Hierarchical Losses and New Resources for Fine-grained Entity Typing and Linking

Shikhar Murty* UMass Amherst smurty@cs.umass.edu

Patrick Verga* UMass Amherst pat@cs.umass.edu

Luke Vilnis UMass Amherst luke@cs.umass.edu

Irena Radovanovic Chan Zuckerberg Initiative iradovanovic@chanzuckerberg.com

Andrew McCallum UMass Amherst mccallum@cs.umass.edu

Abstract

Extraction from raw text to a knowledge base of entities and fine-grained types is often cast as prediction into a flat set of entity and type labels, neglecting the rich hierarchies over types and entities contained in curated ontologies. Previous attempts to incorporate hierarchical structure have yielded little benefit and are restricted to shallow ontologies. This paper presents new methods using real and complex bilinear mappings for integrating hierarchical information, yielding substantial improvement over flat predictions in entity linking and fine-grained entity typing, and achieving new state-of-the-art results for end-to-end models on the benchmark FIGER dataset. We also present two new human-annotated datasets containing wide and deep hierarchies which we will release to the community to encourage further research in this direction: *MedMentions*, a collection of PubMed abstracts in which 246k mentions have been mapped to the massive UMLS ontology; and *TypeNet*, which aligns Freebase types with the WordNet hierarchy to obtain nearly 2k entity types. In experiments on all three datasets we show substantial gains from hierarchy-aware training.

1 Introduction

Identifying and understanding entities is a central component in knowledge base construction (Roth et al., 2015) and essential for enhancing downstream tasks such as relation extraction (Yaghoobzadeh et al., 2017b), question answering (Das et al., 2017; Welbl et al., 2017) and search (Dalton et al., 2014). This has led to considerable research in automatically identifying entities in text, predicting their types, and linking them to existing structured knowledge sources.

Current state-of-the-art models encode a textual mention with a neural network and classify the mention as being an instance of a fine grained type or entity in a knowledge base. Although in many cases the types and their entities are arranged in a hierarchical ontology, most approaches ignore this structure, and previous attempts to incorporate hierarchical information yielded little improvement in performance (Shimaoka et al., 2017). Additionally, existing benchmark entity typing datasets only consider small label sets arranged in very shallow hierarchies. For example, FIGER (Ling and Weld, 2012), the *de facto* standard fine grained entity type dataset, contains only 113 types in a hierarchy only two levels deep.

In this paper we investigate models that explicitly integrate hierarchical information into the embedding space of entities and types, using a hierarchy-aware loss on top of a deep neural network classifier over textual mentions. By using this additional information, we learn a richer, more robust representation, gaining statistical efficiency when predicting similar concepts and aiding the classification of rarer types. We first validate our methods on the narrow, shallow type system of FIGER, out-performing state-of-the-art methods not incorporating hand-crafted features and matching those that do.

To evaluate on richer datasets and stimulate further research into hierarchical entity/typing prediction with larger and deeper ontologies, we introduce two new human annotated datasets. The first is *MedMentions*, a collection of PubMed ab-

*equal contribution

Data and code for experiments: https://github.com/MurtyShikhar/Hierarchical-Typing

stracts in which 246k concept mentions have been annotated with links to the Unified Medical Language System (UMLS) ontology (Bodenreider, 2004), an order of magnitude more annotations than comparable datasets. UMLS contains over 3.5 million concepts in a hierarchy having average depth 14.4. Interestingly, UMLS does not distinguish between types and entities (an approach we heartily endorse), and the technical details of linking to such a massive ontology lead us to refer to our MedMentions experiments as entity linking. Second, we present *TypeNet*, a curated mapping from the Freebase type system into the WordNet hierarchy. TypeNet contains over 1900 types with an average depth of 7.8.

In experimental results, we show improvements with a hierarchically-aware training loss on each of the three datasets. In entity-linking MedMentions to UMLS, we observe a 6% relative increase in accuracy over the base model. In experiments on entity-typing from Wikipedia into TypeNet, we show that incorporating the hierarchy of types and including a hierarchical loss provides a dramatic 29% relative increase in MAP. Our models even provide benefits for shallow hierarchies allowing us to match the state-of-art results of Shimaoka et al. (2017) on the FIGER (GOLD) dataset without requiring hand-crafted features.

We will publicly release the TypeNet and MedMentions datasets to the community to encourage further research in truly fine-grained, hierarchical entity-typing and linking.

2 New Corpora and Ontologies

2.1 MedMentions

Over the years researchers have constructed many large knowledge bases in the biomedical domain (Apweiler et al., 2004; Davis et al., 2008; Chatr-aryamontri et al., 2017). Many of these knowledge bases are specific to a particular sub-domain encompassing a few particular types such as genes and diseases (Piñero et al., 2017).

UMLS (Bodenreider, 2004) is particularly comprehensive, containing over 3.5 million concepts (UMLS does not distinguish between entities and types) defining their relationships and a curated hierarchical ontology. For example *LETM1 Protein* IS-A *Calcium Binding Protein* IS-A *Binding Protein* IS-A *Protein* IS-A *Genome Encoded Entity*. This fact makes UMLS particularly well suited for methods explicitly exploiting hierarchical structure.

