

Fine-Grained Entity Typing in Hyperbolic Space

Federico López*

Benjamin Heinzerling

Michael Strube

*Research Training Group AIPHES

Heidelberg Institute for Theoretical Studies

firstname.lastname@h-its.org

Abstract

How can we represent hierarchical information present in large type inventories for entity typing? We study the ability of hyperbolic embeddings to capture hierarchical relations between mentions in context and their target types in a shared vector space. We evaluate on two datasets and investigate two different techniques for creating a large hierarchical entity type inventory: from an expert-generated ontology and by automatically mining type co-occurrences. We find that the hyperbolic model yields improvements over its Euclidean counterpart in some, but not all cases. Our analysis suggests that the adequacy of this geometry depends on the *granularity* of the type inventory and the way hierarchical relations are inferred.¹

1 Introduction

Entity typing classifies textual mentions of entities according to their semantic class. The task has progressed from finding company names (Rau, 1991), to recognizing coarse classes (*person*, *location*, *organization*, and *other*, Tjong Kim Sang and De Meulder, 2003), to fine-grained inventories of about one hundred types, with finer-grained types proving beneficial in applications such as relation extraction (Yaghoobzadeh et al., 2017) and question answering (Yavuz et al., 2016). The trend towards larger inventories has culminated in *ultra-fine* and *open* entity typing with thousands of classes (Choi et al., 2018; Zhou et al., 2018).

However, large type inventories pose a challenge for the common approach of casting entity typing as a multi-label classification task (Yogatama et al., 2015; Shimaoka et al., 2016), since exploiting inter-type correlations becomes more

¹Code available at: <https://github.com/nlpAThits/figet-hyperbolic-space>

Sentence	Annotation
...when the president said...	politician, president
...during the negotiation, he ...	politician, diplomat
...after the last meeting, she ...	politician, president
...the president argued...	politician, president

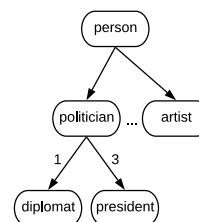


Figure 1: Examples of annotations and hierarchical type inventory with co-occurrence frequencies.

difficult as the number of types increases. A natural solution for dealing with a large number of types is to organize them in hierarchy ranging from general, *coarse* types such as “person” near the top, to more specific, *fine* types such as “politician” in the middle, to even more specific, *ultra-fine* entity types such as “diplomat” at the bottom (see Figure 1). By virtue of such a hierarchy, a model learning about diplomats will be able to transfer this knowledge to related entities such as politicians.

Prior work integrated hierarchical entity type information by formulating a hierarchy-aware loss (Ren et al., 2016; Murty et al., 2018; Xu and Barbosa, 2018) or by representing words and types in a joint Euclidean embedding space (Shimaoka et al., 2017; Abhishek et al., 2017). Noting that it is impossible to embed arbitrary hierarchies in Euclidean space, Nickel and Kiela (2017) propose hyperbolic space as an alternative and show that hyperbolic embeddings accurately encode hierarchical information. Intuitively (and as explained in more detail in Section 2), this is because distances in hyperbolic space grow exponentially as one moves away from the origin, just like the number of elements in a hierarchy grows exponentially with its depth.

While the intrinsic advantages of hyperbolic embeddings are well-established, their usefulness in downstream tasks is, so far, less clear. We be-

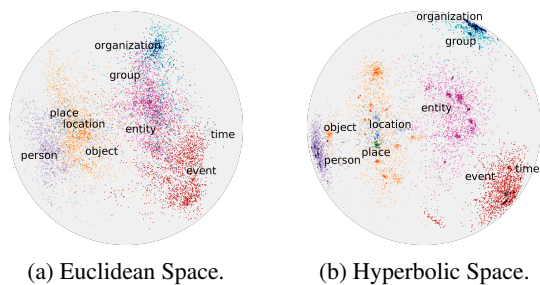


Figure 2: Type inventory of the Ultra-Fine dataset aligned to the WordNet noun hierarchy and projected on two dimensions in different spaces.

lieve this is due to two difficulties: First, incorporating hyperbolic embeddings into a neural model is non-trivial since training involves optimization in hyperbolic space. Second, it is often not clear what the best hierarchy for the task at hand is.

In this work, we address both of these issues. Using ultra-fine grained entity typing (Choi et al., 2018) as a test bed, we first show how to incorporate hyperbolic embeddings into a neural model (Section 3). Then, we examine the impact of the hierarchy, comparing hyperbolic embeddings of an expert-generated ontology to those of a large, automatically-generated one (Section 4). As our experiments on two different datasets show (Section 5), hyperbolic embeddings improve entity typing in some but not all cases, suggesting that their usefulness depends both on the type inventory and its hierarchy. In summary, we make the following contributions:

1. We develop a fine-grained entity typing model that embeds both entity types and entity mentions in hyperbolic space.
2. We compare two different entity type hierarchies, one created by experts (WordNet) and one generated automatically, and find that their adequacy depends on the dataset.
3. We study the impact of replacing the Euclidean geometry with its hyperbolic counterpart in an entity typing model, finding that the improvements of the hyperbolic model are noticeable on *ultra-fine* types.

