Using Hyperbolic Geometry for FG-NET over Distantly Supervised data

Muhammad Asif Ali,1,2 Yifang Sun,1 Bing Li,1 Wei Wang,1,3
1School of Computer Science and Engineering, UNSW, Australia
2Shenzhen Institute of Computing Sciences, China
3College of Computer Science and Technology, DGUT, China
asif.ali@sic.ac.cn, bing.li@unsw.edu.au {yifangs, weiw}@cse.unsw.edu.au

Abstract
Fine-Grained Named Entity Typing (FG-NET) classifies an entity mention into a fine range of entity types. A large number of entity types make it difficult to manually label the training data, thus distant supervision is used to automatically acquire the training data. Distant supervision incurs a lot of training noise which hinders the performance improvement of the FG-NET systems. In this paper, we propose to use hyperbolic geometry for FG-NET with the hope that it can help overcoming the noise incurred by distant supervision.

Keywords — FG-NET, Distant Supervision

1 Introduction
Named Entity Typing (NET) is a fundamental operation in natural language processing, it aims at assigning discrete type labels to the entity mentions in the text. It has immense applications, including: knowledge base construction [2]; information retrieval [12]; question answering [18]; relation extraction [27] etc. Traditional NET systems work with only a coarse set of type labels, e.g., organization, person, location, etc., which severely limit their potential in the down-streaming tasks. In recent past, the idea of NET is extended to Fine-Grained Named Entity Typing (FG-NET) that assigns a wide range of correlated entity types to the entity mentions [13]. Compared to NET, the FG-NET has shown a remarkable improvement in the sub-sequent applications. For example, Ling and Weld, [13] showed that FG-NET can boost the performance of the relation extraction by 93%.

FG-NET encompasses hundreds of correlated entity types with little contextual differences, which makes it labour-intensive and error-prone to acquire manually labeled training data. Therefore, distant supervision is widely used to acquire training data for this task. Distant supervision relies on: (i) automated routines to detect the entity mention, and (ii) using type-hierarchy from existing knowledge-bases, e.g., Probase [24], to assign type labels to the entity mention. However, it assigns type-labels to the entity mention irrespective of the mention’s context, which results in label noise [20]. Examples in this regard are shown in Figure 1 where the distant supervision assigns labels: {person, author, president, actor, politician} to the entity mention: “Donald Trump”, whereas, from contextual perspective, it should be labeled as: {person, president, politician} in S1, and {person, actor} in S2. Likewise, the entity mention: “Vladimir Putin” should be labeled as: {person, author} and {person, athlete} in S3 and S4 respectively. This label noise in-turn propagates in the model learning and severely effects/limits the end-performance of the FG-NET systems.

Earlier research on FG-NET either ignored the label noise [13], or applied some heuristics to prune the noisy labels [8]. Ren et al., [19] bifurcated the training data into clean and noisy data samples, and used different set of loss functions to model them. However, the modeling heuristics proposed by these models are not able to cope with the label noise, which limits the end-performance of the FG-NET systems relying on distant supervision. We, moreover, observe that these models are designed assuming a euclidean nature of the problem, which is inappropriate for FG-NET, as the fine-grained type hierarchy exhibit a hierarchical structure. Given that it is not possible to embed hierarchies in euclidean space [15], this assumption, in turn limits the ability of the existing models to: (i) effectively represent FG-NET data, (ii) cater label noise, and (iii) perform FG-NET classification task in a robust way.

The inherent advantage of hyperbolic geometry to embed hierarchies is well-established in literature. It enforces the items on the top of the hierarchy to be placed close to the origin, and the items down the hierarchy near infinity. This enables the embedding norm to cater to the depth in the hierarchy, and the distance between embeddings represent the similarity between the items. Thus the items sharing a parent node are close to each other in the embeddings space. This makes the hyperbolic space a perfect paradigm for embedding the distantly supervised FG-NET data, as it explicitly allows label-smoothing by sharing the contextual information across noisy entity mentions corresponding to the same type hierarchy, as shown in Figure 2(b), for a 2D Poincaré Ball. For example, given the type hierarchy: “Person” ← “Leader” ← “Politician” ← “President”, the hyperbolic
In his 34th presidential session, Trump said, he is no longer in favor of trade war.

In this 2004 book: 'Judo: History, Theory, Practice' Putin discussed basics of Judo.

Vladimir Putin began judo classes in Russian capital, when he was just eleven.

