AFET: Automatic Fine-Grained Entity Typing by Hierarchical Partial-Label Embedding

Xiang Ren[†*] **Wenqi He**[†*] **Meng Qu**[†] **Lifu Huang**[♯] **Heng Ji**[♯] **Jiawei Han**[†]

[†] University of Illinois at Urbana-Champaign, Urbana, IL, USA

[♯] Computer Science Department, Rensselaer Polytechnic Institute, USA

[†]{xren7, wenqihe3, mengqu2, hanj}@illinois.edu [♯]{huangl7, jih}@rpi.edu

Abstract

Distant supervision has been widely used in current systems of fine-grained entity typing to automatically assign categories (entity types) to entity mentions. However, the types so obtained from knowledge bases are often incorrect for the entity mention's local context. This paper proposes a novel embedding method to separately model "clean" and "noisy" mentions, and incorporates the given type hierarchy to induce loss functions. We formulate a joint optimization problem to learn embeddings for mentions and type-paths, and develop an iterative algorithm to solve the problem. Experiments on three public datasets demonstrate the effectiveness and robustness of the proposed method, with an average 15% improvement in accuracy over the next best compared method[1].

1 Introduction

Assigning types (*e.g.*, `person`, `organization`) to mentions of entities in context is an important task in natural language processing (NLP). The extracted entity type information can serve as primitives for relation extraction (Mintz et al., 2009) and event extraction (Ji and Grishman, 2008), and assists a wide range of downstream applications including knowledge base (KB) completion (Dong et al., 2014), question answering (Lin et al., 2012) and entity recommendation (Yu et al., 2014). While traditional named entity recognition systems (Ratinov and Roth, 2009; Nadeau and Sekine, 2007) focus on a small set of coarse types (typically fewer than 10), recent studies (Ling and Weld, 2012; Yosef et al., 2012) work on a much larger set of fine-grained types (usually over 100) which form a tree-structured hierarchy (see the blue region of Fig. 1). Fine-grained typing allows one mention to have multiple types, which together constitute a *type-path* (not necessarily ending in a leaf node) in the given type hierarchy, *depending on the local context* (*e.g.*, sentence). Consider the example in Fig. 1, "*Arnold Schwarzenegger*" could be labeled as {`person`, `businessman`} in S3 (investment). But he could also be labeled as {`person`, `politician`} in S1 or {`person`, `artist`, `actor`} in S2. Such fine-grained type representation provides more informative features for other NLP tasks. For exam-

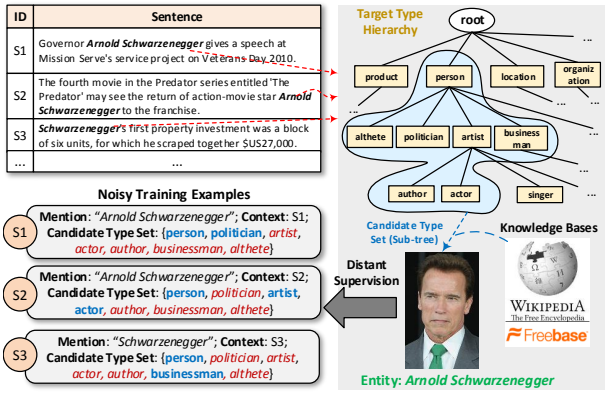


Figure 1: Current systems may detect *Arnold Schwarzenegger* in sentences S1-S3 and assign the same types to all (listed within braces), when only some types are correct for context (blue labels within braces).

[*]Equal contribution.

[1]Codes and datasets used in this paper can be downloaded at `https://github.com/shanzhenren/AFET`.

Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pages 1369–1378,
Austin, Texas, November 1-5, 2016.

ple, since relation and event extraction pipelines rely on entity recognizer to identify possible arguments in a sentence, fine-grained argument types help distinguish hundreds or thousands of different relations and events (Ling and Weld, 2012).

Traditional named entity recognition systems adopt manually annotated corpora as training data (Nadeau and Sekine, 2007). But the process of manually labeling a training set with large numbers of fine-grained types is too expensive and error-prone (hard for annotators to distinguish over 100 types consistently). Current fine-grained typing systems annotate training corpora automatically using knowledge bases (*i.e.*, *distant supervision*) (Ling and Weld, 2012; Ren et al., 2016a). A typical workflow of distant supervision is as follows (see Fig. 1): (1) identify entity mentions in the documents; (2) link mentions to entities in KB; and (3) assign, to the candidate type set of each mention, all KB types of its KB-linked entity. However, existing distant supervision methods encounter the following limitations when doing automatic fine-grained typing.

• **Noisy Training Labels.** Current practice of distant supervision may introduce *label noise* to training data since it fails to take a mention's local contexts into account when assigning type labels (*e.g.*, see Fig. 1). Many previous studies ignore the label noises which appear in a majority of training mentions (see Table. 1, row (1)), and assume *all* types obtained by distant supervision are "correct" (Yogatama et al., 2015; Ling and Weld, 2012). The noisy labels may mislead the trained models and cause negative effect. A few systems try to denoise the training corpora using simple pruning heuristics such as deleting mentions with conflicting types (Gillick et al., 2014). However, such strategies significantly reduce the size of training set (Table 1, rows (2a-c)) and lead to performance degradation (later shown in our experiments). The larger the target type set, the more severe the loss.

