Finer Grained Entity Typing with TypeNet

Shikhar Murty, Patrick Verga, Luke Vilnis, Andrew McCallum
College of Information and Computer Sciences
University of Massachusetts Amherst
`{smurty, pat, luke, mccallum}@cs.umass.edu`

Abstract

We consider the challenging problem of entity typing over an extremely fine grained set of types, wherein a single mention or entity can have many simultaneous and often hierarchically-structured types. Despite the importance of the problem, there is a relative lack of resources in the form of fine-grained, deep type hierarchies aligned to existing knowledge bases. In response, we introduce TypeNet, a dataset of entity types consisting of over 1941 types organized in a hierarchy, obtained by manually annotating a mapping from 1081 Freebase types to WordNet. We also experiment with several models comparable to state-of-the-art systems and explore techniques to incorporate a structure loss on the hierarchy with the standard mention typing loss, as a first step towards future research on this dataset.

1 Introduction

Recognizing entities and their types is a core problem in natural language processing, underlying complex natural language understanding problems in relation extraction (Yaghoobzadeh et al., 2017), knowledge base construction, question answering (Lee et al., 2006), and query comprehension (Dalton et al., 2014). Early attempts at entity recognition focused only on very coarse grained types (Tjong Kim Sang and De Meulder, 2003; Hovy et al., 2006). More recently, there has been growing interest in models explicitly focused on entity typing with finer grained typesets e.g. FIGER (Ling and Weld, 2012).

The increasingly sophisticated natural language understanding tasks undertaken by the machine learning community often require commensurately more sophisticated world knowledge. This world knowledge, often organized hierarchically in *ontologies*, motivates our creation of a new fine-grained, deep, and high-quality dataset of hierarchical types.

Despite the increasing focus on fine-grained typing, existing typesets still contain only on the order of 100 different types. Further, these typesets are either endowed with only a shallow hierarchy, typically on the order of two levels deep or don't have links to existing KBs (see Table 1). In this work, we advocate for larger, deeper typesets and models that exploit the inherently hierarchical nature of these types. To this end, we present TypeNet, an expert-annotated type hierarchy containing 1941 individual types, with an average depth of 7.8.

We also evaluate several models for fine-grained entity typing, and establish a strong baseline of 74.8 MAP on the CoNLL-YAGO dataset (Hoffart et al., 2011) for our best model. With each entity having on the order of 30 types, there are clearly exciting opportunities for improvement from future research. Additionally, we investigate multi-task models that explicitly incorporate the hierarchical relations between types into the learning objective.
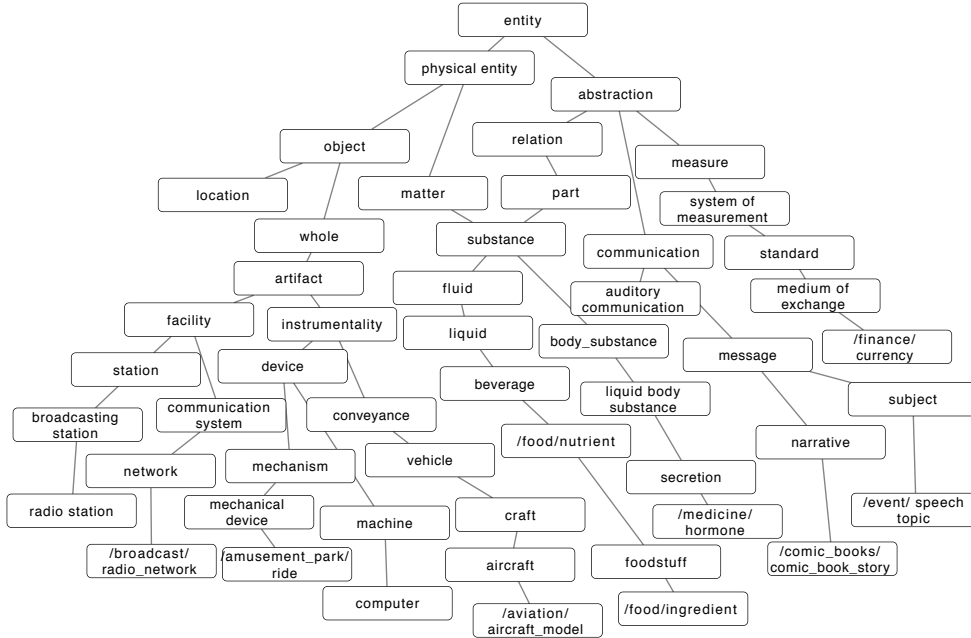
Figure 1: Sampled types in the TypeNet hierarchy

2 Dataset Creation

We now discuss TypeNet[1], a new dataset of entity types for extremely fine grained entity typing. TypeNet was created by manually aligning Freebase types to noun synsets from the WordNet hierarchy (Fellbaum, 1998), naturally producing a hierarchical type set.