Figure 1: FG-NET training data acquired by distant supervision

Figure 2: (a) An illustration of how the entity type “President” shares the context of the entity type “Politician” which in turn shares the context of the entity-type “Leader” and so on; (b) Embedding FG-NET data in 2-D Poincare Ball, where each disjoint type may be embedded along a different direction

embeddings, on contrary to the euclidean embeddings, offer a perfect geometry for the entity type “President” to share and augments the context of “Politician”, which in turn adds to the context of “Leader” and “Person” etc., shown in Figure 2(a). We hypothesize that such hierarchically-organized contextually similar neighbours provide a robust platform for the end task, i.e., FG-NET over distantly supervised data, also discussed in detail in the section 4.5.1.

Nevertheless, we propose Fine-Grained Entity Typing with Refinement in Hyperbolic space (FGNET-RH), shown in Figure 3. FGNET-RH follows a two-stage process, stage-I: encode the mention along with its context using multiple LSTM networks, stage-II: form a graph to refine mention’s encoding from stage-I by sharing contextual information in the hyperbolic space. In order to maximize the benefits of using the hyperbolic geometry in combination with the graph structure, FGNET-RH maps the mention encodings (from stage-I) to the hyperbolic space. And, performs all the operations: linear transformation, type-specific contextual aggregation etc., in the hyperbolic space, required for appropriate additive context-sharing along the type hierarchy to smoothen the noisy type-labels prior to the entity typing. The major contributions of FGNET-RH are enlisted as follows:

1. FGNET-RH accommodates the benefits of: the graph structures and the hyperbolic geometry to perform fine-grained entity typing over distantly supervised noisy data in a robust fashion.

2. FGNET-RH explicitly allows label-smoothing over the noisy training data by using graphs to combine the type-specific contextual information along the type-hierarchy in the hyperbolic space.

3. Experimentation using two models of the hyperbolic space, i.e., the Hyperboloid and the Poincaré-Ball, shows that FGNET-RH outperforms the existing research by up to 3.5% in terms of strict accuracy.

2 Related Work

Existing research on FG-NET can be bifurcated into two major categories: (i) traditional feature-based systems, and (ii) embedding models.

Traditional feature-based systems rely on feature extraction, later using these features to train machine learning models for classification. Amongst them, Ling and Weld [13] developed FiGER, that uses hand-crafted features to develop a multi-label, multi-class perceptron classifier. Yosef et al., [29] developed HYENA, i.e., a hierarchical type classification model using hand-crafted features in combination with the SVM classifier. Gillick et al., [8] proposed context-dependent fine-grained typing using hand-crafted features along with logistic regression classifier. Shimaoka et al., [21] developed neural architecture for fine-grained entity typing using a combination of automated and hand-crafted features.
Embedding models use widely available embedding resources with customized loss functions to form classification models. Yogatama et al., [28] used embeddings along with Weighted Approximate Rank Pairwise (WARP) loss. Ren et al., [19] proposed AFET that uses different set of loss functions to model the clean and the noisy entity mentions. Abhishek et al., [11] proposed end-to-end architecture to jointly embed the mention and the label embeddings. Xin et al., [25] used language models to compute the compatibility between the context and the entity type prior to entity typing. Choi et al., [4] proposed ultra-fine entity typing encompassing more than 10,000 entity types. They used crowd-sourced data along with the distantly supervised data for model training.

Graph convolution networks are introduced in recent past in order to extend the concept of convolutions from regular-structured grids to graphs [11]. Ali et al., [2] proposed attentive convolutional network for fine-grained entity typing. Nickel et al., [15] illustrated the benefits of hyperbolic geometry for embedding the graph structured data. Chami et al., [3] combined graph convolutions with the hyperbolic geometry. López et al., [14] used hyperbolic geometry for ultra-fine entity typing. To the best of our knowledge, we are the first to explore the combined benefits of the graph convolution networks in relation with the hyperbolic geometry for FG-NET over distantly supervised noisy data.

