **SNERL (Simultaneous Neural Entity-Relation Linker)**, can be trained by leveraging readily available resources from existing knowledge bases and does not utilize any mentionlevel supervision.


p((Carbamazepine Epoxide, chemically induced disease, Erythoderma Desquamativa))
Pool Mention-level Probabilities
p(chemically induced disease | carbamazepine erythroderma)
p(Carbamazepine Epoxide | carbamazepine text)
p(Erythoderma Desquamativa | erythroderma text)
MLP relation
MLP head
MLP tail
MLP linking
MLP context
Max
Mean
carb
Lethal congenital erythroderma
Disease
Transformer + Convolution
Sensitivity to carbamazepine presenting with eosinophilia and erythroderma . The first such reaction to carbamazepine

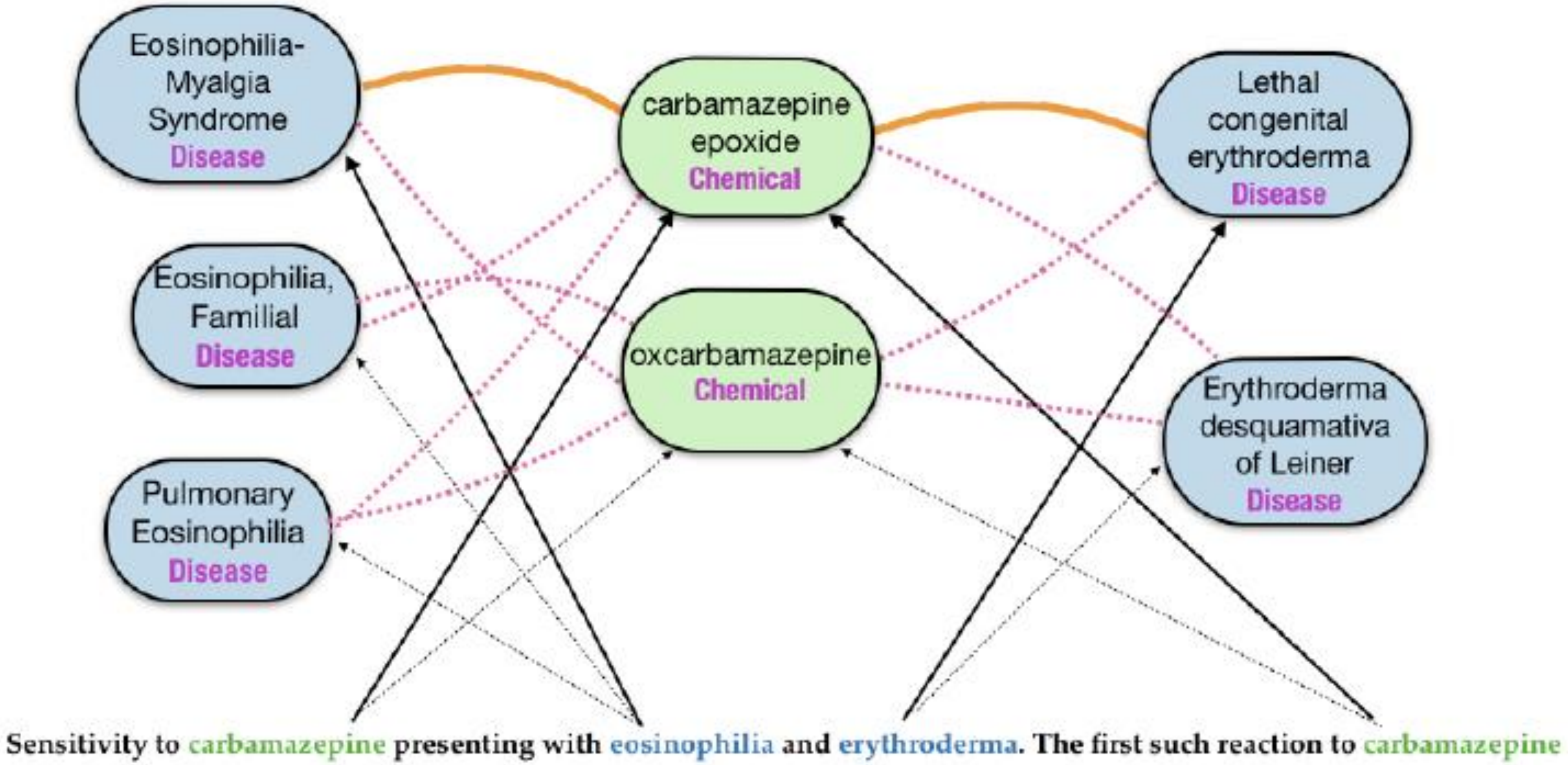


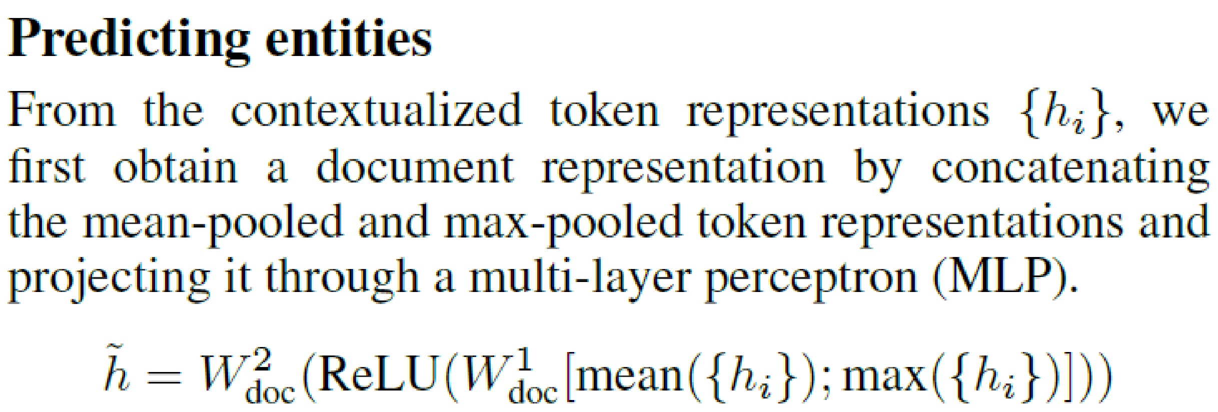
Notations: Let $[N]$ denote the set of natural numbers $\{1, \ldots, N\}$. Each document consists of a set of words $\{x_i\}$ indexed by $i \in [V]$ where V is the vocabulary size. Entity mentions in the document are found using a named entity recognition (NER) system (Wei, Kao, and Lu 2013). Let $\{m_j\}$ 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

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


Eosinophilia-Myalgia Syndrome
Disease
Eosinophilia, Familial
Disease
Pulmonary Eosinophilia
Disease
carbamazepine epoxide
Chemical
oxcarbamazepine
Chemical
Lethal congenital erythroderma
Disease
Erythroderma desquamativa of Leiner
Disease
Sensitivity to carbamazepine presenting with eosinophilia and erythroderma. The first such reaction to carbamazepine