• **Type Correlation.** Most existing methods (Yogatama et al., 2015; Ling and Weld, 2012) treat every type label in a training mention's candidate type set *equally* and *independently* when learning the classifiers but ignore the fact that types in the given hierarchy are semantically correlated (*e.g.*, `actor` is more relevant to `singer` than to `politician`). As a consequence, the learned classifiers may bias toward popular types but perform poorly on infrequent types since training data on infrequent types is scarce. Intuitively, one should pose smaller penalty on types which are semantically more relevant to the true types. For example, in Fig. 1 `singer` should receive a smaller penalty than `politician` does, by knowing that `actor` is a true type for "*Arnold Schwarzenegger*" in S2. This provides classifiers with additional information to distinguish between two types, especially those infrequent ones.

Dataset	Wiki	OntoNotes	BBN	NYT
# of target types	113	89	47	446
(1) noisy mentions (%)	27.99	25.94	22.32	51.81
(2a) sibling pruning (%)	23.92	16.09	22.32	39.26
(2b) min. pruning (%)	28.22	8.09	3.27	32.75
(2c) all pruning (%)	45.99	23.45	25.33	61.12

Table 1: A study of label noise. (1): %mentions with multiple *sibling types* (*e.g.*, `actor`, `singer`); (2a)-(2c): %mentions deleted by the three pruning heuristics (2014) (see Sec. 4), for three experiment datasets and New York Times annotation corpus (2014).

In this paper, we approach the problem of *automatic fine-grained entity typing* as follows: (1) Use different objectives to model training mentions with correct type labels and mentions with noisy labels, respectively. (2) Design a novel *partial-label loss* to model true types within the noisy candidate type set which requires only the "*best*" candidate type to be relevant to the training mention, and progressively estimate the best type by leveraging various text features extracted for the mention. (3) Derive type correlation based on two signals: (i) the given type hierarchy, and (ii) the shared entities between two types in KB, and incorporate the correlation so induced by enforcing *adaptive margins* between different types for mentions in the training set. To integrate these ideas, we develop a novel embedding-based framework called **AFET**. First, it uses distant supervision to obtain candidate types for each mention, and extract a variety of text features from the mentions themselves and their local contexts. Mentions are partitioned into a "clean" set and a "noisy" set based on the given type hierarchy. Second, we embed mentions and types jointly into a low-dimensional space, where, in that space, objects (*i.e.*, features and types) that are semantically close to each other also have similar representations. In the proposed objective, an adaptive margin-based rank loss is pro-

posed to model the set of clean mentions to capture type correlation, and a partial-label rank loss is formulated to model the "best" candidate type for each noisy mention. Finally, with the learned embeddings (*i.e.*, mapping matrices), one can predict the type-path for each mention in the test set in a top-down manner, using its text features. The major contributions of this paper are as follows:

1. We propose an automatic fine-grained entity typing framework, which reduces label noise introduced by distant supervision and incorporates type correlation in a principle way.
2. A novel optimization problem is formulated to jointly embed entity mentions and types to the same space. It models noisy type set with a partial-label rank loss and type correlation with adaptive-margin rank loss.
3. We develop an iterative algorithm for solving the joint optimization problem efficiently.
4. Experiments with three public datasets demonstrate that AFET achieves significant improvement over the state of the art.

2 Automatic Fine-Grained Entity Typing

Our task is to automatically uncover the type information for entity mentions (*i.e.*, token spans representing entities) in natural language sentences. The task takes a document collection $\mathcal{D}$ (automatically labeled using a KB Ψ in conjunction with a target type hierarchy $\mathcal{Y}$) as input and predicts a type-path in $\mathcal{Y}$ for each mention from the test set $\mathcal{D}_t$.

Type Hierarchy and Knowledge Base. Two key factors in distant supervision are the target type hierarchy and the KB. A *type hierarchy*, $\mathcal{Y}$, is a tree where nodes represent types of interests from Ψ. Previous studies manually create several clean type hierarchies using types from Freebase (Ling and Weld, 2012) or WordNet (Yosef et al., 2012). In this study, we adopt the existing hierarchies constructed using Freebase types[2]. To obtain types for entities $\mathcal{E}_\Psi$ in Ψ, we use the human-curated entity-type facts in Freebase, denoted as $\mathcal{F}_\Psi = \{(e, y)\} \subset \mathcal{E}_\Psi \times \mathcal{Y}$.

[2]We use the Freebase dump as of 2015-06-30.

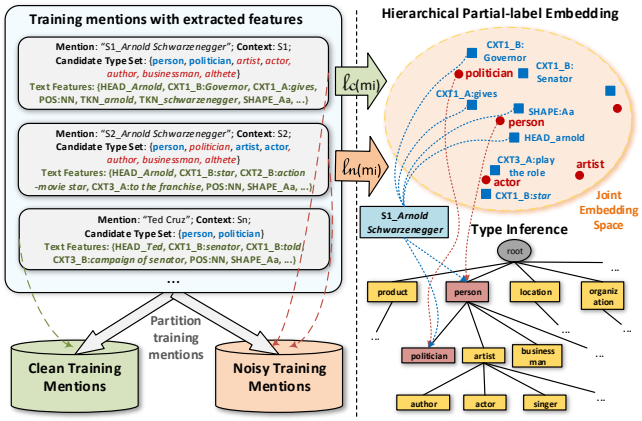


Figure 2: Framework Overview of AFET.

