Fine-Grained Named Entity Typing over Distantly Supervised Data Based on Refined Representations*

Muhammad Asif Ali,[1] Yifang Sun,[1] Bing Li,[1] Wei Wang[1,2]
[1]School of Computer Science and Engineering, UNSW, Australia
[2]College of Computer Science and Technology, DGUT, China
{muhammadasif.ali, bing.li}@unsw.edu.au, {yifangs, weiw}@cse.unsw.edu.au

Abstract

Fine-Grained Named Entity Typing (FG-NET) is a key component in Natural Language Processing (NLP). It aims at classifying an entity mention into a wide range of entity types. Due to a large number of entity types, distant supervision is used to collect training data for this task, which noisily assigns type labels to entity mentions irrespective of the context. In order to alleviate the noisy labels, existing approaches on FG-NET analyze the entity mentions entirely independent of each other and assign type labels solely based on mention's sentence-specific context. This is inadequate for highly overlapping and/or noisy type labels as it hinders information passing across sentence boundaries. For this, we propose an edge-weighted attentive graph convolution network that refines the noisy mention representations by attending over corpus-level contextual clues prior to the end classification. Experimental evaluation shows that the proposed model outperforms the existing research by a relative score of upto 10.2% and 8.3% for macro-f1 and micro-f1 respectively.

1 Introduction

Named Entity Typing (NET) aims at classifying an entity mention to a set of entity types (e.g., person, location and organization) based on its context. It is one of the crucial components in NLP, as it helps in numerous down streaming applications, e.g., information retrieval (Carlson et al. 2010), Knowledge Base Construction (KBC) (Dong et al. 2014), question answering (Lee et al. 2006), machine translation (Britz et al. 2017), etc. Fine-Grained Named Entity Typing (FG-NET) is an extension of traditional NET to a much wide range of entity types (Corro et al. 2015; Ren et al. 2016a), typically over hundred types arranged in a hierarchical structure. It has shown promising results in different applications including KBC (Dong et al. 2014), relation extraction (Mitchell et al. 2018), etc.

In FG-NET, an entity mention is labeled with multiple overlapping entity types based on the context. For instance, in the sentence: *"After having recorded his role, Trump spent the whole day directing the movie." Trump* can be annotated as both *actor* and *director* at the same time. Owing to a broad range of highly correlated entity types with small contextual differences (Section 4.6), manual labeling is error-prone and time-consuming, thus distant supervision is widely used to automatically acquire the training data. Distant supervision follows a two-step approach, i.e., detecting the entity mentions followed by assigning type labels to the mentions using existing knowledge bases. However, it assigns type labels irrespective of the mention's context, which results in high label noise (Ren et al. 2016b). This phenomenon is illustrated in Figure 1, where, for the sentences denoted as: S1:S4, the entity mention *"Imran Khan"* is labeled with all possible labels in the knowledge-base {*person, author, athlete, coach, politician*}. Whereas, from the contextual perspective, in S1 the mention should be labeled as {*person, athlete*}; in S2 it should be assigned labels {*person, author*}, etc. This label noise propagates in model learning, which hinders the improvement in performance.

In an attempt to deal with the noisy training data, existing research on FG-NET relies on the following different approaches: (i) assume all labels to be correct (Ling and Weld 2012; Yogatama, Gillick, and Lazic 2015), which severely affects the model performance; (ii) apply different pruning heuristics to prune the noisy labels (Gillick et al. 2014), however, these heuristics drastically reduce the size of training data; (iii) bifurcate the training data into two categories: clean and noisy, if the type labels correspond to the same type path or otherwise (Ren et al. 2016a; Abhishek, Anand, and Awekar 2017), they ignore the fact that the labels, even corresponding to the same type path, may be noisy. For these approaches, it is hard to guarantee that underlying modeling assumptions will have a substantial impact on alleviating the label noise. In addition, these approaches model the entity mentions entirely independent of each other, which hinders effective propagation of label-specific contextual information across noisy entity mentions.

In order to address the challenges associated with the noisy training data, we introduce a novel approach that puts an equal emphasis on analyzing the entity mentions *w.r.t* label-specific corpus-level context in addition to the sentence-specific context. Specifically, we propose Fine-Grained named Entity Typing with Refined Representations (FGET-RR), shown in Figure 2. FGET-RR initially uses mention's sentence-specific context to generate the noisy mention representation (Phase-I). Later, it uses corpus-level contextual clues to form a sparse graph that surrounds a sub-

*M.A. ALi and W. Wang are the co-corresponding authors.

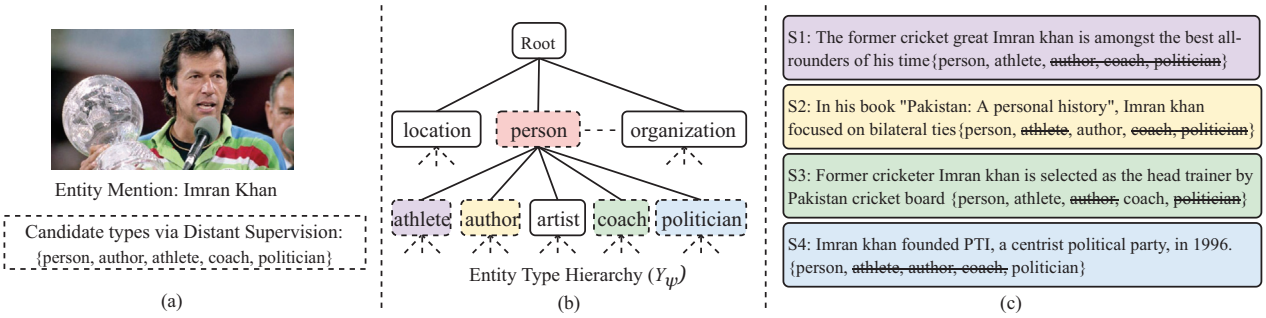

Figure 1: (a) Entity mention and candidate entity types acquired via distant supervision, (b) Target Entity Type Hierarchy (c) Noisy training data with irrelevant entity types struck-through.