3 Proposed Approach

3.1 Problem Definition In this paper, we build a multi-class, multi-label entity typing system using distantly supervised data to classify an entity mention into a set of fine-grained entity types. Specifically, we propose attentive type-specific contextual aggregation in the hyperbolic space to fine-tune the mention’s encodings learnt over noisy data prior to entity typing. We assume the availability of training corpus $C_{train}$ acquired via distant supervision, and manually labeled test corpus $C_{test}$. Each corpus $C$ (train/test) encompasses a set of sentences. For each sentence, the contextual token $\{c_i\}_{i=1}^N$, the mention spans $\{m_i\}_{i=1}^N$ (corresponding to the entity mentions), and the candidate type labels $\{t_i\}_{i=1}^N \in \{0, 1\}^T$ ($T$-dimensional vector with $t_{i,x} = 1$ if $x^{th}$ type corresponds to the true label and zero otherwise) have been priory identified. The type labels are inferred from type hierarchy in the knowledge base $\psi$ with the schema $T_\psi$. Similar to Ren et al., [19], we bifurcate the training data $D_{tr}$ into clean $D_{tr-clean}$ and noisy $D_{tr-noisy}$, if the corresponding mention’s type-path follows a single path in the type-hierarchy $T_\psi$ or otherwise. Following the type-path in Figure (ii), a mention with labels $\{person, author\}$ will be considered as clean, whereas, a mention with labels $\{person, president, author\}$ will be considered as noisy.

3.2 Overview Our proposed model, FGNET-RH, follows a two-step approach, labeled as stage-I and stage-II in the Figure. Stage-I follows text encoding pipeline to generate mention’s encoding in relation with its context. Stage-II is focused on label noise reduction, for this, we map the mention’s encoding (from stage-I) in the hyperbolic space and use a graph to share aggregated type-specific contextual information along the type-hierarchy in order to refine the mention encoding. Finally, the refined mention encoding is embedded along with the label encodings in the hyperbolic space for entity typing. Details of each stage are given in the following sub-sections.

3.3 Stage-I (Noisy Mention Encoding) Stage-I follows a standard text processing pipeline using multiple LSTM networks [9] to encode the entity mention in relation with its context. Individual components of stage-I are explained as follows:

**Mention Encoding:** We use LSTM network to encode the character sequence corresponding to the mention tokens. We use $\phi_c = [m \hat{c}] \in \mathbb{R}^c$ to represent the encoded mention’s tokens.

**Context Encoding:** For context encoding, we use multiple Bi-LSTM networks to encode the tokens corresponding to the left and the right context of the entity mention. We use $\phi_l = \{l \hat{c} \}$ and $\phi_r = \{r \hat{c} \}$ to represent the encoded left and the right context respectively.

**Position Encoding:** For position encoding, we use LSTM network to encode the position of the left and the right contextual tokens. We use $\phi_p = \{p \hat{c} \}$ to represent the encoded position corresponding to the mention’s left and the right context.

**Mention Encodings:** Finally, we concatenate all the mention-specific encodings to get L-dimensional noisy encoding: $x_m \in \mathbb{R}^L$, where $L = e + 2 * c + 2 * p$.

\[
x_m = [\phi_p; \phi_c; \phi_r] \tag{3.1}
\]

3.4 Stage-II (Fine-tuning the Mention Encodings) Stage-II is focused on alleviating the label noise. Underlying assumption in combating the label noise is that the contextually similar mentions should get similar type labels. For this, we form a graph to cluster contextually-similar mentions and employ hyperbolic geometry to share the contextual information along the type-hierarchy. As shown in Figure, the stage-II follows the following pipeline:

1. Construct a graph such that contextually and semantically similar mentions end-up being the neighbors in the graph.

2. Use exponential map to project the noisy mention encodings from stage-I to the hyperbolic space.
3. In the hyperbolic space, use the corresponding exponential and logarithmic transformations to perform the core operations, i.e., (i) linear transformation, and (ii) contextual aggregation, required to fine-tune the encodings learnt in stage-I prior to entity typing.

We work with two models in the hyperbolic space, i.e., the Hyperboloid (ℍ^d,K) and the Poincaré-Ball (D^d). In the following sub-sections, we provide the mathematical formulation for the Hyperboloid model of the hyperbolic space. Similar formulation can be designed for the Poincaré-Ball model.

**3.4.1 Hyperboloid Model** d-dimensional hyperboloid model of the hyperbolic space (denoted by ℍ^d,K) is a space of constant negative curvature −1/K, with ℳ^d,K as the euclidean tangent space at point p, such that:

\[ ℍ^d,K = \{ p \in \mathbb{R}^{d+1} : \langle p, p \rangle = -K, p_0 > 0 \} \]

\[ ℳ^d,K = \{ r \in \mathbb{R}^{d+1} : \langle r, p \rangle_L = 0 \} \]

where \( \langle \cdot, \cdot \rangle_L : \mathbb{R}^{d+1} \times \mathbb{R}^{d+1} \to \mathbb{R} \) denotes the Minkowski inner product, with \( \langle p, q \rangle_L = -p_0 q_0 + p_1 q_1 + \ldots + p_d q_d \).