Accurately linking textual biological entity mentions to an existing knowledge base is extremely important but few richly annotated resources are available. Even when resources do exist, they often contain no more than a few thousand annotated entity mentions which is insufficient for training state-of-the-art neural network entity linkers. State-of-the-art methods must instead rely on string matching between entity mentions and canonical entity names (Leaman et al., 2013; Wei et al., 2015; Leaman and Lu, 2016). To address this, we constructed MedMentions, a new, large dataset identifying and linking entity mentions in PubMed abstracts to specific UMLS concepts. Professional annotators exhaustively annotated UMLS entity mentions from 3704 PubMed abstracts, resulting in 246,000 linked mention spans. The average depth in the hierarchy of a concept from our annotated set is 14.4 and the maximum depth is 43.

MedMentions contains an order of magnitude more annotations than similar biological entity linking PubMed datasets (Doğan et al., 2014; Wei et al., 2015; Li et al., 2016). Additionally, these datasets contain annotations for only one or two entity types (genes or chemicals and disease etc.). MedMentions instead contains annotations for a wide diversity of entities linking to UMLS. Statistics for several other datasets are in Table 1 and further statistics are in 2.

Dataset	mentions	unique entities
MedMentions	246,144	25,507
BCV-CDR	28,797	2,356
NCBI Disease	6,892	753
BCII-GN Train	6,252	1,411
NLM Citation GIA	1,205	310

Table 1: Statistics from various biological entity linking data sets from scientific articles. NCBI Disease (Doğan et al., 2014) focuses exclusively on disease entities. BCV-CDR (Li et al., 2016) contains both chemicals and diseases. BCII-GN and NLM (Wei et al., 2015) both contain genes.

Statistic	Train	Dev	Test
#Abstracts	2,964	370	370
#Sentences	28,457	3,497	3,268
#Mentions	199,977	24,026	22,141
#Entities	22,416	5,934	5,521

Table 2: MedMentions statistics.

2.2 TypeNet

TypeNet is a new dataset of hierarchical entity types for extremely fine-grained entity typing. TypeNet was created by manually aligning Freebase types (Bollacker et al., 2008) to noun synsets from the WordNet hierarchy (Fellbaum, 1998), naturally producing a hierarchical type set.

To construct TypeNet, we first consider all Freebase types that were linked to more than 20 entities. This is done to eliminate types that are either very specific or very rare. We also remove all Freebase API types, e.g. the [/freebase, /dataworld, /schema, /atom, /scheme, and /topics] domains.

For each remaining Freebase type, we generate a list of candidate WordNet synsets through a substring match. An expert annotator then attempted to map the Freebase type to one or more synsets in the candidate list with a *parent-of*, *child-of* or *equivalence* link by comparing the definitions of each synset with example entities of the Freebase type. If no match was found, the annotator manually formulated queries for the online WordNet API until an appropriate synset was found. See Table 9 for an example annotation.

Two expert annotators independently aligned each Freebase type before meeting to resolve any conflicts. The annotators were conservative with assigning equivalence links resulting in a greater number of *child-of* links. The final dataset contained 13 *parent-of*, 727 *child-of*, and 380 *equivalence* links. Note that some Freebase types have multiple *child-of* links to WordNet, making TypeNet, like WordNet, a directed acyclic graph. We then took the union of each of our annotated Freebase types, the synset that they linked to, and any ancestors of that synset.

Typeset	Count	Depth	Gold KB links
CoNLL-YAGO	4	1	Yes
OntoNotes 5.0	19	1	No
Gillick et al. (2014)	88	3	Yes
Figer	112	2	Yes
Hyena	505	9	No
Freebase	2k	2	Yes
WordNet	16k	14	No
TypeNet*	1,941	14	Yes

Table 3: Statistics from various type sets. TypeNet is the largest type hierarchy with a gold mapping to KB entities. *The entire WordNet could be added to TypeNet increasing the total size to 17k types.

We also added an additional set of 614 *FB* $\rightarrow$ *FB* links 4. This was done by computing conditional probabilities of Freebase types given other Freebase types from a collection of 5 million randomly chosen Freebase entities. The conditional probability P($t_2 \mid t_1$) of a Freebase type t_2 given another Freebase type t_1 was calculated as $\frac{\#(t_1,t_2)}{\#t_1}$. Links with a conditional probability less than or equal to 0.7 were discarded. The remaining links were manually verified by an expert annotator and valid links were added to the final dataset, preserving acyclicity.

Freebase Types	1081
WordNet Synsets	860
child-of links	727
equivalence links	380
parent-of links	13
Freebase-Freebase links	614

Table 4: Stats for the final TypeNet dataset. child-of, parent-of, and equivalence links are from Freebase types $\rightarrow$ WordNet synsets.

3 Model

3.1 Background: Entity Typing and Linking

We define a textual mention m as a sentence with an identified entity. The goal is then to classify m with one or more labels. For example, we could take the sentence m = "*Barack Obama is the President of the United States.*" with the identified entity string **Barack Obama**. In the task of *entity linking*, we want to map m to a specific entity in a knowledge base such as "m/02mjmr" in Freebase. In *mention-level typing*, we label m with one or more types from our type system T such as t^m = {president, leader, politician} (Ling and Weld, 2012; Gillick et al., 2014; Shimaoka et al., 2017). In *entity-level typing*, we instead consider a bag of mentions B_e which are all linked to the same entity. We label B_e with t^e, the set of all types expressed in all $m \in B_e$ (Yao et al., 2013; Neelakantan and Chang, 2015; Verga et al., 2017; Yaghoobzadeh et al., 2017a).