This was done by first filtering out all Freebase types that were linked to ≥ 20 entities. We do this to eliminate extremely fine grained types that are very rare. We follow this by filtering Freebase API types. This includes types in the domains */freebase*,*/dataworld*,*/schema*, */atom*, */scheme* and */topics*.

For each Freebase type in our filtered set, we generate a list of candidate WordNet synsets through a substring match. The annotator then attempted to map the Freebase type to one or more synset in the candidate list with a *parent-of*, *child-of* or *equivalence* link by examining definitions of each synset and example entities of the Freebase type. If no match was found, the annotator queried the online WordNet API until an appropriate synset was found.

This procedure was carried out by two separate annotators independently after which conflicts were discussed and resolved. 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. Finally, all the ancestors of the Freebase types (following the manually created *child-of* and WordNet hypernym links) were added to construct the dataset.

We also carried out a procedure to add an additional set of 614 *FB → FB* links. 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 $\frac{\#(t_1, t_2)}{\#t_1}$. We then threshold these probabilities to 0.7, and manually filter the resulting links.

3 Model

We now describe the various neural models we use for our experiments.

---

[1]https://github.com/iesl/TypeNet

Typeset	Count	Depth	Gold KB links
CoNLL-YAGO	4	1	Yes
Ontonotes	19	1	No
Gillick et al. (2014)	93	3	Yes
Figer	121	2	Yes
Hyena	505	9	No
Freebase	2k	2	Yes
WordNet	16k	14	No
TypeNet	1941*	14	Yes

Table 1: 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.

Input Layer: We represent a mention $m = (\vec{m_1}, \vec{m_2}, ..., \vec{m_n})$ as a sequence of word vectors where each vector is of a fixed dimension d. To obtain a mention vector representation, we use a Convolutional Neural Network (CNN) based architecture. The CNN learns mention representations from sliding w-gram features of the mention. For a mention m with n tokens represented as vectors, the CNN outputs $n - w + 1$ vectors, which are then max-pooled along every dimension to obtain $\vec{m}_{\text{CNN}}$:

$$\vec{x_j} = \text{ReLU}(\vec{b} + \sum_{k=0}^{w} W[k :] \cdot \vec{m}_{[j - \frac{w}{2} + k]}) \tag{1}$$

$$\vec{m}_{\text{CNN}} = \max\{\vec{x}_1, \vec{x}_2, ..., \vec{x}_{n-w+1}\} \tag{2}$$

where W is a CNN filter of size $w \times d \times d$, and $\vec{b}$ is a bias vector of size d. We then concatenate $\vec{m}_{\text{CNN}}$ with another vector $\vec{m}_{\text{SFM}}$ obtained by averaging the surface form of the entity to which the mention links. We do this to provide our model explicit signal about the entity present in the mention. The concatenation is then passed through a series of affine, ReLU and affine transforms to obtain the final mention representation $\vec{m}$ (See Fig-2):

$$\vec{m}_{\text{SFM}} = \frac{1}{|t_2 - t_1 + 1|} \sum_{k=t_1}^{k=t_2} \vec{x_k} \tag{3}$$

$$\vec{m} = W_2(\text{ReLU}(W_1 \begin{bmatrix} \vec{m}_{\text{SFM}} \\ \vec{m}_{\text{CNN}} \end{bmatrix} + b_1) + b_2) \tag{4}$$

Loss Function: Like prior work (Shimaoka et al., 2017), we model entity typing as a multi-label problem, and for a given mention, produce a vector of scores corresponding to each type. We optimize a mention-typing loss over each minibatch B_1 of $(m, \{t_i\})$ pairs, where $\{t_i\}$ is the set of gold types annotated for the mention m:

$$\mathcal{L}_{\text{typing}} = \frac{1}{|B_1|} \sum_{(m, \{t_i\}) \in B_1} \left(\sum_{t \in \{t_i\}} -\text{score}(m \in t) \quad - \sum_{t' \notin \{t_i\}} \text{score}(m \notin t') \right) \tag{5}$$

where $\text{score}(m \in t)$ is some function indicating the compatibility score for mention m being in type t, using the interpretation of a type as a set defined by a unary predicate ("has type t").