**Geodesics and Distances:** For two points \( p, q \in ℍ^d,K \), the distance function between them is given by:

\[ d^d_K(p, q) = \sqrt{K} \arccosh(-\langle p, q \rangle_L / K) \]

**Exponential and Logarithmic maps:** We use exponential and logarithmic maps for mapping to and from the hyperbolic and the tangent space respectively. Formally, given a point \( p \in ℍ^d,K \) and tangent vector \( t \in ℳ^d,K \), the exponential map \( \exp_p : ℳ^d,K \to ℍ^d,K \) assigns a point to \( t \) such that \( \exp_p(t) = \gamma(1) \), where \( \gamma \) is the geodesic curve that satisfies \( \gamma(0) = p \) and \( \dot{\gamma}(0) = t \).

The logarithmic map \( \log_p \) being the bijective inverse maps a point in hyperbolic space to the tangent space at \( p \). We use the following equations for the exponential and the logarithmic maps:

\[ \exp_p(v) = \cosh(\|v\|_L \sqrt{K}) p + \sqrt{K} \sinh(\|v\|_L \sqrt{K}) \frac{v}{\|v\|_L} \]

\[ \log_p(q) = d^d_K(p, q) \frac{q + \frac{1}{K} p}{\|q + p\|_L} \]
T, where \( \delta \) is a threshold. Finally, we form pairwise edges for all the mention candidates corresponding to each entity-type, i.e., \{ cand \( i \) \} \( t \) \( = \) \( 1 \), to construct the graph \( G \), with adjacency matrix \( A \).

### 3.4.3 Mapping Noisy Mention Encodings to the Hyperbolic space

The mention encodings learnt in the stage-I are noisy, as they are learnt over distantly supervised data. These encodings lie in the euclidean space, and in order to refine them, we first map them to the hyperbolic space, where we may best exploit the fine-grained type hierarchy in relation with the type-specific context to fine-tune these encodings as an aggregate of contextually-similar neighbors.

Formally, let \( p^E = \{ m \in \mathbb{R}^{N \times L} \} \) be the matrix corresponding to the noisy mentions’ encodings in the euclidean domain. We consider \( o = \{ \sqrt{K}, 0, ..., 0 \} \) as a reference point (origin) in a d-dimensional Hyperboloid with curvature \(-1/K (\mathbb{H}^{d,K}) \), as a point in the tangent space \( \mathcal{T} \mathbb{H}^{d,K} \), and map it to \( p^H \in \mathbb{H}^{d,K} \) using the exponential map given in Equation (3.4), as follows:

\[
\begin{align*}
p^H &= \exp^K((0, p^E)) \\
\exp^K((0, p^E)) &= \left( \sqrt{K} \cosh \left( \frac{\|p^E\|_2}{\sqrt{K}} \right), \right. \\
&\left. \sqrt{K} \sinh \left( \frac{\|p^E\|_2}{\sqrt{K}} \right) \|p^E\|_2 \right)
\end{align*}
\]

#### (3.6)

### 3.4.4 Linear Transformation

In order to perform linear transformation operation on the noisy mention encodings, i.e., \((i)\) multiplication by weight matrix \( W \), and \((ii)\) addition of bias vector \( b \), we rely on the exponential and the logarithmic maps. For multiplication with the weight matrix, firstly, we apply logarithmic map on the encodings in the hyperbolic space, i.e., \( p^H \in \mathbb{H}^{d,K} \), in order to project them to \( \mathcal{T} \mathbb{H}^{d,K} \). This projection is then multiplied by the weight matrix \( W \), and the resultant vectors are projected back to the manifold using the exponential map. For a manifold with curvature constant \( K \), these operations can be summarized in the equation, given below:

\[
W \otimes p^H = \exp^K (W \log^K(p^H))
\]

#### (3.7)

For bias addition, we rely on parallel transport, let \( b \) be the bias vector in \( \mathcal{T} \mathbb{H}^{d,K} \), we parallel transport \( b \) along the tangent space and finally map it to the manifold. Formally, let \( T^K_{\alpha \rightarrow p^H} \) represent the parallel transport of a vector from \( T^\alpha \mathbb{H}^{d,K} \) to \( T^H \mathbb{H}^{d,K} \), we use the following equation for the bias addition:

\[
p^H \oplus b = \exp^K(T^K_{\alpha \rightarrow p^H}(b))
\]