3.2 Mention Encoder

Our model converts each mention m to a d dimensional vector. This vector is used to classify the type or entity of the mention. The basic model depicted in Figure 1 concatenates the averaged word embeddings of the mention string with the output of a convolutional neural network (CNN). The

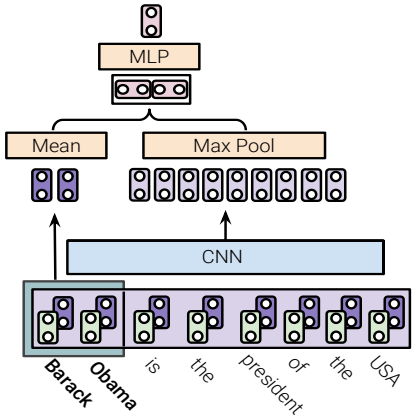

Figure 1: Sentence encoder for all our models. The input to the CNN consists of the concatenation of position embeddings with word embeddings. The output of the CNN is concatenated with the mean of mention surface form embeddings, and then passed through a 2 layer MLP.

word embeddings of the mention string capture global, context independent semantics while the CNN encodes a context dependent representation.

3.2.1 Token Representation

Each sentence is made up of s tokens which are mapped to d_w dimensional word embeddings. Because sentences may contain mentions of more than one entity, we explicitly encode a distinguished mention in the text using position embeddings which have been shown to be useful in state of the art relation extraction models (dos Santos et al., 2015; Lin et al., 2016) and machine translation (Vaswani et al., 2017). Each word embedding is concatenated with a d_p dimensional learned position embedding encoding the token's relative distance to the target entity. Each token within the distinguished mention span has position 0, tokens to the left have a negative distance from $[-s, 0)$, and tokens to the right of the mention span have a positive distance from $(0, s]$. We denote the final sequence of token representations as M.

3.2.2 Sentence Representation

The embedded sequence M is then fed into our context encoder. Our context encoder is a single layer CNN followed by a $\tanh$ non-linearity to produce C. The outputs are max pooled across time to get a final context embedding, m_{CNN}.

$$c_i = \tanh(b + \sum_{j=0}^{w} W[j]M[i - \lfloor \frac{w}{2} \rfloor + j])$$

$$m_{\text{CNN}} = \max_{0 \le i \le n-w+1} c_i$$

Each $W[j] \in \mathbb{R}^{d \times d}$ is a CNN filter, the bias $b \in \mathbb{R}^d$, $M[i] \in \mathbb{R}^d$ is a token representation, and the max is taken pointwise. In all of our experiments we set $w = 5$.

In addition to the contextually encoded mention, we create a global mention encoding, m_{G}, by averaging the word embeddings of the tokens within the mention span.

The final mention representation m_{F} is constructed by concatenating m_{CNN} and m_{G} and applying a two layer feed-forward network with $\tanh$ non-linearity (see Figure 1):

$$m_{\text{F}} = W_2 \tanh(W_1 \begin{bmatrix} m_{\text{SFM}} \\ m_{\text{CNN}} \end{bmatrix} + b_1) + b_2$$

4 Training

4.1 Mention-Level Typing

Mention level entity typing is treated as multi-label prediction. Given the sentence vector m_{F}, we compute a score for each type in typeset T as:

$$y_j = \mathbf{t_j}^\top m_{\text{F}}$$

where $\mathbf{t_j}$ is the embedding for the j$^{\text{th}}$ type in T and y_j is its corresponding score. The mention is labeled with t^m, a binary vector of all types where $t^m_j = 1$ if the j$^{\text{th}}$ type is in the set of gold types for m and 0 otherwise. We optimize a multi-label binary cross entropy objective:

$$\mathcal{L}_{\text{type}}(m) = -\sum_j t^m_j \log y_j + (1 - t^m_j) \log(1 - y_j)$$

4.2 Entity-Level Typing

In the absence of mention-level annotations, we instead must rely on distant supervision (Mintz et al., 2009) to noisily label all mentions of entity e with all types belonging to e. This procedure inevitably leads to noise as not all mentions of an entity express each of its known types. To alleviate this noise, we use multi-instance multi-label learning (MIML) (Surdeanu et al., 2012) which operates over *bags* rather than mentions. A bag of mentions $B_e = \{m^1, m^2, \ldots, m^n\}$ is the set of

all mentions belonging to entity e. The bag is labeled with t^e, a binary vector of all types where $t^e_j = 1$ if the j[th] type is in the set of gold types for e and 0 otherwise.

For every entity, we subsample k mentions from its bag of mentions. Each mention is then encoded independently using the model described in Section 3.2 resulting in a bag of vectors. Each of the k sentence vectors m^i_{F} is used to compute a score for each type in t^e:

$$y^i_j = \mathbf{t_j}^\top m^i_{\text{F}}$$

where $\mathbf{t_j}$ is the embedding for the j[th] type in t^e and y^i is a vector of logits corresponding to the i[th] mention. The final bag predictions are obtained using element-wise LogSumExp pooling across the k logit vectors in the bag to produce entity level logits y:

$$y = \log \sum_i \exp(y^i)$$

We use these final bag level predictions to optimize a multi-label binary cross entropy objective:

$$\mathcal{L}_{\text{type}}(B_e) = -\sum_j t^e_j \log y_j + (1 - t^e_j) \log(1 - y_j)$$

4.3 Entity Linking

Entity linking is similar to mention-level entity typing with a single correct class per mention. Because the set of possible entities is in the millions, linking models typically integrate an alias table mapping entity mentions to a set of possible candidate entities. Given a large corpus of entity linked data, one can compute conditional probabilities from mention strings to entities (Spitkovsky and Chang, 2012). In many scenarios this data is unavailable. However, knowledge bases such as UMLS contain a canonical string name for each of its curated entities. State-of-the-art biological entity linking systems tend to operate on various string edit metrics between the entity mention string and the set of canonical entity strings in the existing structured knowledge base (Leaman et al., 2013; Wei et al., 2015).