2 Background: Poincaré Embeddings

Hyperbolic geometry studies non-Euclidean spaces with constant negative curvature. Two-dimensional hyperbolic space can be modelled as the open unit disk, the so-called *Poincaré disk*, in which the unit circle represents infinity, i.e., as a

point approaches infinity in hyperbolic space, its norm approaches one in the Poincaré disk model. In the general n -dimensional case, the disk model becomes the Poincaré ball (Chamberlain et al., 2017) $\mathcal{B}^n = \{x \in \mathbb{R}^n \mid \|x\| < 1\}$, where $\|\cdot\|$ denotes the Euclidean norm. In the Poincaré model the distance between two points $u, v \in \mathcal{B}^n$ is given by:

$$d_H(\mathbf{u}, \mathbf{v}) = \operatorname{arcosh}\left(1 + 2\frac{\|\mathbf{u} - \mathbf{v}\|^2}{(1 - \|\mathbf{u}\|^2)(1 - \|\mathbf{v}\|^2)}\right) \quad (1)$$

If we consider the origin O and two points, x and y , moving towards the outside of the disk, i.e. $\|x\|, \|y\| \rightarrow 1$, the distance $d_H(x, y)$ tends to $d_H(x, O) + d_H(O, y)$. That is, the path between x and y converges to a path through the origin. This behaviour can be seen as the continuous analogue to a (discrete) tree-like hierarchical structure, where the shortest path between two sibling nodes goes through their common ancestor.

As an alternative intuition, note that the hyperbolic distance between points grows exponentially as points move away from the center. This mirrors the exponential growth of the number of nodes in trees with increasing depths, thus making hyperbolic space a natural fit for representing trees and hence hierarchies (Krioukov et al., 2010; Nickel and Kiela, 2017).

By embedding hierarchies in the Poincaré ball so that items near the top of the hierarchy are placed near the origin and lower items near infinity (intuitively, embedding the “vertical” structure), and so that items sharing a parent in the hierarchy are close to each other (embedding the “horizontal” structure), we obtain Poincaré embeddings (Nickel and Kiela, 2017). More formally, this means that embedding norm represents depth in the hierarchy, and distance between embeddings the similarity of the respective items.

Figure 2 shows the results of embedding the WordNet noun hierarchy in two-dimensional Euclidean space (left) and the Poincaré disk (right). In the hyperbolic model, the types tend to be located near the boundary of the disk. In this region the space grows exponentially, which allows related types to be placed near one another and far from unrelated ones. The actual distance in this model is not the one visualized in the figure but the one given by Equation 1.

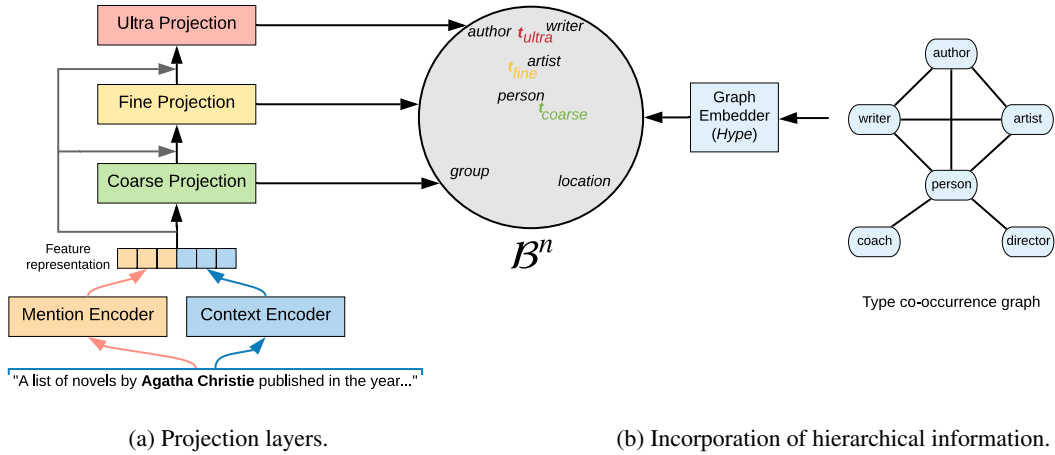


Figure 3: Overview of the proposed model to predict types of a mention within its context.

3 Entity Typing in Hyperbolic Space

3.1 Task Definition

The task we consider is, given a context sentence c containing an entity mention m , predict the correct type labels t_m that describe m from a type inventory T , which includes more than 10,000 types (Choi et al., 2018). The mention m can be a named entity, a nominal, or a pronoun. The ground-truth type set t_m may contain multiple types, making the task a multi-label classification problem.

3.2 Objective

We aim to analyze the effects of hyperbolic and Euclidean spaces when modeling hierarchical information present in the type inventory, for the task of fine-grained entity typing. Since hyperbolic geometry is naturally equipped to model hierarchical structures, we hypothesize that this enhanced representation will result in superior performance. With the goal of examining the relation between the metric space and the hierarchy, we propose a regression model. We learn a function that maps feature representations of a mention and its context onto a vector space such that the instances are embedded closer to their target types.