Automatically Labeled Training Corpora. There exist publicly available labeled corpora such as Wikilinks (Singh et al., 2012) and ClueWeb (Gabrilovich et al., 2013). In these corpora, entity mentions are identified and mapped to KB entities using anchor links. In specific domains (*e.g.*, product reviews) where such public corpora are unavailable, one can utilize distant supervision to automatically label the corpus (Ling and Weld, 2012). Specifically, an entity linker will detect mentions m_i and map them to one or more entity e_i in $\mathcal{E}_\Psi$. Types of e_i in KB are then associated with m_i to form its type set $\mathcal{Y}_i$, *i.e.*, $\mathcal{Y}_i = \{y \mid (e_i, y) \in \mathcal{F}_\Psi,\ y \in \mathcal{Y}\}$. Formally, a training corpus $\mathcal{D}$ consists of a set of extracted *entity mentions* $\mathcal{M} = \{m_i\}_{i=1}^N$, the *context* (*e.g.*, sentence, paragraph) of each mention $\{c_i\}_{i=1}^N$, and the candidate *type sets* $\{\mathcal{Y}_i\}_{i=1}^N$ for each mention. We represent $\mathcal{D}$ using a set of triples $\mathcal{D} = \{(m_i, c_i, \mathcal{Y}_i)\}_{i=1}^N$.

Problem Description. For each test mention, we aim to predict the correct type-path in $\mathcal{Y}$ based on the *mention's context*. More specifically, the test set $\mathcal{T}$ is defined as a set of mention-context pairs (m, c), where mentions in $\mathcal{T}$ (denoted as $\mathcal{M}_t$) are extracted from their sentences using existing extractors such as named entity recognizer (Finkel et al., 2005). We denote the gold type-path for a test mention m as $\mathcal{Y}^*$. This work focuses on learning a typing model from the noisy training corpus D, and estimating $\mathcal{Y}^*$ from $\mathcal{Y}$ for each test mention m (in set $\mathcal{M}_t$), based on mention m, its context c, and the learned model.

Framework Overview. At a high level, the AFET framework (see also Fig. 2) learns low-dimensional representations for entity types and text features, and

infers type-paths for test mentions using the learned embeddings. It consists of the following steps:

1. Extract text features for entity mentions in training set $\mathcal{M}$ and test set $\mathcal{M}_t$ using their surface names as well as the contexts. (Sec. 3.1).
2. Partition training mentions $\mathcal{M}$ into a clean set (denoted as $\mathcal{M}_c$) and a noisy set (denoted as $\mathcal{M}_n$) based on their candidate type sets (Sec. 3.2).
3. Perform joint embedding of entity mentions $\mathcal{M}$ and type hierarchy $\mathcal{Y}$ into the same low-dimensional space where, in that space, close objects also share similar types (Secs. 3.3-3.6).
4. For each test mention m, estimate its type-path $\mathcal{Y}^*$ (on the hierarchy $\mathcal{Y}$) in a top-down manner using the learned embeddings (Sec. 3.6).

3 The AFET Framework

This section introduces the proposed framework and formulates an optimization problem for learning embeddings of text features and entity types jointly.

3.1 Text Feature Generation

We start with a representation of entity mentions. To capture the shallow syntax and distributional semantics of a mention $m_i \in \mathcal{M}$, we extract various features from both m_i itself and its context c_i. Table 2 lists the set of text features used in this work, which is similar to those used in (Yogatama et al., 2015; Ling and Weld, 2012). We denote the set of M unique features extracted from $\mathcal{D}$ as $\mathcal{F} = \{f_j\}_{j=1}^{M}$.

3.2 Training Set Partition

A training mention m_i (in set $\mathcal{M}$) is considered as a "clean" mention if its candidate type set obtained by distant supervision (*i.e.*, $\mathcal{Y}_i$) is not ambiguous, *i.e.*, candidate types in $\mathcal{Y}_i$ can form a *single* path in tree $\mathcal{Y}$. Otherwise, a mention is considered as "noisy" mention if its candidate types form *multiple* type-paths in $\mathcal{Y}$. Following the above hypothesis, we judge each mention m_i (in set $\mathcal{M}$) and place it in either the "clean" set $\mathcal{M}_c$, or the "noisy" set $\mathcal{M}_n$. Finally, we have $\mathcal{M} = \mathcal{M}_c \cup \mathcal{M}_n$.

3.3 The Joint Mention-Type Model

We propose to learn mappings into low-dimensional vector space, where, both entity mentions and type

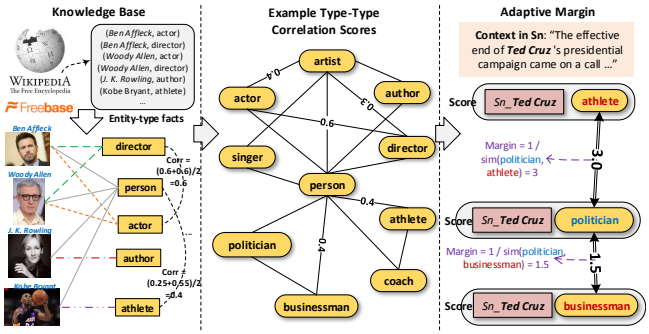


Figure 3: An illustration of KB-based type correlation computation, and the proposed adaptive margin.

labels (in the training set) are represented, and in that space, *two objects are embedded close to each other if and only if they share similar types*. In doing so, we later can derive the representation of a test mention based on its text features and the learned mappings. Mapping functions for entity mentions and entity type labels are different as they have different representations in the *raw* feature space, but are jointly learned by optimizing a global objective of interests to handle the aforementioned challenges.

Each entity mention $m_i \in M$ can be represented by a M-dimensional feature vector $\mathbf{m}_i \in \mathbb{R}^M$, where $m_{i,j}$ is the number of occurrences of feature f_j (in set $\mathcal{F}$) for m_i. Each type label $y_k \in \mathcal{Y}$ is represented by a K-dimensional binary indicator vector $\mathbf{y}_k \in \{0,1\}^K$, where $\mathbf{y}_{k,k} = 1$, and 0 otherwise.