set of noisy mentions with a set of confident mentions having high contextual overlap. And, performs edge-weighted attentive graph convolutions to recompute/refine the representation of noisy mention as an aggregate of the confident neighboring mentions lying at multiple hops (Phase-II). Finally, the refined mention representation is embedded along with the type label representations for entity typing.

We argue that the proposed framework has following advantages: (i) it allows appropriate information sharing by efficient propagation of corpus-level contextual clues across noisy mentions; (ii) it analyzes the aggregated label-specific context, which is more refined compared with the noisy mention-specific context; (iii) it effectively correlates the local (sentence-level) and the global (corpus-level) context to refine mention's representation, required to perform the end-task in a robust way. We summarize the major contributions of this paper as follows:

- We introduce FGET-RR, a novel approach for FG-NET that pays an equal importance on analyzing the entity mentions with respect to the corpus-level context in addition to the sentence-level context to perform entity typing in a performance-enhanced fashion.
- We propose an edge-weighted attentive graph convolution network to refine the noisy mention representations. To the best of our knowledge, this is the first work that, in contrast to the existing models that de-noise the data at model's input, refines the representations learnt over distantly supervised data.
- We demonstrate the effectiveness of the proposed model by comprehensive experimentation. FGET-RR outperforms the existing research by a margin of upto 10.2% and 8.3% in terms of macro-f1 and micro-f1 scores respectively.

2 Related Work

Earlier research on NET relies on assigning entity mentions to a small number of entity types, i.e., person, location, organization, etc. (Sang and Meulder 2003). In the recent decade, the traditional NET is extended to a wide range of fine-grained entity types (Ling and Weld 2012; Yosef et al. 2013). All the systems in FG-NET majorly focus on entity typing only, i.e., they assume that the mention boundaries have been pre-identified. Yogatama, Gillick, and Lazic, (2015) used embeddings to jointly embed entity mentions and the type information. Gillick et al., (2014) proposed pruning heuristics to prune the noisy mentions. Corro et al., (2015) introduced the most fine-grained system so far, with types encompassing Word-Net Hierarchy (Miller 1998). Ren et al., (2016a) introduced Automated Fine-grained named Entity Typing (AFET) using a set of hand-crafted features to represent mention, later jointly embed the feature vectors and the label vectors for classification.

Shimaoka et al., (2016) used an averaging encoder to encode the entity mention, bi-directional LSTM to encode the context, followed by attention to attend over label-specific context. Inui et al., (2017) extended (Shimaoka et al. 2016) by incorporating hand-crafted features along with attention. Abhishek, Anand, and Awekar, (2017) used end-to-end architecture to encode entity mention and its context. Xu and Barbosa, (2018) modified the FG-NET problem definition from multi-label classification to single-label classification problem with a hierarchical aware loss to handle noisy data. Xin et al., (2018) proposed FG-NET based on language models that compute compatibility between the type labels and the context to eliminate inconsistent types.

Graph Convolution Networks (GCNs) have received considerable research attention in the recent past. GCNs extend the convolutions from regular grids to graph-structured data in spatial and/or spectral domain. They are widely been used in classification settings, i.e., both semi-supervised (Kipf and Welling 2017), and supervised (Yao, Mao, and Luo 2019). While GCNs have successfully been used for image de-noising (Valsesia, Fracastoro, and Magli 2019), we are the first to effectively utilize it to refine the representations learnt over noisy text data.

3 The Proposed Model

3.1 Problem Definition

In this paper, we aim to build a multi-label, multi-class entity typing system that can use distantly supervised data to classify an entity mention into a set of fine-grained entity types based on the context. Specifically, we refine the representations learnt on the noisy data prior to entity typing. Similar to the existing research (used for comparative