The ground-truth type set contains a varying number of types per instance. In our regression setup, however, we aim to predict a fixed amount of labels for all the instances. This imposes strong upper bounds to the performance of our proposed model. Nonetheless, as the strict accuracy of state-of-the-art methods for the Ultra-Fine dataset is below 40% (Choi et al., 2018; Xiong et al., 2019), the evaluation we perform is still informative in qualitative terms, and enables us to gain better intu-

itions with regard to embedding hierarchical structures in different metric spaces.

3.3 Method

Given the encoded feature representations of a mention m and its context c , noted as $e(m, c) \in \mathbb{R}^{n'}$ our goal is to learn a mapping function $f : \mathbb{R}^{n'} \rightarrow \mathcal{S}^n$, where \mathcal{S}^n is the target vector space. We intend to approximate embeddings of the type labels t_m , previously projected into the space. Subsequently, we perform a search of the nearest type embeddings of the embedded representation in order to assign the categorical label corresponding to the mention within that context. Figure 3 presents an overview of the model.

The label distribution on the dataset is diverse and fine-grained. Each instances is annotated with three levels of granularity, namely *coarse*, *fine* and *ultra-fine*, and on the development and test set there are, on average, five labels per item. This poses a challenging problem for learning and predicting with only one projection. As a solution, we propose three different projection functions, f_{coarse} , f_{fine} , and f_{ultra} , each one of them fine-tuned to predict labels of a specific granularity.

We hypothesize that the complexity of the projection increases as the granularity becomes *finer*, given that the target label space per granularity increases. Inspired by Sanh et al. (2019), we arrange the three projections in a hierarchical manner that reflects these difficulties. The *coarse* projection task is set at the bottom layer of the model and more complex (*finer*) interactions at higher layers. With the projected embedding of each layer, we aim to introduce an inductive bias in the next projection that will help to guide it into the correct

region of the space. Nevertheless, we use shortcut connections so that top layers can have access to the encoder layer representation.

3.4 Mention and Context Representations

To encode the context c containing the mention m , we apply the encoder schema of Choi et al. (2018) based on Shimaoka et al. (2016). We replace the location embedding of the original encoder with a word position embedding p_i to reflect relative distances between the i -th word and the entity mention. This modification induces a bias on the attention layer to focus less on the mention and more on the context. Finally we apply a standard Bi-LSTM and a self-attentive encoder (McCann et al., 2017) on top to get the context representation $C \in \mathbb{R}^{d_c}$.

For the mention representation we derive features from a character-level CNN, concatenate them with the Glove word embeddings (Pennington et al., 2014) of the mention, and combine them with a similar self-attentive encoder. The mention representation is denoted as $M \in \mathbb{R}^{d_m}$. The final representation is achieved by the concatenation of mention and context $[M; C] \in \mathbb{R}^{d_m+d_c}$.

3.5 Projecting into the Ball

To learn a projection function that embeds our feature representation in the target space, we apply a variation of the re-parameterization technique introduced in Dhingra et al. (2018). The re-parameterization involves computing a direction vector r and a norm magnitude λ from $e(m, c)$ as follows:

$$\begin{aligned} \bar{r} &= \varphi_{dir}(e(m, c)), & r &= \frac{\bar{r}}{\|\bar{r}\|}, \\ \bar{\lambda} &= \varphi_{norm}(e(m, c)), & \lambda &= \sigma(\bar{\lambda}), \end{aligned} \quad (2)$$

where $\varphi_{dir} : \mathbb{R}^{n'} \rightarrow \mathbb{R}^n$, $\varphi_{norm} : \mathbb{R}^{n'} \rightarrow \mathbb{R}$ can be arbitrary functions, whose parameters will be optimized during training, and σ is the sigmoid function that ensures the resulting norm $\lambda \in (0, 1)$. The re-parameterized embedding is defined as $v = \lambda r$, which lies in \mathcal{S}^n .

By making use of this simple technique, the embeddings are guaranteed to lie in the Poincaré ball. This avoids the need to correct the gradient or the utilization of Riemannian-SGD (Bonnabel, 2011). Instead, it allows the use of any optimization method in deep learning, such as Adam (Kingma and Ba, 2014).

We parameterize the direction function $\varphi_{dir} : \mathbb{R}^{d_m+d_c} \rightarrow \mathbb{R}^n$ as a multi-layer perceptron (MLP)

with a single hidden layer, using rectified linear units (ReLU) as nonlinearity, and dropout. We do not apply the ReLU function after the output layer in order to allow negative values as components of the direction vector. For the norm magnitude function $\varphi_{norm} : \mathbb{R}^{d_m+d_c} \rightarrow \mathbb{R}$ we use a single linear layer.