Specifically, we aim to learn a mapping function from the mention's feature space to a low-dimensional vector space, *i.e.*, $\Phi_{\mathcal{M}}(\mathbf{m}_i) : \mathbb{R}^M \mapsto \mathbb{R}^d$ and a mapping function from type label space to the same low-dimensional space, *i.e.*, $\Phi_{\mathcal{Y}}(\mathbf{y}_k) : \mathbb{R}^K \mapsto \mathbb{R}^d$. In this work, we adopt linear maps, as similar to the mapping functions used in (Weston et al., 2011).

$$\Phi_{\mathcal{M}}(\mathbf{m}_i) = \mathbf{U}\mathbf{m}_i; \quad \Phi_{\mathcal{Y}}(\mathbf{y}_k) = \mathbf{V}\mathbf{y}_k, \qquad (1)$$

where $\mathbf{U} \in \mathbb{R}^{d \times M}$ and $\mathbf{V} \in \mathbb{R}^{d \times K}$ are the projection matrices for mentions and type labels, respectively.

3.4 Modeling Type Correlation

In type hierarchy (tree) $\mathcal{Y}$, types closer to each other (*i.e.*, shorter path) tend to be more related (*e.g.*, `actor` is more related to `artist` than to `person` in the right column of Fig. 2). In KB Ψ, types assigned to similar sets of entities should be more related to each other than those assigned to quite different entities (Jiang et al., 2015) (*e.g.*, `actor` is

Feature	Description	Example
Head	Syntactic head token of the mention	"HEAD_Turing"
Token	Tokens in the mention	"Turing", "Machine"
POS	Part-of-Speech tag of tokens in the mention	"NN"
Character	All character trigrams in the head of the mention	":tu", "tur", ..., "ng:"
Word Shape	Word shape of the tokens in the mention	"Aa" for "Turing"
Length	Number of tokens in the mention	"2"
Context	Unigrams/bigrams before and after the mention	"CXT_B:Maserati ,", "CXT_A:and the"
Brown Cluster	Brown cluster ID for the head token (learned using $\mathcal{D}$)	"4_1100", "8_1101111", "12_111011111111"
Dependency	Stanford syntactic dependency (Manning et al., 2014) associated with the head token	"GOV:nn", "GOV:turing"

Table 2: Text features used in this paper. "*Turing Machine*" is used as an example mention from "*The band's former drummer Jerry Fuchs—who was also a member of Maserati,* ***Turing Machine*** *and The Juan MacLean—died after falling down an elevator shaft.*".

more related to `director` than to `author` in the left column of Fig. 3). Thus, type correlation between y_k and $y_{k'}$ (denoted as $w_{kk'}$) can be measured either using the *one over the length of shortest path in* $\mathcal{Y}$, or using the *normalized number of shared entities in KB*, which is defined as follows.

$$w_{kk'} = \Big(|\mathcal{E}_k \cap \mathcal{E}_{k'}|/|\mathcal{E}_k| + |\mathcal{E}_k \cap \mathcal{E}_{k'}|/|\mathcal{E}_{k'}|\Big)/2. \quad (2)$$

Although a shortest path is efficient to compute, its accuracy is limited—It is not always true that a type (*e.g.*, `athlete`) is more related to its parent type (*i.e.*, `person`) than to its sibling types (*e.g.*, `coach`), or that all sibling types are equally related to each other (*e.g.*, `actor` is more related to `director` than to `author`). We later compare these two methods in our experiments.

With the type correlation computed, we propose to apply *adaptive* penalties on different negative type labels (for a training mention), instead of treating all of the labels *equally* as in most existing work (Weston et al., 2011). The hypothesis is intuitive: given the positive type labels for a mention, we force the negative type labels which are related to the positive type labels to receive smaller penalty. For example, in the right column of Fig. 3, negative label `businessman` receives a smaller penalty (*i.e.*, margin) than `athele` does, since `businessman` is more related to `politician`.

Hypothesis 1 (Adaptive Margin) *For a mention, if a negative type is correlated to a positive type, the margin between them should be smaller.*

We propose an adaptive-margin rank loss to model the set of "clean" mentions (*i.e.*, $\mathcal{M}_c$), based on the above hypothesis. The intuition is simple: for each mention, rank all the positive types ahead of negative types, where the ranking score is measured by similarity between mention and type. We denote

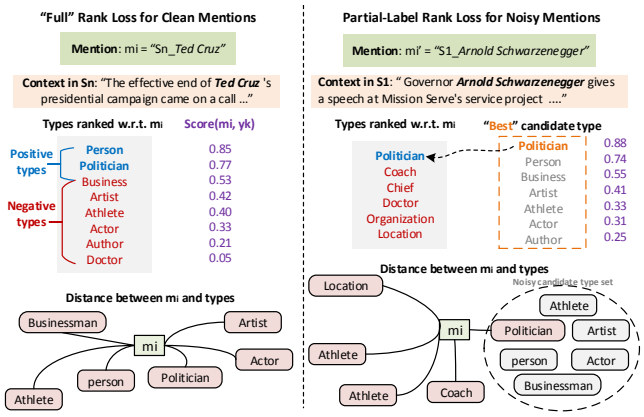


Figure 4: An illustration of the partial-label rank loss.