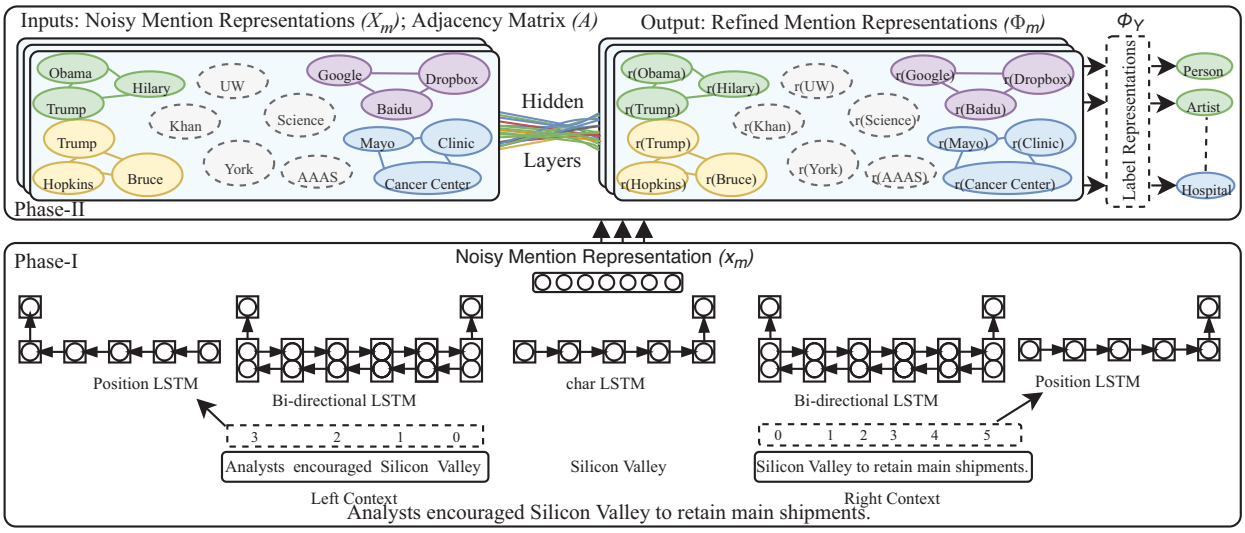

Figure 2: Proposed model for FG-NET (FGET-RR); Phase-I learns mention's representation based on local sentence-specific context; Phase-II refines the representations learnt in Phase-I by sharing corpus-level type-specific context

evaluation in Table 1), we assume the availability of training data D_{train} acquired via distant supervision and manually labeled test data D_{test}. Formally, the data set D is a set of sentences/paragraphs for which the entity mentions $\{m_i\}_{i=1}^{N}$ (tokens corresponding to the entities), the context $\{c_i\}_{i=1}^{N}$ and the candidate type labels $\{y_i\}_{i=1}^{N} \in \{0,1\}^Y$ (Y-dimensional binary vector with $y_{i,t} = 1$ if t^{th} type corresponds to the true label and zero otherwise) have been pre-identified. Here, the type labels correspond to type hierarchy in the knowledge base ψ with the schema Y_ψ. We represent the data as a set of triples $D = \{(m_i, c_i, y_i)\}_{i=1}^{N}$. Following (Ren et al. 2016a), we bifurcate the training mentions M_{train} into clean M_{clean} and noisy M_{noisy} depending upon if the mention's type path corresponds to a single path in Y_ψ or otherwise. For example, considering the type-path in Figure 1 (b), a mention with labels {*person, athlete, author*} will be considered as a noisy, whereas, a mention with labels {*person, artist*} will be considered as clean.

3.2 Overview

Our proposed model (shown in Figure 2) consists of two phases: in Phase-I, we learn local context-dependent noisy mention representations using LSTM networks (Hochreiter and Schmidhuber 1997). In Phase-II, we form a sparse graph that takes the representations learnt in Phase-I as input and perform edge-weighted attentive graph convolutions to refine these representations. Finally, we embed the refined mention representations along with the label representations for FG-NET.

We argue that the proposed two-phase design has the following advantages: (i) it allows us to quantify the contribution of each phase, as it provides the maximal flexibility to correlate and/or analyze these phases independently, (ii) it enables effective propagation of corpus-level contextual information that facilitates refinement of noisy mention representations.

3.3 Phase-I (Noisy Mention Representations)

Phase-I follows a standard approach with multiple LSTM networks to encode sequential text data. We use $\overrightarrow{x}$ and $\overleftarrow{x}$ to represent the left-to-right and the right-to-left LSTM encodings. The components of Phase-I are explained as follows:

Mention Encoder: To encode the morphological structure of entity mentions, we first decompose the mention into character sequence. Later, use a standard LSTM network to encode the character sequence. We use $\phi_{men} = [\overrightarrow{men_{char}}] \in \mathbf{R}^d$ to represent the encoded mention.

Context Encoder: In order to encode the context, we use bidirectional LSTMs to encode the tokens corresponding to the left and the right context of the entity mention, as shown in Figure 2. Note that for each bi-directional LSTM, we feed mention tokens along with the context to get the context encoding. The motivation is to analyze the context in relation with the entity mention. We use $\phi_{left} = [\overleftarrow{c_{left}}; \overrightarrow{c_{left}}] \in \mathbf{R}^c$, and $\phi_{right} = [\overleftarrow{c_{right}}; \overrightarrow{c_{right}}] \in \mathbf{R}^c$ to represent bi-directional encoding of the left and the right context respectively.

Position Encoder: The position feature is used to encode the relative distance between the mention and the contextual words using LSTM network. Previously, this feature has shown good performance in relation classification (Zeng et al. 2014). We use $\phi_{lpos} = [\overleftarrow{left_{pos}}] \in \mathbf{R}^p$ and $\phi_{rpos} = [\overrightarrow{right_{pos}}] \in \mathbf{R}^p$ to encode the relative positions of the left and the right contextual tokens.