#### (3.8)

### 3.4.5 Type-Specific Contextual Aggregation

Aggregation is a crucial step for noise reduction in FG-NET, it helps to smoothen the type-label by refining/fine-tuning the noisy mention encodings by accumulating information from contextually similar neighbors lying at multiple hops. Given the graph \( G \), with nodes \( \{ V \} \) being the entity mentions, we use the pairwise embedding vectors along the edges of the graph to compute the attention weights \( \eta_{ij} = \cos(men_i, men_j) \forall (i,j) \in V \). In order to perform the aggregation operation, we first use the logarithmic map to project the results of the linear transformation from hyperbolic space to the tangent space. Later, we use the neighboring information contained in \( G \) to compute the refined mention encoding as attentive aggregate of the neighboring mentions. Finally, we map these results back to the manifold using the exponential map \( \exp^K \). Our methodology for contextual aggregation is summarized in the following equation:

\[
AGG_{ctxz}(p^H)_i = \exp^K_{\chi_i} \left( \sum_{j \in N(i)} (\eta_{ij} \odot A) \log^K(p^H_j) \right)
\]

where \( \eta_{ij} \odot A \) is the Hadamard product of the attention weights and the adjacency matrix \( A \). It accommodates the degree of contextual similarity among the mention pairs in \( G \).

### 3.4.6 Non-Linear Activation

Contextually aggregated mention’s encoding is finally passed through a non-linear activation function \( \sigma \) (ReLU in our case). For this, we follow similar steps, i.e., \((i)\) map the encodings to the tangent space, \((ii)\) apply the activation function in the tangent space, \((iii)\) map the results back to the hyperbolic space using the exponential map. These steps are summarized in the following equation:

\[
\sigma(p^H) = \exp^K(\sigma(\log^K(p^H)))
\]

#### (3.10)

### 3.5 Complete Model

We combine the above-mentioned steps to get the refined mention encodings at \( l \)th-layer \( z^l,H \) out as follows:

\[
\begin{align*}
p^{l,H} &= W^l \otimes p^{l-1,H} \oplus b^l; \\
z^{l,H} &= AGG_{ctxz}(p^{l,H}); \\
z^{l,H}_{out} &= \sigma(z^{l,H})
\end{align*}
\]

Let \( z^{l,H}_{out} \in \mathbb{H}^{d,K} \) correspond to the refined mentions’ encodings hierarchically organized in the hyperbolic space. We embed them along with the fine-grained type label encodings \( \{ \phi_t \}^{T}_{t=1} \in \mathbb{H}^{d} \). For that we learn a function \( f(z^{l,H}_{out}, \phi_t) = \phi_t \times z^{l,H} + bias_t \), and separately learn the loss functions for the clean and the noisy mentions.
In order to set up a fair platform for comparative evaluation, we use the same data settings (training, dev and test splits) as used by all the models considered as baselines in Table 2. All the experiments are performed using Intel Gold 6240 CPU with 256 GB main memory.

**Model Parameters:** For stage-I, the hidden layer size of the context and the position encoders is set to 100d. The hidden layer size of the mention character encoder is 200d. Character, position and label embeddings are randomly initialized. We report the model performance using 300d Glove [16] and 1024d deep contextualized embeddings [17].

For stage-II, we construct graphs with 5.4M and 0.6M edges for BBN and OntoNotes respectively. Curvature constant of the hyperbolic space is set to $K = 1$. All the models are trained using Adam optimizer [10] with learning rate $= 0.001$.

### 4.3 Model Comparison
We evaluate FGNET-RH against the following baseline models: (i) Figer [13]; (ii) Hyena [29]; (iii) AFET, AFET-NoCo and AFET-NoPa [19]; (iv) Attentive [21]; (v) FNET [1]; (vi) NFGEC + LME [25]; and (vii) FGET-RR [2]. For performance comparison, we use the scores reported in the original papers, as they are computed using a similar data settings as that of ours.

Note that we do not compare our model against [4] [14] because these models use crowd-sourced data in addition to the distantly supervised data for model training. Likewise, we exclude [26] from evaluation because Xu and Barbosa changed the fine-grained problem definition from multi-label to single-label classification problem. This makes their problem settings different from that of ours and the end results are no longer comparable.