$f_k(m_i)$ as the similarity between (m_i, y_k) and is defined as the inner product of $\Phi_{\mathcal{M}}(\mathbf{m}_i)$ and $\Phi_{\mathcal{Y}}(\mathbf{y}_k)$.

$$\ell_c(m_i, \mathcal{Y}_i, \overline{\mathcal{Y}}_i) = \sum_{y_k \in \mathcal{Y}_i} \sum_{y_{\bar{k}} \in \overline{\mathcal{Y}}_i} L\Big\lfloor \mathrm{rank}_{y_k}\Big(f(m_i)\Big)\Big\rfloor \Theta_{i,k,\bar{k}};$$

$$\Theta_{i,k,\bar{k}} = \max\Big\{0, \gamma_{k,\bar{k}} - f_k(m_i) + f_{\bar{k}}(m_i)\Big\};$$

$$\mathrm{rank}_{y_k}\Big(f(m_i)\Big) = \sum_{y_{\bar{k}} \in \overline{\mathcal{Y}}_i} \mathbb{1}\Big(\gamma_{k,\bar{k}} + f_{\bar{k}}(m_i) > f_k(m_i)\Big).$$

Here, $\gamma_{k,\bar{k}}$ is the adaptive margin between positive type k and negative type $\bar{k}$, which is defined as $\gamma_{k,\bar{k}} = 1 + 1/(w_{k,\bar{k}} + \alpha)$ with a smooth parameter α. $L(x) = \sum_{i=1}^{x} \frac{1}{i}$ transforms rank to a weight, which is then multiplied to the max-margin loss $\Theta_{i,k,\bar{k}}$ to optimize precision at x (Weston et al., 2011).

3.5 Modeling Noisy Type Labels

True type labels for noisy entity mentions $\mathcal{M}_n$ (i.e., mentions with ambiguous candidate types in the given type hierarchy) in each sentence are not available in knowledge bases. To effectively model the set of noisy mentions, we propose not to treat *all*

candidate types (*i.e.*, $\{\mathcal{Y}_i\}$ as true labels. Instead, we model the "true" label among the candidate set as *latent value*, and try to infer that using text features.

Hypothesis 2 (Partial-Label Loss) *For a noisy mention, the maximum score associated with its candidate types should be greater than the scores associated with any other non-candidate types*

We extend the partial-label loss in (Nguyen and Caruana, 2008) (used to learn linear classifiers) to enforce Hypothesis 2, and integrate with the adaptive margin to define the loss for m_i (in set $\mathcal{M}_n$).

$$\ell_n(m_i, \mathcal{Y}_i, \overline{\mathcal{Y}}_i) = \sum_{\bar{k} \in \overline{\mathcal{Y}}_i} L\Big\lfloor \text{rank}_{y_{k^*}}\Big(f(m_i)\Big)\Big\rfloor \Omega_{i,\bar{k}};$$

$$\Omega_{i,k} = \max\Big\{0, \gamma_{k^*,\bar{k}} - f_{k^*}(m_i) + f_{\bar{k}}(m_i)\Big\};$$

$$\text{rank}_{y_{k^*}}\Big(f(m_i)\Big) = \sum_{y_{\bar{k}} \in \overline{\mathcal{Y}}_i} \mathbb{1}\Big(\gamma_{k^*,\bar{k}} + f_{\bar{k}}(m_i) > f_{k^*}(m_i)\Big)$$

where we define . $y_{k^*} = \text{argmax}_{y_k \in \mathcal{Y}_i} f_k(m_i)$ and $y_{\bar{k}^*} = \text{argmax}_{y_k \in \overline{\mathcal{Y}}_i} f_k(m_i)$.

Minimizing ℓ_n encourages a *large margin* between the maximum scores $\max_{y_k \in \mathcal{Y}_i} f_{y_k}(m_i)$ and $\max_{y_{\bar{k}} \in \overline{\mathcal{Y}}_i} f_{y_k}(m_i)$. This forces m_i to be embedded closer to the most "relevant" type in the noisy candidate type set, *i.e.*, $y^* = \text{argmax}_{y_k \in \mathcal{Y}_i} f_{y_k}(m_i)$, than to *any other* non-candidate types (*i.e.*, Hypothesis 2). This constrasts sharply with multi-label learning (Yosef et al., 2012), where a large margin is enforced between *all* candidate types and non-candidate types without considering noisy types.

3.6 Hierarchical Partial-Label Embedding

Our goal is to embed the heterogeneous graph G into a d-dimensional vector space, following the three proposed hypotheses in the section. Intuitively, one can *collectively* minimize the objectives of the two kinds of loss functions ℓ_c and ℓ_n, across all the training mentions. To achieve the goal, we formulate a joint optimization problem as follows.

$$\min_{\mathbf{U},\,\mathbf{V}} \mathcal{O} = \sum_{m_i \in \mathcal{M}_c} \ell_c(m_i, \mathcal{Y}_i, \overline{\mathcal{Y}}_i) + \sum_{m_i \in \mathcal{M}_n} \ell_n(m_i, \mathcal{Y}_i, \overline{\mathcal{Y}}_i).$$

We use an alternative minimization algorithm based on block-wise coordinate descent (Tseng, 2001) to *jointly* optimize the objective $\mathcal{O}$. One can also apply stochastic gradient descent to do online update.

Type Inference. With the learned mention embeddings $\{\mathbf{u}_i\}$ and type embeddings $\{\mathbf{v}_k\}$, we perform top-down search in the given type hierarchy $\mathcal{Y}$ to estimate the correct type-path $\mathcal{Y}_i^*$. Starting from the tree's root, we recursively find the best type among the children types by measuring the dot product of the corresponding mention and type embeddings, *i.e.*, $\text{sim}(\mathbf{u}_i, \mathbf{v}_k)$. The search process stops when we reach a leaf type, or the similarity score is below a pre-defined threshold $\eta > 0$.

Data sets	Wiki	OntoNotes	BBN
#Types	113	89	47
#Documents	780,549	13,109	2,311
#Sentences	1.51M	143,709	48,899
#Training mentions	2.69M	223,342	109,090
#Ground-truth mentions	563	9,604	121,001
#Features	644,860	215,642	125,637
#Edges in graph	87M	5.9M	2.9M

Table 3: Statistics of the datasets.