For each mention in our dataset, we generate 100 candidate entities $e_c = (e_1, e_2, \ldots, e_{100})$ each with an associated string similarity score csim. See Appendix A.5.1 for more details on candidate generation. We generate the sentence representation m_{F} using our encoder and compute a similarity score between m_{F} and the learned embedding e of each of the candidate entities. This score and string cosine similarity csim are combined via a learned linear combination to generate our final score. The final prediction at test time $\hat{e}$ is the maximally similar entity to the mention.

$$\phi(m, e) = \alpha\, e^\top m_{\text{F}} + \beta\, \text{csim}(m, e)$$
$$\hat{e} = \underset{e \in e_c}{\text{argmax}}\ \phi(m, e)$$

We optimize this model by multinomial cross entropy over the set of candidate entities and correct entity e.

$$\mathcal{L}_{\text{link}}(m, e_c) = -\phi(m, e) + \log \sum_{e' \in e_c} \exp \phi(m, e')$$

5 Encoding Hierarchies

Both entity typing and entity linking treat the label space as prediction into a flat set. To explicitly incorporate the structure between types/entities into our training, we add an additional loss. We consider two methods for modeling the hierarchy of the embedding space: real and complex bilinear maps, which are two of the state-of-the-art knowledge graph embedding models.

5.1 Hierarchical Structure Models

Bilinear: Our standard bilinear model scores a hypernym link between (c_1, c_2) as:

$$s(c_1, c_2) = {c_1}^\top A c_2$$

where $A \in \mathbb{R}^{d \times d}$ is a learned real-valued non-diagonal matrix and c_1 is the child of c_2 in the hierarchy. This model is equivalent to RESCAL (Nickel et al., 2011) with a single IS-A relation type. The type embeddings are the same whether used on the left or right side of the relation. We merge this with the base model by using the parameter A as an additional map before type/entity scoring.

Complex Bilinear: We also experiment with a complex bilinear map based on the ComplEx model (Trouillon et al., 2016), which was shown to have strong performance predicting the hypernym relation in WordNet, suggesting suitability for asymmetric, transitive relations such as those in our type hierarchy. ComplEx uses complex valued vectors for types, and diagonal complex matrices for relations, using Hermitian inner products (taking the complex conjugate of the second argument, equivalent to treating the right-hand-side

type embedding to be the complex conjugate of the left hand side), and finally taking the real part of the score[1]. The score of a hypernym link between (c_1, c_2) in the ComplEx model is defined as:

$$\begin{aligned} s(c_1, c_2) &= \text{Re}(< c_1, r_{\text{IS-A}}, c_2 >) \\ &= \text{Re}(\sum_k c_{1k} r_k \bar{c}_{2k}) \\ &= \langle \text{Re}(c_1), \text{Re}(r_{\text{IS-A}}), \text{Re}(c_2) \rangle \\ &+ \langle \text{Re}(c_1), \text{Im}(r_{\text{IS-A}}), \text{Im}(c_2) \rangle \\ &+ \langle \text{Im}(c_1), \text{Re}(r_{\text{IS-A}}), \text{Im}(c_2) \rangle \\ &- \langle \text{Im}(c_1), \text{Im}(r_{\text{IS-A}}), \text{Re}(c_2) \rangle \end{aligned}$$

where c_1, c_2 and $r_{\text{IS-A}}$ are complex valued vectors representing c_1, c_2 and the IS-A relation respectively. $\text{Re}(z)$ represents the real component of z and $\text{Im}(z)$ is the imaginary component. As noted in Trouillon et al. (2016), the above function is antisymmetric when $r_{\text{IS-A}}$ is purely imaginary.

Since entity/type embeddings are complex vectors, in order to combine it with our base model, we also need to represent mentions with complex vectors for scoring. To do this, we pass the output of the mention encoder through two different affine transformations to generate a real and imaginary component:

$$\begin{aligned} \text{Re}(m_{\text{F}}) &= W_{\text{real}} m_{\text{F}} + b_{\text{real}} \\ \text{Im}(m_{\text{F}}) &= W_{\text{img}} m_{\text{F}} + b_{\text{img}} \end{aligned}$$

where m_{F} is the output of the mention encoder, and $W_{\text{real}}, W_{\text{img}} \in \mathbb{R}^{d \times d}$ and $b_{\text{real}}, b_{\text{img}} \in \mathbb{R}^d$.