3.6 Optimization of the Model

We aim to find projection functions f_i that embed the instance representations closer to the respective target types, in a given vector space \mathcal{S}^n . As target space \mathcal{S}^n we use the Poincaré Ball \mathcal{B}^n and compare it with the Euclidean unit ball \mathbb{R}^n . Both \mathcal{B}^n and \mathbb{R}^n are metric spaces, therefore they are equipped with a distance function, namely the hyperbolic distance d_H defined in Equation 1, and the Euclidean distance d_E respectively, which we intend to minimize. Moreover, since the Poincaré Model is a conformal model of the hyperbolic space, *i.e.* the angles between Euclidean and hyperbolic vectors are equal, the cosine distance d_{\cos} can be used, as well.

We propose to minimize a combination of the distance defined by each metric space and the cosine distance to approximate the embeddings. Although formally this is not a distance metric since it does not satisfy the Cauchy-Schwarz inequality, it provides a very strong signal to approximate the target embeddings accounting for the main concepts modeled in the representation: *relatedness*, captured via the distance and orientation in the space, and *generality*, via the norm of the embeddings.

To mitigate the instability in the derivative of the hyperbolic distance² we follow the approach proposed in Sala et al. (2018) and minimize the square of the distance, which does have a continuous derivative in \mathcal{B}^n . Thus, in the Poincaré Model we minimize the distance for two points $u, v \in \mathcal{B}^n$ defined as:

$$d_{\mathcal{B}}(u, v) = \alpha(d_H(u, v))^2 + \beta d_{\cos}(u, v) \quad (3)$$

Whereas in the Euclidean space, for $x, y \in \mathbb{R}^n$ we minimize:

$$d_{\mathbb{R}}(x, y) = \alpha d_E(x, y) + \beta d_{\cos}(x, y) \quad (4)$$

The hyperparameters α and β are added to compensate the bounded image of the cosine distance function in $[0, 1]$.

² $\lim_{y \rightarrow x} \partial_x |d_H(x, y)| \rightarrow \infty \forall x \in \mathcal{B}^n$

Split	Coarse	Fine	Ultra-fine
Train	2,416,593	4,146,143	3,997,318
Dev	1,918	1,289	7,594
Test	1,904	1,318	7,511

Table 1: Type instances in the dataset grouped by split and granularity.

4 Hierarchical Type Inventories

In this section, we investigate two methods for deriving a hierarchical structure for a given type inventory. First, we introduce the datasets on which we perform our study since we exploit some of their characteristics to construct a hierarchy.

4.1 Data

We focus our analysis on the the Ultra-Fine entity typing dataset introduced in Choi et al. (2018). Its design goals were to increase the diversity and coverage entity type annotations. It contains 10,331 target types defined as free-form noun phrases and divided in three levels of granularity: *coarse*, *fine* and *ultra-fine*. The data consist of 6,000 crowdsourced examples and approximately 6M training samples in the open-source version³, automatically extracted with distant supervision, by entity linking and nominal head word extraction. Our evaluation is done on the original crowdsourced dev/test splits.

To gain a better understanding of the proposed model under different geometries, we also experiment on the OntoNotes dataset (Gillick et al., 2014) as it is a standard benchmark for entity typing.

4.2 Deriving the Hierarchies

The two methods we analyze to derive a hierarchical structure from the type inventory are the following.

Knowledge base alignment: Hierarchical information can be provided explicitly, by aligning the type labels to a knowledge base schema. In this case the types follow the tree-like structure of the ontology curated by experts. On the Ultra-Fine dataset, the type vocabulary T (i.e. noun phrases) is extracted from WordNet (Miller, 1992). Nouns in WordNet are organized into a deep hierarchy, defined by hypernym or “IS A” relationships. By aligning the type labels to the hypernym structure existing in WordNet, we obtain a type hierarchy. In this case, all paths lead to the root type *entity*.

³Choi et al. (2018) uses the licensed Gigaword to build part of the dataset resulting in about 25.2M training samples.

In the OntoNotes dataset the annotations follow a pre-established, much smaller, hierarchical taxonomy based on “IS A” relations, as well.

Type co-occurrences: Although in practical scenarios hierarchical information may not always be available, the distribution of types has an implicit hierarchy that can be inferred automatically. If we model the ground-truth labels as nodes of a graph, its adjacency matrix can be drawn and weighted by considering the co-occurrences on each instance. That is, if t_1 and t_2 are annotated as true types for a training instance, we add an edge between both types. To weigh the edge we explore two variants: the frequency of observed instances where this co-relation holds, and the *pointwise mutual information* (pmi), as a measure of the association between the two types⁴. By mining type co-occurrences present in the dataset as an affinity score, the hierarchy can be inferred. This method alleviates the need for a type inventory explicitly aligned to an ontology or pre-defined label correlations.

To embed the target type representations into the different metric spaces we make use of the library *Hype*⁵ (Nickel and Kiela, 2018). This library allows us to embed graphs into low-dimensional continuous spaces with different metrics, such as hyperbolic or Euclidean, ensuring that related objects are closer to each other in the space. The learned embeddings capture notions of both similarity, through the relative distance among each other, and hierarchy, through the distance to the origin, i.e. the norm. The projection of the hierarchy derived from WordNet is depicted in Figure 2.