In this work, we also introduce a structure loss among the types to incorporate the hierarchy. For this, we have a separate minibatch B_2 of $(t, \{t_a\})$ pairs, where $\{t_a\}$ is the set of ancestor types for the type t:

$$\mathcal{L}_{\text{structure}} = \frac{1}{|B_2|} \sum_{(t, \{t_i\}) \in B_2} \left(\sum_{t_a \in \{t_a\}} -\text{score}(t \in t_a) \quad - \sum_{t' \notin \{t_a\}} \text{score}(t \notin t') \right) \tag{6}$$

The exact scoring functions used for different models are summarized in Table 2, but are either variations of binary cross entropy or order embedding loss. We experiment with models whose loss is either $\mathcal{L}_{\text{typing}}$ alone, or a weighted combination of $\mathcal{L}_{\text{typing}}$ and $\mathcal{L}_{\text{structure}}$.
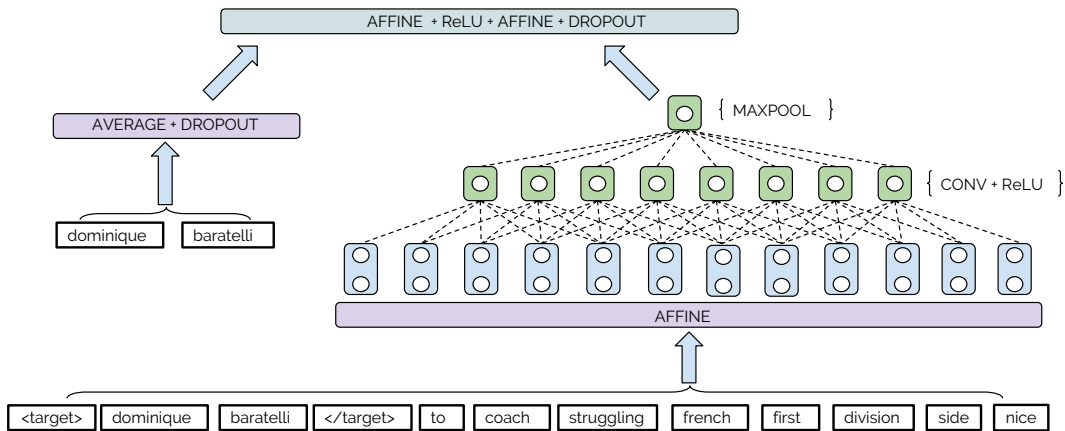
Figure 2: Mention Encoder

Model	score($x \in y$)	score($x \notin y$)
Order Embedding	$-\|\max(0, y-x)\|^2$	$\max(0, \alpha - \|\max(0, y-x)\|^2)$
Bilinear	$\log \sigma(x^\top A y)$	$\log(1 - \sigma(x^\top A y))$
Dot	$\log \sigma(x^\top y)$	$\log(1 - \sigma(x^\top y))$

Table 2: Scoring functions used for modeling type membership, where σ is the logistic sigmoid, and A is a learned parameter matrix (bilinear form) that scores the asymmetric membership relation.

Hyperparameters: We use pretrained 300 dimensional case sensitive GloVe vectors by Pennington et al. (2014) and a CNN with a filter width of 5. The type vectors are all 300 dimensional initialized using Glorot initialization (Glorot et al., 2011). We use dropout (Srivastava et al., 2014) as described in Fig-2 and optimize using Adam (Kingma and Ba, 2014). We tune our hyperparameters via grid search and early stopping on the development set.

4 Results

Dataset and Evaluation metrics: To perform our experiments, we use the CoNLL-YAGO (Hoffart et al., 2011) dev/test split and a subsampled version of Wikipedia (2016/09/20 dump) for training. We obtain labels for mentions via *distant supervision*, by assuming as positive types *all* the TypeNet types of the entity linked to a mention. We do not perform any heuristic pruning/denoising of these types, even though not all of them are relevant for a mention. This is because most pruning methods in literature are either harsh for extremely fine types (Gillick et al., 2014), or did not give an increase in performance (Shimaoka et al., 2017). However, we plan on improving this in future work and release a gold test data set.

To obtain the TypeNet types of an entity, we filter out all its Freebase types present in TypeNet, and finally for every type filtered out, add all its ancestors from TypeNet, giving us an average of 30.73 types per mention, *much* greater than earlier datasets such as FIGER (GOLD) (Ling and Weld, 2012) for which the average was 1.73 types per mention.

Since we have on average 30x more types per entities, we use Mean Average Precision (MAP) to measure performance unlike prior work on fine grained entity typing (Shimaoka et al., 2017). Our results are summarized in Table 3.

Discussion: We observe that the CNN encoder model works best, and multitasking mention typing with structure decreases performance if the structure is modeled using a dot product. This is expected since hypernymy is an asymmetric relation. Furthermore, modeling structure with a bilinear objective improves performance over a dot product objective but fails to perform better than the regular model. We believe this is due to the nature of our test data which is made up mostly of leaf type predictions

4

	Model	MAP
Order	Mention	50.2
	+structure	50.7
	CNN	44.0
	+structure	44.6
Bilinear	Mention	70.0
	+structure (dot)	69.5
	CNN	**74.8**
	+structure (dot)	73.5
	+structure (bilinear)	**74.8**

Table 3: Mean average precision for various models on TypeNet. "Mention" refers to simply averaging the words of the entity mention surface form. CNN concatenates the mention representations with a sentence representation. Scores with +structure additionally multi-task the structure loss training objective.

from a non-diverse set of types (people, locations, and organizations). A more diverse dataset requiring predictions at different depths of the hierarchy could use this structure more effectively, since we posit that without using structure, the model will incorrectly predict leaf types when only parent types are true.