Mention Representation: Finally, we concatenate all the mention-specific encodings to get the noisy mention representation: $x_m \in \mathbf{R}^f$, where $f = d + 2 * c + 2 * p$

$$x_m = [\phi_{lpos}; \phi_{left}; \phi_{men}; \phi_{right}; \phi_{rpos}] \quad (1)$$

3.4 Phase-II (Refining Mention Representations)

In order to refine the noisy mention representations learnt on distantly supervised data (Phase-I), we propose an edge-weighted attentive Graph Convolution Network (GCN). GCN extends convolution from regular structured grids to arbitrary graphs. We exploit the fact that for a given graph node, the GCN uses the information contained in the neighboring nodes to come up with a new representation of the node. For this, we construct an undirected graph with nodes as entity mentions and enforce the mentions with high contextual overlap to be adjacent to each other by forming edges. Formally, let $G = (V, E)$ be a graph with $|V| = n$ nodes (entity mentions); $|E|$ edges; we use A to denote its symmetric adjacency matrix. The construction of G is outlined in Algorithm 1 and explained as follows:

Graph Construction: Firstly, we learn $1024d$ deep contextualized ELMO embeddings (Peters et al. 2018) for all the sentences in data set D. We average out the embedding vectors corresponding to the mention tokens to acquire context-dependent mention embeddings $ELMO_{men}$. Later, for the training data D_{train}, we compute pivot vectors, i.e., $\{Pivot_y\}_{y=1}^{Y}$, as representatives for each entity type $y \in Y$, by averaging the mention embeddings corresponding to the type y ($ELMO_{men_y}$). We use these pivot vectors to capture confident mention candidates for each entity type $\{Candidates\}_{y=1}^{Y}$, i.e., the mentions with high contextual overlap having $cos(ELMO_{men}, \{Pivot_y\}_{y=1}^{Y}) \geq thr$, as illustrated in lines (7-13) of Algorithm 1. We observed that a reasonably high value for the threshold thr offers the following benefits: (i) avoids computational overhead, (ii) captures only the most confident mention candidates. Finally, for the candidate mentions corresponding to each entity type $\{Candidates\}_{y=1}^{Y}$, we form pairwise edges to construct the graph G, with adjacency matrix A (line 14-16).

Attentive Aggregation: Depending upon the value of thr, the graph G surrounds a subset of nodes (noisy mentions), with a set of confident mentions having high type-specific contextual overlap, by forming edges. Later, for noisy mention representations, we aggregate the information contained in the neighbors to come up with the refined mention representations. Specifically, unlike the existing work (Kipf and Welling 2017), we propose an edge-weighted attentive graph convolution network that uses the following layer-wise propagation rule:

$$L^{(1)} = \rho(\widetilde{\eta_{ij} \odot} A L^{(0)} W_0) \quad (2)$$

where η_{ij} is the attention term computed via pairwise similarity of the context-dependent mention embeddings, i.e., $cos(ELMO_{men^i}, ELMO_{men^j})\ \forall (i,j) \in V$; $\eta_{ij} \odot A$ is the Hadamard product of attention weights and the adjacency matrix; $\widetilde{\eta_{ij} \odot} A = \tilde{D}^{-1/2}((\eta_{ij} \odot A) + I)\tilde{D}^{-1/2}$ is the normalized symmetric matrix; $\tilde{D}$ is the degree matrix of $(\eta_{ij} \odot A)$; $L^{(0)}$ is input from the previous layer, in our case: $L^{(0)} = X_m \in \mathbf{R}^{N \times f}$ is the matrix corresponding to the noisy mentions' representations from Equation (1), ρ is the activation function and W_0 is the matrix of learn-able parameters. Note, by adding identity matrix I to $(\eta_{ij} \odot A)$, the model assumes that every node $v \in V$ is connected to itself, i.e, $(v, v) \in E$. We observed, that for our problem, this simple symmetrically normalized edge-weighted attentive formulation outperforms the attention formulation of (Bahdanau, Cho, and Bengio 2015). We can accumulate information from the higher-order neighborhood by stacking multiple layers:

$$L^{(i+1)} = \rho(\widetilde{\eta_{ij} \odot} A L^{(i)} W_i) \quad (3)$$

where i corresponds to the layer no., with $L^{(0)} = X_m$. For our model, we use a two-layered network to learn the refined mention representations $\Phi_m \in \mathbf{R}^{N \times k}$ as follows:

$$\Phi_m = \widetilde{\eta_{ij} \odot} A(ReLU(\widetilde{\eta_{ij} \odot} A X_m W_0))W_1 \quad (4)$$

3.5 The Complete Model

Let $\phi_m \in \mathbf{R}^k$ be a refined mention representation and $\{\phi_y\}_{y=1}^{Y} \in \mathbf{R}^k$ be the type label representations. For classification, we embed these representations in the same space. For this, we learn a function $f(\phi_m, \phi_y) = \phi_m^T.\phi_y + bias_y$ that incorporates label bias $bias_y$ in addition to the label and the refined mention representations. We extend loss functions from our previous work (Ali et al. 2019) to separately model the clean and the noisy entity mentions, as explained below:

Algorithm 1 Graph Construction

Input: Embeddings $(ELMO_{men})$; $D = D_{train} + D_{test}$
Output: Graph: G

1: $\{Pivot_y\}_{y=1}^{Y} \leftarrow \mathbf{0}$; $G \leftarrow \emptyset$
2: **for** $men \leftarrow 1$ to D_{train} **do**
3: **for** $y \in men_{labels}$ **do**
4: $Pivot_y \leftarrow Pivot_y + ELMO_{men_y}$
5: **end for**
6: **end for**
7: $\{Candidates\}_{y=1}^{Y} \leftarrow \emptyset$
8: **for** $men \leftarrow 1$ to D **do**
9: $y^* = \text{argmax}_{y \in Y} \cos(ELMO_{men}, Pivot_y)$
10: **if** $\cos(ELMO_{men}, Pivot_{y^*}) \geq thr$ **then**
11: $Candidates_{y^*} \leftarrow Candidates_{y^*} \cup men$
12: **end if**
13: **end for**
14: **for** $y \leftarrow 1$ to Y **do**
15: $G \leftarrow G \cup \{edge(v_1, v_2) \in Candidates_y\}$
16: **end for**
17: **return** G

Loss Function for clean data: In order to model the clean entity mentions M_{clean}, we use a margin-based loss to embed the mention representation close to its true label representations, while at the same time pushing it away from the false type labels. The loss function for modeling the clean entity mentions is shown in Equation (5).

$$L_{clean} = \sum_{y \in T_y} \mathsf{ReLU}(1 - f(\phi_m, \phi_y)) + \sum_{y' \in T'_y} \mathsf{ReLU}(1 + f(\phi_m, \phi_{y'})) \quad (5)$$

where T_y represents the true labels and $T_{y'}$ represents the false labels in Y_ψ.