5.2 Training with Hierarchies

Learning a hierarchy is analogous to learning embeddings for nodes of a knowledge graph with a single hypernym/IS-A relation. To train these embeddings, we sample (c_1, c_2) pairs, where each pair is a positive link in our hierarchy. For each positive link, we sample a set N of n negative links. We encourage the model to output high scores for positive links, and low scores for negative links via a binary cross entropy (BCE) loss:

$$\begin{aligned} \mathcal{L}_{\text{struct}} &= -\log \sigma(s(c_{1i}, c_{2i})) \\ &+ \sum_N \log(1 - \sigma(s(c_{1i}, c'_{2i}))) \\ \mathcal{L} &= \mathcal{L}_{\text{type/link}} + \gamma \mathcal{L}_{\text{struct}} \end{aligned}$$

[1]This step makes the scoring function technically not bilinear, as it commutes with addition but not complex multiplication, but we term it *bilinear* for ease of exposition.

where $s(c_1, c_2)$ is the score of a link (c_1, c_2), and $\sigma(\cdot)$ is the logistic sigmoid. The weighting parameter γ is $\in \{0.1, 0.5, 0.8, 1, 2.0, 4.0\}$. The final loss function that we optimize is $\mathcal{L}$.

6 Experiments

We perform three sets of experiments: mention-level entity typing on the benchmark dataset FIGER, entity-level typing using Wikipedia and TypeNet, and entity linking using MedMentions.

6.1 Models

CNN: Each mention is encoded using the model described in Section 3.2. The resulting embedding is used for classification into a flat set labels. Specific implementation details can be found in Appendix A.2.

CNN+Complex: The CNN+Complex model is equivalent to the CNN model but uses complex embeddings and Hermitian dot products.

Transitive: This model does not add an additional hierarchical loss to the training objective (unless otherwise stated). We add additional labels to each entity corresponding to the transitive closure, or the union of all ancestors of its known types. This provides a rich additional learning signal that greatly improves classification of specific types.

Hierarchy: These models add an explicit hierarchical loss to the training objective, as described in Section 5, using either complex or real-valued bilinear mappings, and the associated parameter sharing.

6.2 Mention-Level Typing in FIGER

To evaluate the efficacy of our methods we first compare against the current state-of-art models of Shimaoka et al. (2017). The most widely used type system for fine-grained entity typing is FIGER which consists of 113 types organized in a 2 level hierarchy. For training, we use the publicly available W2M data (Ren et al., 2016) and optimize the mention typing loss function defined in Section-4.1 with the additional hierarchical loss where specified. For evaluation, we use the manually annotated FIGER (GOLD) data by Ling and Weld (2012). See Appendix A.2 and A.3 for specific implementation details.

6.2.1 Results

In Table 5 we see that our base CNN models (CNN and CNN+Complex) match LSTM models of Shimaoka et al. (2017) and Gupta et al. (2017), the

Model	Acc	Macro F1	Micro F1
Ling and Weld (2012)	47.4	69.2	65.5
Shimaoka et al. (2017) †	55.6	75.1	71.7
Gupta et al. (2017)†	57.7	72.8	72.1
Shimaoka et al. (2017)‡	**59.6**	**78.9**	**75.3**
CNN	57.0	75.0	72.2
+ hierarchy	58.4	76.3	73.6
CNN+Complex	57.2	75.3	72.9
+ hierarchy	**59.7**	**78.3**	**75.4**

Table 5: Accuracy and Macro/Micro F1 on FIGER (GOLD). † is an LSTM model. ‡ is an attentive LSTM along with additional hand crafted features.

Model	Low Data	Full Data
CNN	51.72	68.15
+ hierarchy	54.82	75.56
+ transitive	57.68	77.21
+ hierarchy + transitive	58.74	**78.59**
CNN+Complex	50.51	69.83
+ hierarchy	55.30	72.86
+ transitive	53.71	72.18
+ hierarchy + transitive	**58.81**	77.21

Table 6: MAP of entity-level typing in Wikipedia data using TypeNet. The second column shows results using 5% of the total data. The last column shows results using the full set of 344,246 entities.

Model	original	normalized
mention tfidf	61.09	74.66
CNN	67.42	82.40
+ hierarchy	67.73	82.77
CNN+Complex	67.23	82.17
+ hierarchy	**68.34**	**83.52**

Table 7: Accuracy on entity linking in MedMentions. Maximum recall is 81.82% because we use an imperfect alias table to generate candidates. Normalized scores consider only mentions which contain the gold entity in the candidate set. Mention tfidf is $csim$ from Section 4.3.

previous state-of-the-art for models without hand-crafted features. When incorporating structure into our models, we gain 2.5 points of accuracy in our CNN+Complex model, matching the overall state of the art attentive LSTM that relied on hand-crafted features from syntactic parses, topic models, and character n-grams. The structure can help our model predict lower frequency types which is a similar role played by hand-crafted features.

6.3 Entity-Level Typing in TypeNet

Next we evaluate our models on entity-level typing in TypeNet using Wikipedia. For each entity, we follow the procedure outlined in Section 4.2. We predict labels for each instance in the entity's bag and aggregate them into entity-level predictions using $\mathrm{LogSumExp}$ pooling. Each type is assigned a predicted score by the model. We then rank these scores and calculate average precision for each of the types in the test set, and use these scores to calculate mean average precision (MAP). We evaluate using MAP instead of accuracy which is standard in large knowledge base link prediction tasks (Verga et al., 2017; Trouillon et al., 2016). These scores are calculated only over Freebase types, which tend to be lower in the hierarchy. This is to avoid artificial score inflation caused by trivial predictions such as 'entity.' See Appendix A.4 for more implementation details.