Interestingly, we observe the order embeddings model from Vendrov et al. (2016) to have a poor performance for our task. We attribute this to the fact that the loss function is poorly suited to the problem since it uses unrelated concepts as negative examples, which in the traditional order embedding model actually implies a reversal of the parent-child relation, rather than simply forcing the types to be unrelated. For example, consider a hypernym link *person* $\Rightarrow$ *organism*, and a negative example, *person* $\Rightarrow$ *stadium*. The loss function from Vendrov et al. (2016) attempts to increase the order violation between *person* and *stadium*, making *stadium* a hyponym of *person*. We also observe particularly poor performance combining the order embeddings with the CNN encoder.

5   Related work

Table-1 summarizes existing hierarchical type systems including popular data sets such as FIGER (Ling and Weld, 2012) and Gillick et al. (2014).

Del Corro et al. (2015) considered the task of extremely fine grained entity typing. They use manually crafted rules and patterns (Hearst patters, appositives, etc). Hearst (1992) to extract candidate entity types that match Wordnet synsets. They apply an optional KB type filtering step for entity e by matching a candidate type t_c to any of e's KB types T if t_c is a string match of any t_i, $hypernym(t_i)$, or $hyponym(t_i)$. We instead manually annotated the exact mapping from 1081 Freebase types to the specific WordNet synset sense allowing us to leverage distant supervision to trained supervised classifiers.

The knowledge base Yago (Suchanek et al., 2008) includes integration with WordNet and type hierarchies have been derived from its type system (Yosef et al., 2012). However the links between entity types and WordNet types are performed heuristically whereas TypeNet contains gold links between Freebase and WordNet.

There has been a growing interest in learning representations of hierarchically organized objects. Vilnis and McCallum (2016) proposed Gaussian embeddings which learn containment properties of words by approximating them with Gaussian distributions. Vendrov et al. (2016) introduced order embeddings by minimizing an order violation loss. Recently Nickel and Kiela (2017) proposed Poincaré embeddings.

6   Conclusion and Future Work

We introduced TypeNet, a human labeled alignment between Freebase entity types and WordNet synsets. We used this typeset to distantly label the CoNLL-YAGO entity linking dataset and reported

initial results with several models comparable to the state-of-the-art models previously used on pre-existing datasets e.g. Shimaoka et al. (2017).

We additionally present results from models incorporating a structure loss over the type hierarchy, which does not appear to be required by the CoNLL-YAGO dataset, but should be helpful on more diverse datasets from different domains e.g. ClueWeb, which we will explore in future work and encourage the community to do the same.

We are exploring more sophisticated methods of incorporating the type hierarchy into the typing loss, as well as joint models for related tasks such as simultaneous typing and entity linking.

We are excited to see what the community will do with TypeNet, the largest and deepest entity type hierarchy with manual alignment to Freebase. We hope this will spur improvements in fine-grained entity and mention typing, linking and associated downstream tasks.

References

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*. ACM, pages 365–374.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, pages 868–878. http://aclweb.org/anthology/D15-1103.

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

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

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. pages 315–323.

Marti A Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics-Volume 2*. Association for Computational Linguistics, pages 539–545.

Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. 2011. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 782–792.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pages 57–60.

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

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Asia Information Retrieval Symposium*. Springer, pages 581–587.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition.

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

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*. pages 1532–1543. http://aclweb.org/anthology/D/D14/D14-1162.pdf.

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*. Association for Computational Linguistics, Valencia, Spain, pages 1271–1280. http://www.aclweb.org/anthology/E17-1119.

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* 15(1):1929–1958. http://dl.acm.org/citation.cfm?id=2670313.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2008. YAGO: A large ontology from wikipedia and wordnet. *J. Web Sem.* 6(3):203–217. https://doi.org/10.1016/j.websem.2008.06.001.

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*. Association for Computational Linguistics, pages 142–147.

Ivan Vendrov, Ryan Kiros, Sanja Fidler, and Raquel Urtasun. 2016. Order-embeddings of images and language. *Proceedings of the 2016 Conference on Learning Representations* .

Luke Vilnis and Andrew McCallum. 2016. Word representations via gaussian embedding. *Proceedings of the 2015 Conference on Learning Representations* .

Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. 2017. 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*. Association for Computational Linguistics, Valencia, Spain, pages 1183–1194. http://www.aclweb.org/anthology/E17-1111.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. *Proceedings of COLING 2012: Posters* pages 1361–1370.