Loss Function for noisy data: In order to model the noisy entity mentions M_{noisy}, we use a variant of the loss function in Equation 5 to focus on the most relevant label among noisy type labels. The loss function for modeling the noisy entity mentions is illustrated in Equation (6).

$$L_{noisy} = \mathsf{ReLU}(1 - f(\phi_m, \phi_{y^*})) + \sum_{y' \in T'_y} \mathsf{ReLU}(1 + f(\phi_m, \phi_{y'})) \quad (6)$$
$$y^* = \underset{y \in T_y}{\operatorname{argmax}}\, f(\phi_m, \phi_y)$$

where y^* corresponds to the most relevant label among the set of noisy labels, T_y represents the set of noisy labels and $T_{y'}$ represents the false labels in Y_ψ.

Finally, we minimize L_{noisy} + L_{clean} as the loss function of FGET-RR.

Model Training and Inference: Owing to the adjacency matrix A involved in Phase-II, our current implementation trains two phases iteratively. Specifically, we repeat the following process till convergence: (i) perform mini-batch Stochastic Gradient Descent (SGD) in Phase-I, (ii) concatenate the noisy representations learnt in Phase-I (i.e., X_m) and perform gradient descent in Phase-II. We leave an appropriate formulation of SGD for Phase-II as future work. For inference, we use mention's refined representation ϕ_m and carry-out a top-down search in the type-hierarchy, i.e., we recursively select the type y that yields the best score $f(\phi_m, \phi_y)$ until we hit a leaf node or the score falls below a threshold of zero.

4 Experiments

4.1 Dataset

For evaluation, we use publicly available data sets provided by (Ren et al. 2016a). Table 2 shows the statistics of these data sets. A detailed description is as follows:

Wiki/Figer: Its training data consists of Wikipedia sentences automatically labeled via distant supervision by mapping the entity mentions to Freebase types (Bollacker et al. 2008). The testing data consists of news reports manually labeled by (Ling and Weld 2012).

OntoNotes: It consists of sentences from newswire documents contained in OntoNotes corpus (Weischedel et al. 2011) mapped to Freebase types via DBpedia Spotlight (Daiber et al. 2013). The testing data is manually annotated by (Gillick et al. 2014).

BBN: It consists of sentences from the Wall Street Journal annotated by (Weischedel and Brunstein 2005). The training data is annotated using DBpedia Spotlight.

4.2 Experimental Settings:

In order to come up with a unanimous platform for comparative evaluation, we use the priorly defined data split by the existing models to training, test and dev sets. The training data is used for model training (i.e., learning noisy representations in Phase-I and refinement in Phase-II). The dev set is used for parameter tuning and the model performance is reported on the test set. All the experiments are performed on Intel Xenon Xeon(R) CPU E5-2640 (v4) with 256 GB main memory and Nvidia Titan V GPU.

Hyperparameters: We separately analyze the performance of FGET-RR using $300d$ Glove embeddings (Pennington, Socher, and Manning 2014) and $1024d$ deep contextualized ELMO embeddings (Peters et al. 2018). Character, position and label embeddings are randomly initialized. For position and bi-directional context encoders, the hidden layer size of LSTM is set to $100d$. For mention encoder the hidden layer size is $200d$. Maximum sequence length is set to 100. For Phase-II, we use graphs with 1.6M, 0.6M, 5.4M edges for Wiki, Ontonotes, and BBN respectively. For model training, we use Adam optimizer (Kingma and Ba 2015) with learning rate (0.0008-0.001).

4.3 Baseline Models / Model Comparison

We compare FGET-RR with the existing state-of-the-art research on FG-NET, namely: (i) **FIGER** (Ling and Weld 2012), (ii) **HYENA** (Yosef et al. 2013), (iii) **AFET** (Ren et al. 2016a) and its variants **AFET-NoCo**, **AFET-NoPa**, **AFET-CoH**, (iv) **Attentive** (Shimaoka et al. 2016), (v) **FNET** (Abhishek, Anand, and Awekar 2017), and (vi) **NFGEC+LME** (Xin et al. 2018) [1]. For all these models, we use the scores reported in the published papers, as they are computed using the same data settings as that of ours.

Note that our model is not comparable with the implementation of Xu and Barbosa (2018), because Xu and Barbosa changed the FG-NET problem definition to single-label classification problem and updated the training and testing data accordingly. It is hard to transform their work for multi-label, multi-class classification settings.

4.4 Main Results

We compare the results of our proposed approach (FGET-RR) with the baseline models in Table 1. We boldface the overall best scores with the previous state-of-the-art

[1] We used code shared by authors to compute results for BBN data.