### 4.4 Main Results
The results of the proposed model are shown in Table 2. For each data, we boldface the best scores with the existing state-of-the-art underlined. These results show that FGNET-RH outperforms the existing state-of-the-art models by a significant margin. For the BBN data, FGNET-RH achieves 3.5%, 1.2% and 1.5% improvement in strict accuracy, mac-F1 and mic-F1 respectively, compared to the FGET-RR. For OntoNotes, FGNET-RH improves the mac-F1 and mic-F1 scores by 1.2% and 1.6%.

These results show that FGNET-RH offers multi-faceted benefits, i.e., using hyperbolic space in combination with the graphs to encode the hierarchy, while at the same time catering to noise in the best possible way. Especially, augmented context sharing along the hierarchy leads to considerable improvement in the performance compared to the baseline models.

### 4.5 Ablation Study
In this section, we evaluate the impact of different model components on label de-noising. Specifically, we analyze the performance of FGNET-RH using variants of the adjacency graph, including: (i) ran-

| Dataset       | BBN | OntoNotes |
|---------------|-----|-----------|
| Training Mentions | 86078 | 220398 |
| Testing Mentions  | 13187 | 9603 |
| % clean mentions (training) | 75.92 | 72.61 |
| % clean mentions (testing)  | 100 | 94.0 |
| Entity Types          | 47  | 89       |

Table 1: Fine-Grained Named Entity Typing data sets
Table 2: FG-NET performance comparison against baseline models

| Model                  | OntoNotes strict mac-F1 | mic-F1 | BBN strict mac-F1 | mic-F1 |
|------------------------|--------------------------|--------|-------------------|--------|
| FIGER [13]             | 0.369 0.578             | 0.516  | 0.467 0.672       | 0.612  |
| HYENA [29]             | 0.249 0.497             | 0.446  | 0.523 0.576       | 0.587  |
| AFET-NoCo [19]         | 0.486 0.652             | 0.594  | 0.655 0.711       | 0.716  |
| AFET-NoPa [19]         | 0.463 0.637             | 0.591  | 0.669 0.715       | 0.724  |
| AFET-CoH [19]          | 0.521 0.680             | 0.609  | 0.657 0.703       | 0.712  |
| AFET [19]              | 0.551 0.711             | 0.647  | 0.670 0.727       | 0.735  |
| Attentive [21]         | 0.473 0.655             | 0.586  | 0.484 0.732       | 0.724  |
| FNET-AIIC [1]          | 0.514 0.672             | 0.626  | 0.655 0.736       | 0.752  |
| FNET-NoM [1]           | 0.521 0.683             | 0.626  | 0.615 0.742       | 0.755  |
| FNET [1]               | 0.522 0.685             | 0.633  | 0.604 0.741       | 0.757  |
| NFGEC+LME [25]         | 0.529 0.724             | 0.652  | 0.607 0.743       | 0.760  |
| FGET-RR [2] (Glove)    | 0.567 0.737             | 0.680  | 0.740 0.811       | 0.817  |
| FGET-RR [2] (ELMO)     | 0.577 0.743             | 0.685  | 0.703 0.819       | 0.823  |
| FGNET-RH (Hyperboloid + Glove) | 0.580 0.738 | 0.685 | 0.766 0.828 | 0.835 |
| FGNET-RH (Hyperboloid + ELMO) | 0.575 0.752 | 0.696 | 0.712 0.824 | 0.823 |
| FGNET-RH (Poincaré-Ball + Glove) | 0.579 0.741 | 0.684 | 0.760 0.829 | 0.833 |
| FGNET-RH (Poincaré-Ball + ELMO) | 0.573 0.740 | 0.685 | 0.698 0.828 | 0.830 |

Table 3: FGNET-RH performance comparison using different adjacency matrices and Glove Embeddings

| Model                  | OntoNotes strict mac-F1 | mic-F1 | BBN strict mac-F1 | mic-F1 |
|------------------------|--------------------------|--------|-------------------|--------|
| FIGER [13]             | 0.484 0.643             | 0.597  | 0.486 0.647       | 0.653  |
| FGNET-RH [A]           | 0.531 0.699             | 0.632  | 0.735 0.808       | 0.815  |
| FGNET-RH [η ⊙ A]       | 0.580 0.738             | 0.685  | 0.766 0.828       | 0.835  |
| Hyperboloid (H²)       |                          |        |                   |        |
| FGNET-RH [R]           | 0.490 0.665             | 0.608  | 0.633 0.704       | 0.724  |
| FGNET-RH [A]           | 0.571 0.737             | 0.679  | 0.746 0.814       | 0.822  |
| FGNET-RH [η ⊙ A]       | 0.579 0.741             | 0.684  | 0.760 0.829       | 0.833  |
| Poincaré-Ball (P²)     |                          |        |                   |        |