5 Experiments

We perform experiments on the Ultra-Fine (Choi et al., 2018) and OntoNotes (Gillick et al., 2014) datasets to evaluate which kind of hierarchical information is better suited for entity typing, and under which geometry the hierarchy can be better exploited.

5.1 Setup

For evaluation we run experiments on the Ultra-Fine dataset with our model projecting onto the hyperbolic space, and compare to the same setting in Euclidean space. The type embeddings are

⁴We adapt pmi in order to satisfy the condition of non-negativity.

⁵<https://github.com/facebookresearch/poincare-embeddings/>

Model	Space	Coarse		Fine		Ultra-fine		Coarse + Ultra		Variation	
		MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1	MaF1	MiF1
MULTITASK	-	60.6	58.0	37.8	34.7	13.6	11.7	-	-	-	-
WORDNET	Hyper	45.9	44.3	22.5	21.5	7.0	6.7	41.8	37.2	-4.1	-7.1
	Euclid	56.1	54.2	26.6	25.3	7.2	6.5	56.6	48.5	0.6	-5.7
WORDNET + FREQ	Hyper	54.6	52.8	18.4	18.0	11.3	10.8	46.5	40.6	-8.0	-12.2
	Euclid	56.7	54.9	27.3	26.0	12.1	11.5	55.8	49.1	-0.9	-5.8
FREQ	Hyper	56.5	54.6	26.8	25.7	16.0	15.2	59.7	53.5	3.2	-1.1
	Euclid	56.1	54.2	25.8	24.4	12.1	11.4	60.0	53.0	3.9	-1.3
PMI	Hyper	54.7	53.0	26.9	25.8	16.0	15.4	57.5	51.8	2.8	-1.2
	Euclid	56.5	54.6	26.9	25.6	12.2	11.5	59.7	53.0	3.2	-1.5

(a) Results on the same three granularities analyzed by Choi et al. (2018).

(b) Comparison to previous *coarse* results.

Table 2: Results on the test set for different hierarchies and spaces. The best results of our models are marked in bold. On (b) we report the comparison of adding the closest *coarse* label to the *ultra-fine* prediction, with respect to the *coarse* results on (a).

created based on the following hierarchical structures derived from the dataset: the type vocabulary aligned to the WordNet hierarchy (WORDNET), type co-occurrence frequency (FREQ), *pointwise mutual information* among types (PMI), and finally, the combination of WordNet’s transitive closure of each type with the co-occurrence frequency graph (WORDNET + FREQ).

We compare our model to the multi-task model of Choi et al. (2018) trained on the open-source version of their dataset (MULTITASK). The final type predictions consist of the closest neighbor from the *coarse* and *fine* projections, and the three closest neighbors from the *ultra-fine* projection. We report Loose Macro-averaged and Loose Micro-averaged F1 metrics computed from the precision/recall scores over the same three granularities established by Choi et al. (2018). For all models we optimize Macro-averaged F1 on *coarse* types on the validation set, and evaluate on the test set. All experiments project onto a target space of 10 dimensions. The complete set of hyperparameters is detailed in the Appendix.

6 Results and Discussion

6.1 Comparison of the Hierarchies

Results on the test set are reported in Table 2. From comparing the different strategies to derive the hierarchies, we can see that FREQ and PMI substantially outperform MULTITASK on the *ultra-fine* granularity (17.5% and 29.8% relative improvement in Macro F1 and Micro F1, respectively, with the hyperbolic model). Both hierarchies show a substantially better performance over the WORDNET hierarchy on this granularity as

well (MaF1 16.0 and MiF1 15.4 for PMI vs 7.0 and 6.7 for WORDNET on the Hyperbolic model), indicating that these structures, created solely from the dataset statistics, better reflect the type distribution in the annotations. On FREQ and PMI, types that frequently co-occur on the training set are located closer to each other, improving the prediction based on nearest neighbor.

All the hierarchies show very low performance on *fine* when compared to the MULTITASK model. This exhibits a weakness of our regression setup. On the test set there are 1,998 instances but only 1,318 *fine* labels as ground truth (see Table 1). By forcing a prediction on the *fine* level for all instances, precision decreases notably. More details in Section 6.3.

The combined hierarchy WORDNET + FREQ achieves marginal improvements on *coarse* and *fine* granularities, while it degrades the performance on *ultra-fine* when compared to FREQ.