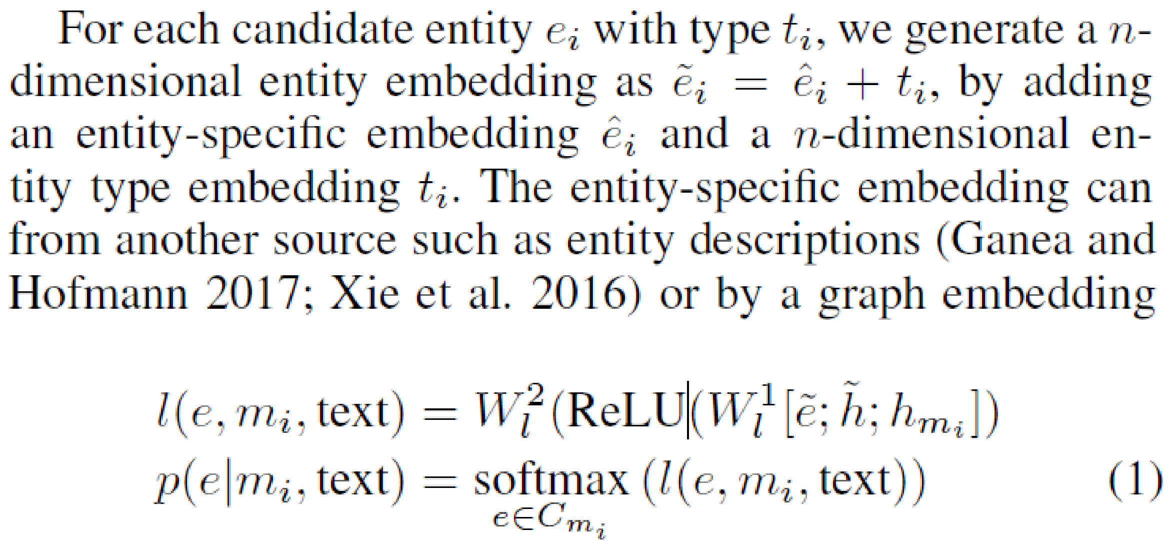
Predicting entities

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).

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

Candidate Generation:

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. 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 from another source such as entity descriptions (Ganea and Hofmann 2017; Xie et al. 2016) or by a graph embedding

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

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/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.

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

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.

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

Experiment

- 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**.
- 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.

	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.

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.

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: Results 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.

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: Results for BRAN-filtered CTD test data (i.e. filtered to tuples where BRAN can make a prediction). Bold values are statistically significant over non-bold values in the same column, and the difference between multiple bold values in a column is not statistically significant.

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).

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).