Effectiveness of Hyperbolic Geometry

In order to verify the effectiveness of refining the mention encodings in the hyperbolic space (stage-II), we perform label-wise performance analysis for the dominant labels in the BBN dataset. Corresponding results for the Hyperboloid and the Poincaré-Ball model (in Table 3) show that FGNET-RH outperforms the existing state-of-the-art, i.e., FGET-RR by Ali et al., achieving higher F1-scores across all the labels. Note that FGNET-RH can achieve higher performance for the base type labels: \{e.g., "Person", "Organization", "GPE" etc.,\} as well as other type labels down in the hierarchy, \{e.g., "Organization/Corporation", "GPE/City" etc.,\}. For \{"Organization" and "Corporation"\} FGNET-RH achieves a higher F1=0.896 and F1=0.855 respectively, compared to the F1=0.881 and F1=0.844 by FGET-RR. This is made possible because embedding in the hyperbolic space enables type-specific context sharing at each level of the type hierarchy by appropriately adjusting the norm of the label vector.

To further strengthen our claims regarding the effectiveness of using hyperbolic space for FG-NET, we analyzed the context of the entity types along the type-hierarchy. We observed, for the fine-grained type labels, the context is additive and may be arranged in a hierarchical structure with the generic terms lying at the root and the specific terms lying along the children nodes. For example, "Government Organization" being a subtype of "Organization" adds tokens similar to \{bill, treasury, deficit, fiscal, senate etc.,\} to the context of "Organization". Likewise, "Hospital" adds tokens similar to \{family, patient, kidney, stone, infection etc.,\} to the context of "Organization".

This finding correlates with the norm of the label vectors, shown in Table 3 for the Poincaré-Ball model. The vector norm of the entity types deep in the hierarchy \{e.g., "Facility/Building", "Facility/Bridge", "Facility/Highway" etc.,\} is greater than that of the base en-
4.5.2 Error Cases

We analyzed the prediction errors of the FGNET-RH and attribute them to the following factors:

**Inadequate Context**: For these cases, type-labels are dictated entirely by the mention tokens, with very little information contained in the context. For example, in the sentence: “The IRS recently won part of its long-running battle against John.”, the entity mention “IRS” is labeled as “/Organization/Corporation” irrespective of any information contained in the mention’s context. Limited information contained in the mention’s context in turn limits the end-performance of FGNET-RH in predicting all possible fine-grained labels thus effecting the recall. For the BBN data set, more than 30% errors may be attributed to the inadequate mention’s context.

**Correlated Context**: FG-NET type hierarchy encompasses semantically correlated entity types, e.g., {“Organization” vs “Corporation”}; {“Actor” vs “Artist”}; {“Actor” vs “Director”}; {“Ship” vs “Spacecraft”}; {“Coach” vs “Athlete”} etc., with highly convoluted context. For example, the context of the entity types ("actor") and ("artist") is extremely overlapping, it contains semantically-related tokens like: (direct, dialogue, dance, acting, etc.,). This high contextual overlap makes it hard for the FGNET-RH to delineate the decision boundary across these correlated entity types. It leads to false predictions by the model thus effecting the precision. For the BBN data set, more than 35% errors may be attributed to the correlated context.

**Label Bias**: Label bias originating from the distant supervision may result in the label-smoothing to be ineffective. This occurs specifically if all the labels originating from the distant supervision are incorrect. For the BBN data set, approximately 5% errors may be attributed to the label bias.

The rest of the errors may be attributed to the inability of FGNET-RH to explicitly deal with different word senses, in-depth syntactic analysis, in-adequacy of underlying embedding models to handle semantics, etc. We plan to accommodate these aspects in the future work.

5 Conclusions

In this paper, we introduced FGNET-RH, a novel approach that combines the benefits of graph structures and hyperbolic geometry to perform entity typing in a robust fashion. FGNET-RH initially learns noisy mention encodings using LSTM networks and constructs a graph to cluster contextually similar mentions using embeddings in euclidean domain, later it performs label-smoothing in hyperbolic domain to refine the noisy encodings prior to the entity-typing. Performance evaluation using the benchmark datasets shows that the FGNET-RH offers a perfect geometry for context sharing across distantly supervised data, and in turn outperforms the existing research on FG-NET by a significant margin.
References

[1] Abhishek, Ashish Anand, and Amit Awekar. Fine-grained entity type classification by jointly learning representations and label embeddings. In *EACL (1)*, pages 797–807. Association for Computational Linguistics, 2017.