6.3.1 Results

Table 6 shows the results for entity level typing on our Wikipedia TypeNet dataset. We see that both the basic CNN and the CNN+Complex models perform similarly with the CNN+Complex model doing slightly better on the full data regime. We also see that both models get an improvement when adding an explicit hierarchy loss, even before adding in the transitive closure. The transitive closure itself gives an additional increase in performance to both models. In both of these cases, the basic CNN model improves by a greater amount than CNN+Complex. This could be a result of the complex embeddings being more difficult to optimize and therefore more susceptible to variations in hyperparameters. When adding in both the transitive closure and the explicit hierarchy loss, the performance improves further. We observe similar trends when training our models in a lower data regime with ~150,000 examples, or about 5% of the total data.

In all cases, we note that the baseline models that do not incorporate any hierarchical information (neither the transitive closure nor the hierarchy loss) perform ~9 MAP worse, demonstrating the benefits of incorporating structure information.

6.4 MedMentions Entity Linking with UMLS

In addition to entity typing, we evaluate our model's performance on an entity linking task using MedMentions, our new PubMed / UMLS dataset described in Section 2.1.

6.4.1 Results

Table 7 shows results for baselines and our proposed variant with additional hierarchical loss. None of these models incorporate transitive clo-

Tips and Pitfalls in **Direct Ligation** of Large Spontaneous Splenorenal Shunt during Liver Transplantation Patients with large spontaneous splenorenal shunt . . . **baseline:** Direct [Direct → General Modifier → Qualifier → Property or Attribute] **+hierarchy:** Ligature (correct) [Ligature → Surgical Procedures → medical treatment approach]
A novel approach for selective chemical functionalization and localized assembly of one-dimensional **nanostructures**. **baseline:** Structure [Structure → order or structure → general epistemology] **+hierarchy:** Nanomaterials (correct) [Nanomaterials → Nanoparticle Complex → Drug or Chemical by Structure]
Gcn5 is recruited onto the **il-2** promoter by interacting with the NFAT in T cells upon TCR stimulation . **baseline:** Interleukin-27 [Interleukin-27 → IL2 → Interleukin Gene] **+hierarchy:** IL2 Gene (correct) [IL2 Gene → Interleukin Gene]

Table 8: Example predictions from MedMentions. Each example shows the sentence with entity mention span in bold. **Baseline**, shows the predicted entity and its ancestors of a model not incorporating structure. Finally, **+hierarchy** shows the prediction and ancestors for a model which explicitly incorporates the hierarchical structure information.

sure information, due to difficulty incorporating it in our candidate generation, which we leave to future work. The *Normalized* metric considers performance only on mentions with an alias table hit; all models have 0 accuracy for mentions otherwise. We also report the overall score for comparison in future work with improved candidate generation. We see that incorporating structure information results in a 1.1% reduction in absolute error, corresponding to a ~6% reduction in relative error on this large-scale dataset.

Table 8 shows qualitative predictions for models with and without hierarchy information incorporated. Each example contains the sentence (with target entity in bold), predictions for the baseline and hierarchy aware models, and the ancestors of the predicted entity. In the first and second example, the baseline model becomes extremely dependent on TFIDF string similarities when the gold candidate is rare ($\leq$ 10 occurrences). This shows that modeling the structure of the entity hierarchy helps the model disambiguate rare entities. In the third example, structure helps the model understand the hierarchical nature of the labels and prevents it from predicting an entity that is overly specific (e.g predicting Interleukin-27 rather than the correct and more general entity IL2 Gene).

Note that, in contrast with the previous tasks, the complex hierarchical loss provides a significant boost, while the real-valued bilinear model does not. A possible explanation is that UMLS is a far larger/deeper ontology than even TypeNet, and the additional ability of complex embeddings to model intricate graph structure is key to realizing gains from hierarchical modeling.

7 Related Work

By directly linking a large set of mentions and typing a large set of entities with respect to a new ontology and corpus, and our incorporation of structural learning between the many entities and types in our ontologies of interest, our work draws on many different but complementary threads of research in information extraction, knowledge base population, and completion.

Our structural, hierarchy-aware loss between types and entities draws on research in Knowledge Base Inference such as Jain et al. (2018), Trouillon et al. (2016) and Nickel et al. (2011). Combining KB completion with hierarchical structure in knowledge bases has been explored in (Dalvi et al., 2015; Xie et al., 2016). Recently, Wu et al. (2017) proposed a hierarchical loss for text classification.

Linking mentions to a flat set of entities, often in Freebase or Wikipedia, is a long-standing task in NLP (Bunescu and Pasca, 2006; Cucerzan, 2007; Durrett and Klein, 2014; Francis-Landau et al., 2016). Typing of mentions at varying levels of granularity, from CoNLL-style named entity recognition (Tjong Kim Sang and De Meulder, 2003), to the more fine-grained recent approaches (Ling and Weld, 2012; Gillick et al., 2014; Shimaoka et al., 2017), is also related to our task. A few prior attempts to incorporate a very shallow hierarchy into fine-grained entity typing have not lead to significant or consistent improvements (Gillick et al., 2014; Shimaoka et al., 2017).

The knowledge base Yago (Suchanek et al., 2007) includes integration with WordNet and type hierarchies have been derived from its type system (Yosef et al., 2012). Del Corro et al. (2015) use manually crafted rules and patterns (Hearst patterns (Hearst, 1992), appositives, etc) to automati-

cally match entity types to Wordnet synsets.