4 Experiments

4.1 Data Preparation

Datasets. Our experiments use three public datasets. (1) **Wiki** (Ling and Weld, 2012): consists of 1.5M sentences sampled from Wikipedia articles; (2) **OntoNotes** (Weischedel et al., 2011): consists of 13,109 news documents where 77 test documents are manually annotated (Gillick et al., 2014); (3) **BBN** (Weischedel and Brunstein, 2005): consists of 2,311 Wall Street Journal articles which are manually annotated using 93 types. Statistics of the datasets are shown in Table 3.

Training Data. We followed the process in (Ling and Weld, 2012) to generate training data for the Wiki dataset. For the BBN and OntoNotes datasets, we used DBpedia Spotlight[3] for entity linking. We discarded types which cannot be mapped to Freebase types in the BBN dataset (47 of 93).

Table 2 lists the set of features used in our experiments, which are similar to those used in (Yogatama et al., 2015; Ling and Weld, 2012) except for topics and ReVerb patterns. We discarded the features which occur only once in the corpus.

4.2 Evaluation Settings

For the Wiki and OntoNotes datasets, we used the provided test set. Since BBN corpus is fully annotated, we followed a 80/20 ratio to partition it into

[3] http://spotlight.dbpedia.org/

training/test sets. We report Accuracy (Strict-F1), Micro-averaged F1 (Mi-F1) and Macro-averaged F1 (Ma-F1) scores commonly used in the fine-grained type problem (Ling and Weld, 2012; Yogatama et al., 2015). Since we use the gold mention set for testing, the Accuracy (**Acc**) we reported is the same as the Strict F1.

Baselines. We compared the proposed method (AFET) and its variant with state-of-the-art typing methods, embedding methods and partial-label learning methods [4]: (1) **FIGER** (Ling and Weld, 2012); (2) **HYENA** (Yosef et al., 2012); (3) **FIGER/HYENA-Min** (Gillick et al., 2014): removes types appearing only once in the document; (4) **ClusType** (Ren et al., 2015): predicts types based on co-occurring relation phrases; (5) **HNM** (Dong et al., 2015): proposes a hybrid neural model without hand-crafted features; (6) **DeepWalk** (Perozzi et al., 2014): applies Deep Walk to a feature-mention-type graph by treating all nodes as the same type; (7) **LINE** (Tang et al., 2015b): uses a second-order LINE model on feature-type bipartite graph; (8) **PTE** (Tang et al., 2015a): applies the PTE joint training algorithm on feature-mention and type-mention bipartite graphs. (9) **WSABIE** (Yogatama et al., 2015): adopts WARP loss to learn embeddings of features and types; (10) **PL-SVM** (Nguyen and Caruana, 2008): uses a margin-based loss to handle label noise. (11) **CLPL** (Cour et al., 2011): uses a linear model to encourage large average scores for candidate types.

We compare AFET and its variant: (1) **AFET**: complete model with KB-induced type correlation; (2) **AFET-CoH**: with hierarchy-induced correlation (*i.e.*, shortest path distance); (3) **AFET-NoCo**: without type correlation (*i.e.*, all margin are "1") in the objective $\mathcal{O}$; and (4) **AFET-NoPa**: without label partial loss in the objective $\mathcal{O}$.

4.3 Performance Comparison and Analyses

Table 4 shows the results of AFET and its variants.

Comparison with the other typing methods. AFET outperforms both FIGER and HYENA systems, demonstrating the predictive power of the learned embeddings, and the effectiveness of modeling type correlation information and noisy candidate types. We also observe that pruning methods do not always improve the performance, since they aggressively filter out rare types in the corpus, which may lead to low Recall. ClusType is not as good as FIGER and HYENA because it is intended for coarse types and only utilizes relation phrases.

Comparison with the other embedding methods. AFET performs better than all other embedding methods. HNM does not use any linguistic features. None of the other embedding methods consider the label noise issue and treat the candidate type sets as clean. Although AFET adopts the WARP loss in WSABIE, it uses an adaptive margin in the objective to capture the type correlation information.

Comparison with partial-label learning methods. Compared with PL-SVM and CLPL, AFET obtains superior performance. PL-SVM assumes that only *one* candidate type is correct and does not consider type correlation. CLPL simply averages the model output for all candidate types, and thus may generate results biased to frequent false types. Superior performance of AFET mainly comes from modeling type correlation derived from KB.

Comparison with its variants. AFET always outperforms its variant on all three datasets. It gains performance from capturing type correlation, as well as handling type noise in the embedding process.

4.4 Case Analyses

Example output on news articles. Table 5 shows the types predicted by AFET, FIGER, PTE and WSABIE on two news sentences from OntoNotes dataset: AFET predicts fine-grained types with better accuracy (*e.g.*, `person_title`) and avoids overly-specific predictions (*e.g.*, `news_company`). Figure 5 shows the types estimated by AFET, PTE and WSABIE on a training sentence from OntoNotes dataset. We found AFET could discover the best type from noisy candidate types.

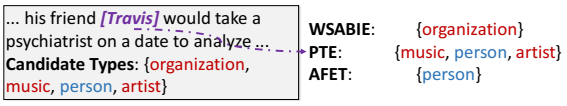


Figure 5: Example output of AFET and the compared methods on a training sentence from **OntoNotes** dataset.