	Wiki			OntoNotes			BBN		
	strict	mac-F1	mic-F1	strict	mac-F1	mic-F1	strict	mac-F1	mic-F1
FIGER (Ling and Weld 2012)	0.474	0.692	0.655	0.369	0.578	0.516	0.467	0.672	0.612
HYENA (Yosef et al. 2013)	0.288	0.528	0.506	0.249	0.497	0.446	0.523	0.576	0.587
AFET-NoCo (Ren et al. 2016a)	0.526	0.693	0.654	0.486	0.652	0.594	0.655	0.711	0.716
AFET-NoPa (Ren et al. 2016a)	0.513	0.675	0.642	0.463	0.637	0.591	0.669	0.715	0.724
AFET-CoH (Ren et al. 2016a)	0.433	0.583	0.551	0.521	0.680	0.609	0.657	0.703	0.712
AFET (Ren et al. 2016a)	0.533	0.693	0.664	0.551	0.711	0.647	0.670	0.727	0.735
Attentive (Shimaoka et al. 2016)	0.581	0.780	0.744	0.473	0.655	0.586	0.484	0.732	0.724
FNET-AllC (Abhishek, Anand, and Awekar 2017)	0.662	0.805	0.770	0.514	0.672	0.626	0.655	0.736	0.752
FNET-NoM (Abhishek, Anand, and Awekar 2017)	0.646	0.808	0.768	0.521	0.683	0.626	0.615	0.742	0.755
FNET (Abhishek, Anand, and Awekar 2017)	0.658	0.812	0.774	0.522	0.685	0.633	0.604	0.741	0.757
NFGEC+LME (Xin et al. 2018)	0.629	0.806	0.770	0.529	0.724	0.652	0.607	0.743	0.760
FGET-RR Phase I-II (Glove + Context Encoders)	0.674	0.817	0.777	0.567	0.737	0.680	**0.740**	0.811	0.817
FGET-RR Phase I-II (Contextualized Embeddings)	**0.710**	**0.847**	**0.805**	**0.577**	**0.743**	**0.685**	0.703	**0.819**	**0.823**

Table 1: FGET-RR performance comparison against baseline models

Dataset	Wiki	OntoNotes	BBN
Training Mentions	2.6 M	220398	86078
Testing Mentions	563	9603	13187
% clean mentions (training)	64.58	72.61	75.92
% clean mentions (testing)	88.28	94.0	100
Entity Types	128	89	47
Max hierarchy depth	2	3	2

Table 2: Fine-Grained Named Entity Typing data sets

underlined. These results show that FGET-RR outperforms all the previous state-of-the-art research by a large margin. Especially noteworthy is the performance of our model on the BBN data outclassing the existing models by a margin of 10.4%, 10.2% and 8.3% in strict accuracy, macro-f1, and micro-f1 respectively. For OntoNotes, our model yields 5.1% improvement in micro-f1 compared to the previous best by Xin et al. (2018). For Wiki data, the FGET-RR improves the performance by 7.5%, 4.3% and 4.0% in strict accuracy, macro-f1, and micro-f1 respectively. Such promising results show that refining the mention representations via corpus-level contextual clues help in alleviating the label noise associated with the distantly supervised data.

4.5 Ablation study

We present a detailed ablation analysis of the FGET-RR in Table 3. Note that we only report the results for the Glove embeddings along with the context encoders. A similar trend is observed by replacing the Glove embeddings and contextual encoders with the deep contextualized ELMO embeddings.

For Phase-I, we analyze the role of position encoder in addition to the mention and the context encoders. We also compare the results of our model with noisy mention representations (Phase-I) with that of refined representations (Phase I-II). For Phase I-II, we examine the impact of variants of the adjacency matrix on representations' refinement, namely: (i) random adjacency matrix (FGET-RR + RND); (ii) identity as the adjacency matrix (FGET-RR + EYE); (iii) adjacency matrix based on pivots, i.e., $\eta_{ij} = 1$ (FGET-RR + PIVOTS); and (iv) edge-weighed attention along with the adjacency matrix (FGET-RR + ATTN).

First two rows in Table 3 show that the *position* feature slightly improved the performance for all data sets, yielding higher scores across all categories. Comparing the results for different adjacency matrices, we observe that the identity matrix didn't play a significant role in improving the model performance. For randomly generated adjacency matrices, a decline in performance shows that when the structure in data is lost the graph convolution layers are no longer useful in de-noising the representations. Note that we use randomly generated adjacency matrices with an equivalent number of edges as that of the original graphs (Section 4.2).

On the contrary, the models with adjacency matrices acquired from type-specific pivots, (FGET-RR + PIVOTS; FGET-RR + ATTN) show a substantial improvement in the performance, which is governed by an appropriate refinement of the noisy representations trained on distantly supervised training data. Especially, the edge-weighted attention yields a much higher score, because attention helps the model to reinforce the decision by attending over the contribution of each neighbor. Overall, the results show that the refined representations indeed augment the model performance by a significant margin.

4.6 Analyses

In this section, we analyze the effectiveness of the refined representations (Phase-II), followed by a detailed analysis of the error cases.

Effectiveness of Phase-II (BBN data): In order to analyze the effectiveness of the refined representations (Phase-II), we perform a comparative performance analysis for the most frequent entity types in the BBN dataset.