Recent work has moved towards unifying these two highly related tasks by improving entity linking by simultaneously learning a fine grained entity type predictor (Gupta et al., 2017). Learning hierarchical structures or transitive relations between concepts has been the subject of much recent work (Vilnis and McCallum, 2015; Vendrov et al., 2016; Nickel and Kiela, 2017)

We draw inspiration from all of this prior work, and contribute datasets and models to address previous challenges in jointly modeling the structure of large-scale hierarchical ontologies and mapping textual mentions into an extremely fine-grained space of entities and types.

8 Conclusion

We demonstrate that explicitly incorporating and modeling hierarchical information leads to increased performance in experiments on entity typing and linking across three challenging datasets. Additionally, we introduce two new human-annotated datasets: MedMentions, a corpus of 246k mentions from PubMed abstracts linked to the UMLS knowledge base, and TypeNet, a new hierarchical fine-grained entity typeset an order of magnitude larger and deeper than previous datasets.

While this work already demonstrates considerable improvement over non-hierarchical modeling, future work will explore techniques such as Box embeddings (Vilnis et al., 2018) and Poincaré embeddings (Nickel and Kiela, 2017) to represent the hierarchical embedding space, as well as methods to improve recall in the candidate generation process for entity linking. Most of all, we are excited to see new techniques from the NLP community using the resources we have presented.

9 Acknowledgements

We thank Nicholas Monath, Haw-Shiuan Chang and Emma Strubell for helpful comments on early drafts of the paper. Creation of the MedMentions corpus is supported and managed by the Meta team at the Chan Zuckerberg Initiative. A pre-release of the dataset is available at http://github.com/chanzuckerberg/MedMentions. This work was supported in part by the Center for Intelligent Information Retrieval and the Center for Data Science, 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

Rolf Apweiler, Amos Bairoch, Cathy H Wu, Winona C Barker, Brigitte Boeckmann, Serenella Ferro, Elisabeth Gasteiger, Hongzhan Huang, Rodrigo Lopez, Michele Magrane, et al. 2004. Uniprot: the universal protein knowledgebase. *Nucleic acids research*, 32(suppl_1):D115–D119.

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

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.

Razvan C Bunescu and Marius Pasca. 2006. Using encyclopedic knowledge for named entity disambiguation. In *Eacl*, volume 6, pages 9–16.

Andrew Chatr-aryamontri, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, Nadine K Kolas, Lara O'Donnell, Sara Oster, Chandra Theesfeld, Adnane Sellam, et al. 2017. The biogrid interaction database: 2017 update. *Nucleic acids research*, 45(D1):D369–D379.

Silviu Cucerzan. 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)*.

Jeffrey Dalton, Laura Dietz, and James Allan. 2014. Entity query feature expansion using knowledge base links. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 365–374. ACM.

Bhavana Dalvi, Einat Minkov, Partha P Talukdar, and William W Cohen. 2015. Automatic gloss finding for a knowledge base using ontological constraints. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 369–378. ACM.

Rajarshi Das, Manzil Zaheer, Siva Reddy, and Andrew McCallum. 2017. Question answering on knowledge bases and text using universal schema and memory networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 2: Short Papers), pages 358–365, Vancouver, Canada. Association for Computational Linguistics.

Allan Peter Davis, Cynthia G Murphy, Cynthia A Saraceni-Richards, Michael C Rosenstein, Thomas C Wiegers, and Carolyn J Mattingly. 2008. Comparative toxicogenomics database: a knowledgebase and discovery tool for chemical–gene–disease networks. *Nucleic acids research*, 37(suppl_1):D786–D792.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Greg Durrett and Dan Klein. 2014. A joint model for entity analysis: Coreference, typing, and linking. *Transactions of the Association for Computational Linguistics*, 2:477–490.

Christiane Fellbaum. 1998. *WordNet*. Wiley Online Library.

Matthew Francis-Landau, Greg Durrett, and Dan Klein. 2016. Capturing semantic similarity for entity linking with convolutional neural networks. In *Proceedings of NAACL-HLT*, pages 1256–1261.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *CoRR*, abs/1412.1820.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2671–2680, Copenhagen, Denmark. Association for Computational Linguistics.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

Prachi Jain, Shikhar Murty, Mausam, and Soumen Chakrabarti. 2018. Mitigating the effect of out-of-vocabulary entity pairs in matrix factorization for knowledge base inference. In *The 27th International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

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

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

Jiao Li, Yueping Sun, Robin J Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J Mattingly, Thomas C Wiegers, and Zhiyong Lu. 2016. Biocreative v cdr task corpus: a resource for chemical disease relation extraction. *Database*, 2016.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, Berlin, Germany. Association for Computational Linguistics.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Edward Loper and Steven Bird. 2002. Nltk: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 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*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Arvind Neelakantan and Ming-Wei Chang. 2015. Inferring missing entity type instances for knowledge base completion: New dataset and methods. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 515–525, Denver, Colorado. Association for Computational Linguistics.

Maximilian Nickel and Douwe Kiela. 2017. Poincar\'e embeddings for learning hierarchical representations. *arXiv preprint arXiv:1705.08039*.

Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Janet Piñero, Àlex Bravo, Núria Queralt-Rosinach, Alba Gutiérrez-Sacristán, Jordi Deu-Pons, Emilio Centeno, Javier García-García, Ferran Sanz, and Laura I Furlong. 2017. Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic acids research*, 45(D1):D833–D839.

Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 1825–1834.