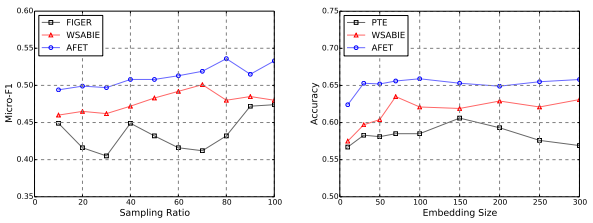
[4] We used the published code for FIGER, ClusType, HNM, LINE, PTE, and DeepWalk, and implemented other baselines which have no public code. Our implementations yield comparable performance as those reported in the original papers.

Typing Method	Wiki			OntoNotes			BBN		
	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1
CLPL (Cour et al., 2011)	0.162	0.431	0.411	0.201	0.347	0.358	0.438	0.603	0.536
PL-SVM (Nguyen and Caruana, 2008)	0.428	0.613	0.571	0.225	0.455	0.437	0.465	0.648	0.582
FIGER (Ling and Weld, 2012)	0.474	0.692	0.655	0.369	0.578	0.516	0.467	0.672	0.612
FIGER-Min (Gillick et al., 2014)	0.453	0.691	0.631	0.373	0.570	0.509	0.444	0.671	0.613
HYENA (Yosef et al., 2012)	0.288	0.528	0.506	0.249	0.497	0.446	0.523	0.576	0.587
HYENA-Min	0.325	0.566	0.536	0.295	0.523	0.470	0.524	0.582	0.595
ClusType (Ren et al., 2015)	0.274	0.429	0.448	0.305	0.468	0.404	0.441	0.498	0.573
HNM (Dong et al., 2015)	0.237	0.409	0.417	0.122	0.288	0.272	0.551	0.591	0.606
DeepWalk (Perozzi et al., 2014)	0.414	0.563	0.511	0.479	0.669	0.611	0.586	0.638	0.628
LINE (Tang et al., 2015b)	0.181	0.480	0.499	0.436	0.634	0.578	0.576	0.687	0.690
PTE (Tang et al., 2015a)	0.405	0.575	0.526	0.436	0.630	0.572	0.604	0.684	0.695
WSABIE (Yogatama et al., 2015)	0.480	0.679	0.657	0.404	0.580	0.527	0.619	0.670	0.680
AFET-NoCo	0.526	0.693	0.654	0.486	0.652	0.594	0.655	0.711	0.716
AFET-NoPa	0.513	0.675	0.642	0.463	0.637	0.591	0.669	0.715	0.724
AFET-CoH	0.433	0.583	0.551	0.521	0.680	0.609	0.657	0.703	0.712
AFET	**0.533**	**0.693**	**0.664**	**0.551**	**0.711**	**0.647**	**0.670**	**0.727**	**0.735**

Table 4: Study of typing performance on the three datasets.

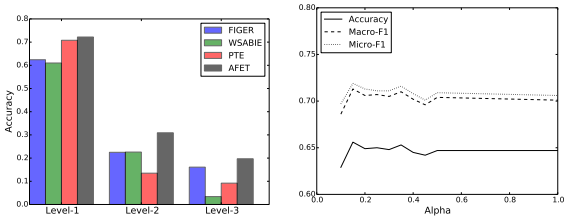
Text	"... going to be an imminent easing of monetary policy, " said Robert Dederick , chief economist at ***Northern Trust Co. in Chicago***.	...It's terrific for advertisers to know the reader will be paying more , " said Michael Drexler , ***national media director*** at Bozell Inc. ad agency.
Ground Truth	`organization`, `company`	`person`, `person_title`
FIGER	`organization`	`organization`
WSABIE	`organization`, `company`, `broadcast`	`organization`, `company`, `news_company`
PTE	`organization`	`person`
AFET	`organization`, `company`	`person`, `person_title`

Table 5: Example output of AFET and the compared methods on two news sentences from **OntoNotes** dataset.



(a) Varying training set size (b) Varying d

Figure 6: Performance change with respect to (a) sampling ratio of training mentions on the Wiki dataset; and (b) embedding dimension d on the BBN dataset.



(a) Test at different levels (b) Varying α

Figure 7: Performance change (a) at different levels of the type hierarchy on the OntoNotes dataset; and (b) with respect to smooth parameter α on the BBN dataset.

Testing the effect of training set size and dimension. Experimenting with the same settings for model learning, Fig. 6(a) shows the performance trend on the Wiki dataset when varying the sampling ratio (subset of mentions randomly sampled from the training set $\mathcal{D}$). Fig. 6(b) analyzes the performance sensitivity of AFET with respect to d—the embedding dimension on the BBN dataset. Accuracy of AFET improves as d becomes large but the gain decreases when d is large enough.

Testing sensitivity of the tuning parameter. Fig. 7(b) analyzes the sensitivity of AFET with respect to α on the BBN dataset. Performance increases as α becomes large. When α is large than 0.5, the performance becomes stable.

Testing at different type levels. Fig. 7(a) reports the Ma-F1 of AFET, FIGER, PTE and WSABIE at different levels of the target type hierarchy (*e.g.*, person and location on level-1, politician and artist on level-2, author and actor on level-3). The results show that it is more difficult to distinguish among more fine-grained types. AFET always outperforms the other two method, and achieves a 22.36% improvement in Ma-F1, compared to FIGER on level-3 types. The gain mainly comes from explicitly modeling the noisy candidate types.

Testing for frequent/infrequent types. We also

Type	animal	city
# of Training Mentions	1882	12421
# of Test Mentions	8	240
WSABIE	0.176	0.546
FIGER	0.167	0.648
PTE	0.222	0.677
AFET	**0.400**	**0.766**

Table 6: Example output of AFET and other methods on frequent/infrequent type from **OntoNotes** dataset.

evaluate the performance on frequent and rare types (Table 6). Note that we use a different evaluation metric, which is introduced in (Yosef et al., 2012) to calculate the F1 score for a type. We find **AFET** can always perform better than other baselines and it works for both frequent and rare types.