[2] Muhammad Asif Ali, Yifang Sun, Bing Li, and Wei Wang. Fine-grained named entity typing over distantly supervised data based on refined representations. In *AAAI*, pages 7391–7398. AAAI Press, 2020.

[3] Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. In *NeurIPS*, pages 4869–4880, 2019.

[4] Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. Ultra-fine entity typing. In *ACL (1)*, pages 87–96. Association for Computational Linguistics, 2018.

[5] Joachim Daiber, Max Jakob, Chris Hokamp, and Pablo N. Mendes. Improving efficiency and accuracy in multilingual entity extraction. In *I-SEMAN TICS*, pages 121–124. ACM, 2013.

[6] Bhuwan Dhingra, Christopher J. Shallue, Mohammad Norouzi, Andrew M. Dai, and George E. Dahl. Embedding text in hyperbolic spaces. In *TextGraphs@NAACL-HLT*, pages 59–69. Association for Computational Linguistics, 2018.

[7] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*, pages 601–610. ACM, 2014.

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

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

[11] Thomas N. Kipf and Max Welling. Semi-supervised learning with graph convolutional networks. In *ICLR (Poster)*, OpenReview.net, 2017.

[12] Ni Lao and William W Cohen. Relational retrieval using a combination of path-constrained random walks. *Machine learning*, 81(1):53–67, 2010.

[13] Xiao Ling and Daniel S. Weld. Fine-grained entity recognition. In *AAAI*. AAAI Press, 2012.

[14] Federico López, Benjamin Heiznerling, and Michael Strube. Fine-grained entity typing in hyperbolic space. In *RepL4NLP@ACL*, pages 169–180. Association for Computational Linguistics, 2019.

[15] Maximilian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. In *NIPS*, pages 6338–6347, 2017.

[16] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543. ACL, 2014.

[17] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *NAACL-HLT*, pages 2227–2237. Association for Computational Linguistics, 2018.

[18] Deepak Ravichandran and Eduard Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 41–47. Association for Computational Linguistics, 2002.

[19] Xiang Ren, Wenqi He, Meng Qu, Lifu Huang, Heng Ji, and Jiawei Han. AFET: automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP*, pages 1369–1378. The Association for Computational Linguistics, 2016.

[20] Xiang Ren, Wenqi He, Meng Qu, Clare R. Voss, Heng Ji, and Jiawei Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD*, pages 1825–1834. ACM, 2016.

[21] Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. An attentive neural architecture for fine-grained entity type classification. In *AKBC@NAACL-HLT*, pages 69–74. The Association for Computer Linguistics, 2016.

[22] Ralph Wieschedel and Ada Brunstein. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia*, 112, 2005.

[23] Ralph Wieschedel, Sameer Pradhan, Lance Ramshaw, Mar tha Palmer, Nianwen Xue, Mitchell Marcus, Ann Taylor, Craig Greenberg, Eduard Hovy, Robert Belvin, et al. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*, 2011.

[24] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Qili Zhu. Probase: a probabilistic taxonomy for text understanding. In *SIGMOD Conference*, pages 481–492. ACM, 2012.

[25] Ji Xin, Hao Zhu, Xu Han, Zhiyuan Liu, and Maosong Sun. Put it back: Entity typing with language model enhancement. In *EMNLP*, pages 993–998. Association for Computational Linguistics, 2018.

[26] Peng Xu and Denilson Barbosa. Neural fine-grained entity type classification with hierarchy-aware loss. In *NAACL-HLT*, pages 16–25. Association for Computational Linguistics, 2018.

[27] Yadollah Yaghoobzadeh, Heike Adel, and Hinrich Schütze. Noise mitigation for neural entity typing and relation extraction. *arXiv preprint arXiv:1612.07495*, 2016.

[28] Dani Yogatama, Daniel Gillick, and Nevena Lazic. Embedding methods for fine grained entity type classification. In *ACL (2)*, pages 291–296. The Association for Computer Linguistics, 2015.

[29] Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. Hyena-live: Fine-grained online entity type classification from natural-language text. In *ACL (Conference System Demonstrations)*, pages 133–138. The Association for Computer Linguistics, 2013.