As shown in Table 4, Phase-II has a clear dominance over Phase-I across all entity types. The major reason for poor performance in Phase-I is highly correlated nature of entity types with substantial contextual overlap. The distant supervision, moreover, adds to the intricacy of the problem. For example, *"organization"* and *"cooperation"* are two highly correlated entity types, *"organization"* is a generic term, whereas, the *"corporation"* being a sub-type, is more intended towards an enterprise and/or company with some business-related concerns, i.e., every corporation is an organization but not otherwise. Analyzing the corpus-level additive lexical contrast along the type-hierarchy revealed that in addition to sharing the context of the *"organization"*, the context of the *"corporation"* is more oriented towards

Adjacency Matrix	Model	Wiki			OntoNotes			BBN		
		strict	mac-F1	mic-F1	strict	mac-F1	mic-F1	strict	mac-F1	mic-F1
Phase-I	*mention + context*	0.649	0.802	0.762	0.521	0.690	0.632	0.612	0.746	0.759
	mention + context + position	0.661	0.807	0.767	0.531	0.694	0.638	0.616	0.755	0.765
Phase I-II (FGET-RR + RND)	*mention + context + position + GCN*	0.642	0.797	0.755	0.464	0.641	0.595	0.617	0.693	0.709
Phase I-II (FGET-RR + EYE)	*mention + context + position + GCN*	0.664	0.812	0.773	0.519	0.681	0.630	0.659	0.754	0.766
Phase I-II (FGET-RR + PIVOTS)	*mention + context + position + GCN*	0.672	0.815	0.775	**0.570**	0.735	0.675	0.736	0.804	0.810
Phase I-II (FGET-RR + ATTN)	*mention + context + position + GCN*	**0.674**	**0.817**	**0.777**	0.567	**0.737**	**0.680**	**0.740**	**0.811**	**0.817**

Table 3: Ablation study for FGET-RR using Glove + Context Encoder

Labels	Support	Phase-I			Phase I-II		
		Prec	Rec	F1	Prec	Rec	F1
/organization	45.30%	0.837	0.850	0.843	0.924	0.842	**0.881**
/organization/corporation	35.70%	0.824	0.759	0.790	0.921	0.779	**0.844**
/person	22.00%	0.746	0.779	0.762	0.86	0.886	**0.872**
/gpe	21.30%	0.878	0.831	0.853	0.924	0.845	**0.883**
/gpe/city	9.17%	0.737	0.738	0.738	0.802	0.767	**0.784**

Table 4: Phase-I vs Phase (I-II) comparison for BBN data

Noisy Representations (Phase-I)		Refined Representations (Phase I-II)	
Mention	Label	Mention	Label
Yves Goupil	/person	Ford Motor	/organization/corporation
Berg	/person	Vauxhall Motors Ltd.	/organization/corporation
Volokhs	/person	Chrysler Corp.	/organization/corporation
lawns	/plant	Advanced Micro Devices Inc.	/organization/corporation
Rafales	/product/vehicle,/product	Chesebrough-Pond s Inc.'	/organization/corporation

Table 5: Top 5-nearest neighboring representations (noisy and refined) for the representation of mention *"Maytag"*

tokens: *cents*, *business*, *stake*, *bank*, etc. However, for distantly supervised data, it is hard to ensure such a distinctive distribution of contextual tokens for each entity mention. In addition, the lack of information sharing among these distinctive tokens across sentence boundaries leads to poor performance for Phase-I, which makes it a much harder problem from the generalization perspective. A similar scenario holds for other entity types, e.g., *actor* vs *artist* vs *director*, etc. Whereas, after sharing type-specific contextual clues, we can see a drastic improvement in the performance for phase I-II, i.e., F1 = 0.844 compared to F1 = 0.790 in phase-I for *"corporation"*, shown in Table 4.

To further verify our claim, we analyze the nearest neighbors for both the noisy and the refined representation spaces along with the corresponding labels. An example in this regard is shown in Table 5, where we illustrate the neighboring representations corresponding to the representation of the mention *"Maytag"* from the sentence: *"We have officials from giants like Du Pont and **Maytag**, along with lesser knowns like Trojan Steel and the Valley Queen Cheese Factory."* with true context-dependent label as *"organization/corporation"*. The neighboring representations corresponding to the noisy representation space are dominated by irrelevant labels, having almost no correlation with the original mention's label. Whereas, for the refined representations, almost all the neighboring representations carry the same label as that of mention *"Maytag"*. This ascertains that the refined representations are more semantically oriented *w.r.t* context, which enables them to accommodate more distinguishing information for entity typing compared to that of the noisy representations.

These analyses strengthen our claim that FGET-RR enables representation refinement and/or label smoothing by implicitly sharing corpus-level contextual clues across entity mentions. This empowers FGET-RR to indeed learn across sentence boundaries, which makes it more robust compared with the previous state-of-the-art methods that classify entity mentions entirely independent of each other.

Error Cases: We categorize the errors into two major categories: (i) missing labels, and (ii) erroneous labels. Missing labels correspond to the entity mentions for which type labels are not predicted by the model, thus effecting the recall, while erroneous labels correspond to mis-labeled instances, effecting the precision. Following the results for the BBN data in Table 4, most of the errors (for phase-I and I-II) correspond to the labels *"/organization/corporation"* and *"organization"*.

For missing labels, most of them are attributed to the cases, where type labels are entirely dictated by the names of corporations, with very little information contained in the context. For example, in the sentence: *"That has got to cause people feel a little more optimistic, says Glenn Cox the correspondence officer of Mcjunkin"*, the entity mention *"Mcjunkin"* is labeled *"organization/corporation"*. For such cases, type information is not explicit from the context. This is also evident by a relatively low recall score for both Phase-I and Phase I-II, shown in Table 4.