By imposing a hierarchical structure over the type vocabulary we can infer types that are located higher up in the hierarchy from the predictions of the lower ones. To analyze this, we add the closest *coarse* label to the *ultra-fine* prediction of each instance. Results are reported in Table 2b. The improvements are noticeable on the Macro score (up to 3.9 F1 points difference on FREQ) whereas Micro decreases. Since we are adding types to the prediction, this technique improves recall and penalizes precision. Macro is computed on the entity level, while Micro provides an overall score, showing that per instance the prediction tends to be better. The improvements can be observed on FREQ and PMI given that their predictions over

a) Example	Rin, Kohaku and Sesshomaru Rin befriends Kohaku, the demonslayer Sango’s younger brother, while Kohaku acts as her guard when Naraku is using her for bait to lure Sesshomaru into battle .
Annotation	<i>event, conflict, war, fight, battle, struggle, dispute, group_action</i>
Prediction	FREQ: event, conflict, war, fight, battle ; WORDNET: event, conflict , difference, engagement, assault
b) Example	The UN mission in Afghanistan dispatched its own investigation, expressing concern about reports of civilian casualties and calling for them to be properly cared for.
Annotation	<i>organization, team, mission</i>
Prediction	FREQ: organization, team, mission , activity, operation; WORDNET: group, institution, branch, delegation, corporation
c) Example	Brazilian President Luiz Inacio Lula da Silva and Turkish Prime Minister Recep Tayyip Erdogan talked about designing a strategy different from sanctions at a meeting Monday, Amorim said.
Annotation	<i>event, meeting, conference, gathering, summit</i>
Prediction	FREQ: event, meeting, conference , film, campaign; WORDNET: entity, meeting, gathering , structure, court

Table 3: Qualitative analysis of instances taken from the development set. The predictions are generated with the hyperbolic models of FREQ and WORDNET. Correct predictions are marked in blue color.

ultra-fine types are better.

6.2 Comparison of the Spaces

When comparing performances with respect to the metric spaces, the hyperbolic models for PMI and FREQ outperform all other models on *ultra-fine* granularity. Compared to its Euclidean counterpart, PMI brings considerable improvements (16.0 vs 12.2 and 15.4 vs 11.5 for Macro and Micro F1 respectively). This can be explained by the exponential growth of this space towards the boundary of the ball, combined with a representation that reflects the type co-occurrences in the dataset. Figure 4 shows a histogram of the distribution of ground-truth types as closest neighbors to the prediction.

On both Euclidean and hyperbolic models, the type embeddings for *coarse* and *fine* labels are located closer to the origin of the space. In this region, the spaces show a much more similar behavior in terms of the distance calculation, and this similarity is reflected on the results as well.

The low performance of the hyperbolic model of WORDNET on *coarse* can be explained by the fact that *entity* is the root node of the hierarchy, therefore it is located closer to the center of the space. Elements placed in the vicinity of the origin have a norm closer to zero, thus their distance to other types tends to be shorter (does not grow exponentially). This often misleads the model into assign *entity* as the *coarse*. See Table 3c for an example.

This issue is alleviated on WORDNET + FREQ. Nevertheless, it appears again when using the *ultra-fine* prediction to infer the *coarse* label. The

drop in performance can be seen in Table 2b: Macro F1 decreases by 8.0 and Micro F1 by 12.2.

6.3 Error analysis

We perform an error analysis on samples from the development set and predictions from two of our proposed hyperbolic models. We show three examples in Table 3. Overall we can see that predictions are reasonable, suggesting synonyms or related words.

In the proposed regression setup, we predict a fixed amount of labels per instance. This schema has drawbacks as shown in example a), where all predicted types by the FREQ model are correct though we can not predict more, and b), where we predict more related types that are not part of the annotations.

In examples b) and c) we see how the FREQ model predicts the *coarse* type correctly whereas the model that uses the WordNet hierarchy predicts *group* and *entity* since these labels are considered more general (*organization IS A group*) thus located closer to the origin of the space.

To analyse precision and recall more accurately, we compare our model to the one of Shimaoka et al. (2016) (ATTNER) and the multi-task model of Choi et al. (2018) (MULTI). We show the results for macro-averaged metrics in Table 4. Our model is able to achieve higher recall but lower precision. Nonetheless we are able to outperform ATTNER with a regression model even though they apply a classifier to the task.

Model	Dev			Test		
	P	R	F1	P	R	F1
ATTNER	53.7	15.0	23.5	54.2	15.2	23.7
FREQ	24.8	25.9	25.4	25.6	26.8	26.2
MULTI	48.1	23.2	31.3	47.1	24.2	32.0

Table 4: Combined performance over the three granularities. Results are extracted from Choi et al. (2018).

Model	Sp	Coarse		Fine		Ultra	
		Ma	Mi	Ma	Mi	Ma	Mi
ONTO	Hy	83.0	81.9	24.0	23.9	2.0	2.0
	Eu	82.2	82.2	28.8	28.7	2.4	2.4
FREQ	Hy	81.7	81.8	27.1	27.1	4.2	4.2
	Eu	81.7	81.7	30.6	30.6	3.8	3.8

Table 5: Macro and micro F1 results on OntoNotes.

6.4 Analysis Case: OntoNotes

To better understand the effects of the hierarchy and the metric spaces we also perform an evaluation on OntoNotes (Gillick et al., 2014). We compare the original hierarchy of the dataset (ONTO), and one derived from the type co-occurrence frequency extracted from the data augmented by Choi et al. (2018) with this type inventory. The results for the three granularities are presented in Table 5.