5 Related Work

There has been considerable work on named entity recognition (NER) (Manning et al., 2014), which focuses on three types (*e.g.*, `person`, `location`) and cast the problem as multi-class classification following the type mutual exclusion assumption (*i.e.*, one type per mention) (Nadeau and Sekine, 2007).

Recent work has focused on a much larger set of fine-grained types (Yosef et al., 2012; Ling and Weld, 2012). As the type mutual exclusion assumption no longer holds, they cast the problem as multi-label multi-class (hierarchical) classification problems (Gillick et al., 2014; Yosef et al., 2012; Ling and Weld, 2012). Embedding techniques are also recently applied to jointly learn feature and type representations (Yogatama et al., 2015; Dong et al., 2015). Del Corro *et al.* (2015) proposed an unsupervised method to generate context-aware candidates types, and subsequently select the most appropriate type. Gillick *et al.* (2014) discuss the label noise issue in fine-grained typing and propose three pruning heuristics. However, these heuristics aggressively delete training examples and may suffer from low recall (see Table. 4).

In the context of distant supervision, label noise issue has been studied for other information extraction tasks such as relation extraction (Takamatsu et al., 2012). In relation extraction, label noise is introduced by the false positive textual matches of entity pairs. In entity typing, however, label noise comes from the assignment of types to entity mentions without considering their contexts. The forms of distant supervision are different in these two problems. Recently, (Ren et al., 2016b) has tackled the problem of label noise in fine-grained entity typing, but focused on how to generate a clean training set instead of doing entity typing.

Partial label learning (PLL) (Zhang, 2014; Nguyen and Caruana, 2008; Cour et al., 2011) deals with the problem where each training example is associated with a set of candidate labels, where *only one is correct*. Unlike existing PLL methods, our method considers type hierarchy and correlation.

6 Conclusion and Future Work

In this paper, we study *automatic fine-grained entity typing* and propose a *hierarchical partial-label embedding* method, AFET, that models "clean" and "noisy" mentions separately and incorporates a given type hierarchy to induce loss functions. AFET builds on a joint optimization framework, learns embeddings for mentions and type-paths, and iteratively refines the model. Experiments on three public datasets show that AFET is effective, robust, and outperforms other comparing methods.

As future work, it would be interesting to study topical features as the context cues of the entity mentions, to leverage multi-sensing embedding to represent linguistic features with multiple senses, and to exploits other effective modeling methods to inject type hierarchy information. The proposed objective function is general and can be considered to incorporate various language features, to conduct integrated modeling of multiple sources, and to be extended to distantly-supervised relation extraction.

7 Acknowledgments

Research was sponsored in part by the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), DARPA DEFT No. FA8750-13-2-0041, National Science Foundation IIS-1017362, IIS-1320617, IIS-1354329, and IIS-1523198, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (`www.bd2k.nih.gov`). The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Timothee Cour, Ben Sapp, and Ben Taskar. 2011. Learning from partial labels. *JMLR*, 12:1501–1536.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. Finet: Context-aware fine-grained named entity typing. In *EMNLP*.

Xin Luna Dong, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *KDD*.

Li Dong, Furu Wei, Hong Sun, Ming Zhou, and Ke Xu. 2015. A hybrid neural model for type classification of entity mentions. In *IJCAI*.

Jesse Dunietz and Dan Gillick. 2014. A new entity salience task with millions of training examples. *EACL*.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*.

Evgeniy Gabrilovich, Michael Ringgaard, and Amarnag Subramanya. 2013. Facc1: Freebase annotation of clueweb corpora.

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

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *ACL*.

Jyun-Yu Jiang, Chin-Yew Lin, and Pu-Jen Cheng. 2015. Entity-driven type hierarchy construction for freebase. In *WWW*.

Thomas Lin, Oren Etzioni, et al. 2012. No noun phrase left behind: detecting and typing unlinkable entities. In *EMNLP*.

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

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. *ACL*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *ACL*.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26.

Nam Nguyen and Rich Caruana. 2008. Classification with partial labels. In *KDD*.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *ACL*.

Xiang Ren, Ahmed El-Kishky, Chi Wang, Fangbo Tao, Clare R Voss, Heng Ji, and Jiawei Han. 2015. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *KDD*.

Xiang Ren, Ahmed El-Kishky, Chi Wang, and Jiawei Han. 2016a. Automatic entity recognition and typing in massive text corpora. In *WWW*.

Xiang Ren, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, and Jiawei Han. 2016b. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD*.

Sameer Singh, Amarnag Subramanya, Fernando Pereira, and Andrew McCallum. 2012. Wikilinks: A large-scale cross-document coreference corpus labeled via links to wikipedia. *UM-CS-2012-015*.

Shingo Takamatsu, Issei Sato, and Hiroshi Nakagawa. 2012. Reducing wrong labels in distant supervision for relation extraction. In *ACL*.

Jian Tang, Meng Qu, and Qiaozhu Mei. 2015a. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015b. Line: Large-scale information network embedding. In *WWW*.

Paul Tseng. 2001. Convergence of a block coordinate descent method for nondifferentiable minimization. *JOTA*, 109(3):475–494.

Ralph Weischedel and Ada Brunstein. 2005. Bbn pronoun coreference and entity type corpus. *Linguistic Data Consortium*, 112.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. Ontonotes: A large training corpus for enhanced processing.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*.

Dani Yogatama, Dan Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *ACL*.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. Hyena: Hierarchical type classification for entity names. In *COLING*.

Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*.

Min-Ling Zhang. 2014. Disambiguation-free partial label learning. In *SDM*.