Likewise, most of the erroneous labels (esp., Phase-I) are caused by the overlapping context for highly correlated entity types, e.g., *"organization"* and *"corporation"*, as explained previously. This problem was somehow eradicated by refining the representations in Phase-II, as is evident by a higher change in precision for Phase I-II relative to that of Phase-I. A similar trend was observed for OntoNotes and Wiki data. Other limiting factors of the proposed model include: (i) the assumption that ELMO embeddings are able to capture distinctive mention representation based on the context, (ii) acquiring pivot vectors from noisy data, which did some smoothing but didn't completely rectify the noise.

5 Conclusions and Future Work

In this paper, we propose FGET-RR, a novel approach for FG-NET that outperforms existing research by a large margin. In the future, we will augment the proposed framework by explicitly identifying type-specific clauses to perform edge conditioned representations' refinement.

Acknowledgements. This work is supported by ARC DPs 170103710 and 180103411, D2DCRC DC25002 and DC25003. The Titan V used for this research was donated by the NVIDIA Corporation.

References

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

Ali, M. A.; Sun, Y.; Zhou, X.; Wang, W.; and Zhao, X. 2019. Antonym-synonym classification based on new sub-space embeddings. In *AAAI*, 6204–6211. AAAI Press.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Bollacker, K. D.; Evans, C.; Paritosh, P.; Sturge, T.; and Taylor, J. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD Conference*, 1247–1250. ACM.

Britz, D.; Goldie, A.; Luong, M.; and Le, Q. V. 2017. Massive exploration of neural machine translation architectures. *CoRR* abs/1703.03906.

Carlson, A.; Betteridge, J.; Wang, R. C.; Jr., E. R. H.; and Mitchell, T. M. 2010. Coupled semi-supervised learning for information extraction. In *WSDM*, 101–110. ACM.

Corro, L. D.; Abujabal, A.; Gemulla, R.; and Weikum, G. 2015. FINET: context-aware fine-grained named entity typing. In *EMNLP*, 868–878. The Association for Computational Linguistics.

Daiber, J.; Jakob, M.; Hokamp, C.; and Mendes, P. N. 2013. Improving efficiency and accuracy in multilingual entity extraction. In *I-SEMANTICS*, 121–124. ACM.

Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; and Zhang, W. 2014. Knowledge vault: a web-scale approach to probabilistic knowledge fusion. In *KDD*, 601–610. ACM.

Gillick, D.; Lazic, N.; Ganchev, K.; Kirchner, J.; and Huynh, D. 2014. Context-dependent fine-grained entity type tagging. *CoRR* abs/1412.1820.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural Computation* 9(8):1735–1780.

Inui, K.; Riedel, S.; Stenetorp, P.; and Shimaoka, S. 2017. Neural architectures for fine-grained entity type classification. In *EACL (1)*, 1271–1280. Association for Computational Linguistics.

Kingma, D. P., and Ba, J. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR (Poster)*. OpenReview.net.

Lee, C.; Hwang, Y.; Oh, H.; Lim, S.; Heo, J.; Lee, C.; Kim, H.; Wang, J.; and Jang, M. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *AIRS*, volume 4182 of *Lecture Notes in Computer Science*, 581–587. Springer.

Ling, X., and Weld, D. S. 2012. Fine-grained entity recognition. In *AAAI*. AAAI Press.

Miller, G. A. 1998. *WordNet: An electronic lexical database*. MIT press.

Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. 2018. Never-ending learning. *Communications of the ACM* 61(5):103–115.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global vectors for word representation. In *EMNLP*, 1532–1543. ACL.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep contextualized word representations. In *NAACL-HLT*, 2227–2237. Association for Computational Linguistics.

Ren, X.; He, W.; Qu, M.; Huang, L.; Ji, H.; and Han, J. 2016a. AFET: automatic fine-grained entity typing by hierarchical partial-label embedding. In *EMNLP*, 1369–1378. The Association for Computational Linguistics.

Ren, X.; He, W.; Qu, M.; Voss, C. R.; Ji, H.; and Han, J. 2016b. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD*, 1825–1834. ACM.

Sang, E. F. T. K., and Meulder, F. D. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *CoNLL*, 142–147. ACL.

Shimaoka, S.; Stenetorp, P.; Inui, K.; and Riedel, S. 2016. An attentive neural architecture for fine-grained entity type classification. In *AKBC@NAACL-HLT*, 69–74. The Association for Computer Linguistics.

Valsesia, D.; Fracastoro, G.; and Magli, E. 2019. Deep graph-convolutional image denoising. *CoRR* abs/1907.08448.

Weischedel, R., and Brunstein, A. 2005. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium, Philadelphia* 112.

Weischedel, R.; Pradhan, S.; Ramshaw, L.; Palmer, M.; Xue, N.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; Belvin, R.; et al. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Xin, J.; Zhu, H.; Han, X.; Liu, Z.; and Sun, M. 2018. Put it back: Entity typing with language model enhancement. In *EMNLP*, 993–998. Association for Computational Linguistics.

Xu, P., and Barbosa, D. 2018. Neural fine-grained entity type classification with hierarchy-aware loss. In *NAACL-HLT*, 16–25. Association for Computational Linguistics.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *AAAI*, 7370–7377. AAAI Press.

Yogatama, D.; Gillick, D.; and Lazic, N. 2015. Embedding methods for fine grained entity type classification. In *ACL (2)*, 291–296. The Association for Computer Linguistics.

Yosef, M. A.; Bauer, S.; Hoffart, J.; Spaniol, M.; and Weikum, G. 2013. Hyena-live: Fine-grained online entity type classification from natural-language text. In *ACL (Conference System Demonstrations)*, 133–138. The Association for Computer Linguistics.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *COLING*, 2335–2344. ACL.