The FREQ model on the hyperbolic geometry achieves the best performance for the *ultra-fine* granularity, in accordance with the results on the Ultra-Fine dataset. In this case the improvements of the frequency-based hierarchy are not so remarkable when compared to the ONTO model given that the type inventory is much smaller, and the annotations follow a hierarchy where there is only one possible path for every label to its *coarse* type.

The low results on the *ultra-fine* granularity are

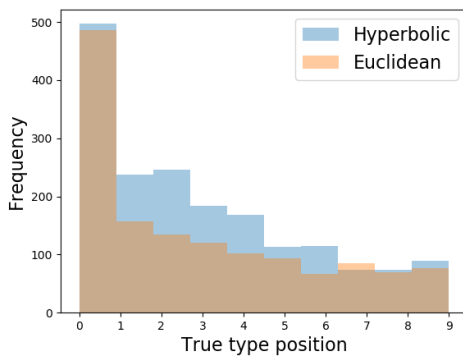


Figure 4: Histogram of ground-truth type neighbor positions for *ultra-fine* predictions in Hyperbolic and Euclidean spaces on the test set.

due to the reduced multiplicity of the annotated types (See Table 8). Most instances have only one or two types, setting very restrictive upper bounds for this setup.

7 Related Work

Type inventories for the task of fine-grained entity typing (Ling and Weld, 2012; Gillick et al., 2014; Yosef et al., 2012) have grown in size and complexity (Del Corro et al., 2015; Murty et al., 2017; Choi et al., 2018). Systems have tried to incorporate hierarchical information on the type distribution in different manners. Shimaoka et al. (2017) encode the hierarchy through a sparse matrix. Xu and Barbosa (2018) model the relations through a hierarchy-aware loss function. Ma et al. (2016) and Abhishek et al. (2017) learn embeddings for labels and feature representations into a joint space in order to facilitate information sharing among them. Our work resembles Xiong et al. (2019) since they derive hierarchical information in an unrestricted fashion, through type co-occurrence statistics from the dataset. These models operate under Euclidean assumptions. Instead, we impose a hyperbolic geometry to enrich the hierarchical information.

Hyperbolic spaces have been applied mostly on complex and social networks modeling (Krioukov et al., 2010; Verbeek and Suri, 2016). In the field of Natural Language Processing, they have been employed to learn embeddings for Question Answering (Tay et al., 2018), in Neural Machine Translation (Gulcehre et al., 2019), and to model language (Leimeister and Wilson, 2018; Tifrea et al., 2019). We build upon the work of Nickel and Kiela (2017) on modeling hierarchical link structure of symbolic data and adapt it with the parameterization method proposed by Dhingra et al. (2018) to cope with feature representations of text.

8 Conclusions

Incorporation of hierarchical information from large type inventories into neural models has become critical to improve performance. In this work we analyze expert-generated and data-driven hierarchies, and the geometrical properties provided by the choice of the vector space, in order to model this information. Experiments on two different datasets show consistent improvements of hyperbolic embedding over Euclidean baselines

on very fine-grained labels when the hierarchy reflects the annotated type distribution.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions, and we also thank Ana Marasović, Mareike Pfeil, Todor Mihaylov and Mark-Christoph Müller for their helpful discussions. This work has been supported by the German Research Foundation (DFG) as part of the Research Training Group Adaptive Preparation of Information from Heterogeneous Sources (AIPHES) under grant No. GRK 1994/1 and the Klaus Tschira Foundation, Heidelberg, Germany.

References

- Abhishek Abhishek, Ashish Anand, and Amit Awekar. 2017. [Fine-grained entity type classification by jointly learning representations and label embeddings](#). In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 797–807, Valencia, Spain. Association for Computational Linguistics.
- Silvère Bonnabel. 2011. [Stochastic gradient descent on riemannian manifolds](#). *IEEE Transactions on Automatic Control*, 58.
- Benjamin Paul Chamberlain, James Clough, and Marc Peter Deisenroth. 2017. [Neural embeddings of graphs in hyperbolic space](#). *CoRR*, abs/1705.10359.
- Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. [Ultra-fine entity typing](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.
- 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*, pages 868–878, Lisbon, Portugal. Association for Computational Linguistics.
- Bhuvan Dhingra, Christopher Shallue, Mohammad Norouzi, Andrew Dai, and George Dahl. 2018. [Embedding text in hyperbolic spaces](#). In *Proceedings of the Twelfth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-12)*, pages 59–69, New Orleans, Louisiana, USA. Association for Computational Linguistics.
- Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. [Context-Dependent Fine-Grained Entity Type Tagging](#). *ArXiv e-prints*.
- Caglar Gulcehre, Misha Denil, Mateusz Malinowski, Ali Razavi, Razvan Pascanu, Karl Moritz Hermann, Peter Battaglia, Victor Bapst, David Raposo, Adam Santoro, and Nando de Freitas. 2019. [Hyperbolic attention networks](#). In *International Conference on Learning Representations*.
- Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, abs/1412.6980.
- Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marin Bogu. 2010. [Hyperbolic geometry of complex networks](#). *Physical review. E, Statistical, nonlinear, and soft matter physics*, 82:036106.
- Matthias Leimeister and Benjamin J. Wilson. 2018. [Skip-gram word embeddings in hyperbolic space](#). *CoRR*, abs/1809.01498.
- Xiao Ling and Daniel S. Weld. 2012. [Fine-grained entity recognition](#). In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI’12*, pages 94–100. AAAI Press.
- Yukun Ma, Erik Cambria, and SA GAO. 2016. [Label embedding for zero-shot fine-grained named entity typing](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180, Osaka, Japan.
- Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. 2017. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pages 6297–6308.
- George A. Miller. 1992. [Wordnet: A lexical database for english](#). In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.
- Shikhar Murty, Patrick Verga, Luke Vilnis, and Andrew McCallum. 2017. [Finer grained entity typing with typenet](#). In *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA.
- Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 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: Long Papers)*, pages 97–109, Melbourne, Australia. Association for Computational Linguistics.