Benjamin Roth, Nicholas Monath, David Belanger, Emma Strubell, Patrick Verga, and Andrew McCallum. 2015. Building knowledge bases with universal schema: Cold start and slot-filling approaches.

Cícero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing ACL*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2017. Neural architectures for fine-grained entity type classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1271–1280, Valencia, Spain. Association for Computational Linguistics.

Valentin I Spitkovsky and Angel X Chang. 2012. A cross-lingual dictionary for english wikipedia concepts.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: a core of semantic knowledge. In *Proceedings of the International Conference on World Wide Web (WWW)*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 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*, pages 455–465. Association for Computational Linguistics.

Erik F Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147. Association for Computational Linguistics.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the International Conference on Machine Learning (ICML)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Conference on Advances in Neural Information Processing (NIPS)*.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. *ICLR*.

Patrick Verga, Arvind Neelakantan, and Andrew McCallum. 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, Long Papers*, pages 613–622, Valencia, Spain. Association for Computational Linguistics.

Luke Vilnis, Xiang Li, Shikhar Murty, and Andrew McCallum. 2018. Probabilistic embedding of knowledge graphs with box lattice measures. In *The 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, Melbourne, Australia.

Luke Vilnis and Andrew McCallum. 2015. Word representations via gaussian embedding. *ICLR*.

Chih-Hsuan Wei, Hung-Yu Kao, and Zhiyong Lu. 2015. Gnormplus: an integrative approach for tagging genes, gene families, and protein domains. *BioMed research international*, 2015.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2017. Constructing datasets for multi-hop reading comprehension across documents. *arXiv preprint arXiv:1710.06481*.

Cinna Wu, Mark Tygert, and Yann LeCun. 2017. Hierarchical loss for classification. *CoRR*, abs/1709.01062.

Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2016. Representation learning of knowledge graphs with hierarchical types. In *IJCAI*, pages 2965–2971.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017a. Corpus-level fine-grained entity typing. *arXiv preprint arXiv:1708.02275*.

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017b. Noise mitigation for neural entity typing and relation extraction. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1183–1194, Valencia, Spain. Association for Computational Linguistics.

Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 79–84. ACM.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *Proceedings of the International Conference on Computational Linguistics (COLING)*.

A Supplementary Materials

A.1 TypeNet Construction

Freebase type: musical_chord
Example entities: psalms_chord, power_chord harmonic_seventh_chord
chord.n.01: a straight line connecting two points on a curve
chord.n.02: a combination of three or more notes that blend harmoniously when sounded together
musical.n.01: a play or film whose action and dialogue is interspersed with singing and dancing

Table 9: Example given to TypeNet annotators. Here, the Freebase type to be linked is musical_chord. This type is annotated in Freebase belonging to the entities psalms_chord, harmonic_seventh_chord, and power_chord. Below the list of example entities are candidate WordNet synsets obtained by substring matching between the Freebase type and all WordNet synsets. The correctly aligned synset is chord.n.02 shown in bold.

A.2 Model Implementation Details

For all of our experiments, we use pretrained 300 dimensional word vectors from Pennington et al. (2014). These embeddings are fixed during training. The type vectors and entity vectors are all 300 dimensional vectors initialized using Glorot initialization (Glorot and Bengio, 2010). The number of negative links for hierarchical training $n \in \{16, 32, 64, 128, 256\}$.

For regularization, we use dropout (Srivastava et al., 2014) with p $\in \{0.5, 0.75, 0.8\}$ on the sentence encoder output and L2 regularize all learned parameters with $\lambda \in \{$1e-5, 5e-5, 1e-4$\}$. All our parameters are optimized using Adam (Kingma and Ba, 2014) with a learning rate of 0.001. We tune our hyper-parameters via grid search and early stopping on the development set.

A.3 FIGER Implementation Details

To train our models, we use the mention typing loss function defined in Section-5. For models with structure training, we additionally add in the hierarchical loss, along with a weight that is obtained by tuning on the dev set. We follow the same inference time procedure as Shimaoka et al. (2017) For each mention, we first assign the type with the largest probability according to the logits, and then assign additional types based on the condition that their corresponding probability be greater than 0.5.

A.4 Wikipedia Data and Implementation Details

At train time, each training example randomly samples an entity bag of 10 mentions. At test time we classify bags of 20 mentions of an entity. The dataset contains a total of 344,246 entities mapped to the 1081 Freebase types from TypeNet. We consider all sentences in Wikipedia between 10 and 50 tokens long. Tokenization and sentence splitting was performed using NLTK (Loper and Bird, 2002). From these sentences, we considered all entities annotated with a cross-link in Wikipedia that we could link to Freebase and assign types in TypeNet. We then split the data by entities into a 90-5-5 train, dev, test split.

A.5 UMLS Implementation details

We pre-process each string by lowercasing and removing stop words. We consider ngrams from size 1 to 5 and keep the top 100,000 features and the final vectors are L2 normalized. For each mention, In our experiments we consider the top 100 most similar entities as the candidate set.

A.5.1 Candidate Generation Details

Each mention and each canonical entity string in UMLS are mapped to TFIDF character ngram vectors. We pre-process each string by lowercasing and removing stop words. We consider ngrams from size 1 to 5 and keep the top 100,000 features and the final vectors are L2 normalized. For each mention, we calculate the cosine similarity, csim, between the mention string and each canonical entity string. In our experiments we consider the top 100 most similar entities as the candidate set.