- Maximilian Nickel and Douwe Kiela. 2017. [Poincaré embeddings for learning hierarchical representations](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6341–6350. Curran Associates, Inc.
- Maximilian Nickel and Douwe Kiela. 2018. [Learning continuous hierarchies in the Lorentz model of hyperbolic geometry](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3779–3788, Stockholm, Sweden. PMLR.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. [Glove: Global vectors for word representation](#). In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.
- Lisa F Rau. 1991. Extracting company names from text. In *Artificial Intelligence Applications, 1991. Proceedings., Seventh IEEE Conference on*, volume 1, pages 29–32. IEEE.
- 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, KDD '16*, pages 1825–1834, New York, NY, USA. ACM.
- Frederic Sala, Chris De Sa, Albert Gu, and Christopher Re. 2018. [Representation tradeoffs for hyperbolic embeddings](#). In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4460–4469, Stockholm, Sweden. PMLR.
- Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. [A hierarchical multi-task approach for learning embeddings from semantic tasks](#). In *AAAI*, volume abs/1811.06031.
- Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. [An attentive neural architecture for fine-grained entity type classification](#). In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 69–74, San Diego, CA. Association for Computational Linguistics.
- 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.
- Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. [Hyperbolic representation learning for fast and efficient neural question answering](#). In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM '18*, pages 583–591, New York, NY, USA. ACM.
- Alexandru Tifrea, Gary Becigneul, and Octavian-Eugen Ganea. 2019. [Poincaré glove: Hyperbolic word embeddings](#). In *International Conference on Learning Representations*.
- 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, CONLL '03*, pages 142–147, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Kevin Verbeek and Subhash Suri. 2016. [Metric embedding, hyperbolic space, and social networks](#). *Computational Geometry*, 59:1 – 12.
- Wenhan Xiong, Jiawei Wu, Deren Lei, Mo Yu, Shiyu Chang, Xiaoxiao Guo, and William Yang Wang. 2019. [Imposing label-relational inductive bias for extremely fine-grained entity typing](#). In *Proceedings of NAACL-HLT 2019*.
- Peng Xu and Denilson Barbosa. 2018. [Neural fine-grained entity type classification with hierarchy-aware loss](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 16–25, New Orleans, Louisiana. Association for Computational Linguistics.
- 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*, pages 1183–1194, Valencia, Spain. Association for Computational Linguistics.
- Semih Yavuz, Izzeddin Gur, Yu Su, Mudhakar Srivatsa, and Xifeng Yan. 2016. [Improving semantic parsing via answer type inference](#). In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 149–159, Austin, Texas. Association for Computational Linguistics.
- Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. [Embedding methods for fine grained entity type classification](#). In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296, Beijing, China. Association for Computational Linguistics.
- Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012.

HYENA: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India.

Ben Zhou, Daniel Khashabi, Chen-Tse Tsai, and Dan Roth. 2018. **Zero-shot open entity typing as type-compatible grounding.** In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2065–2076, Brussels, Belgium. Association for Computational Linguistics.

A Appendix

A.1 Hyperparameters

Both hyperbolic and Euclidean models were trained with the following hyperparameters.

Parameter	Value
Word embedding dim	300
Max mention tokens	5
Max mention chars	25
Context length (per side)	10
Char embedding dim	50
Position embedding dim	25
Context LSTM dim	200
Attention dim	100
Mention dropout	0.5
Context dropout	0.2
Max gradient norm	10
Projection hidden dim	500
Optimizer	Adam
Learning rate	0.001
Batch size	1024
Epochs	50

Table 6: Model hyperparameters.

A.2 Dataset statistics

Split	Samples	Coarse	Fine	Ultra-fine
Train	6,240,105	2,148,669	2,664,933	3,368,607
Dev	1,998	1,612	947	1,860
Test	1,998	1,598	964	1,864

Table 7: Amount of samples with at least one label of the granularity organized by split on Ultra-Fine Dataset.

Split	Samples	Coarse	Fine	Ultra
Train	793,487	828,840	735,162	301,006
Dev	2,202	2,337	869	76
Test	8,963	9,455	3,521	417

Table 8: Samples and label distribution by split on OntoNotes Dataset.