Entity Typing: A Critical Step for Mining Structures from Massive Unstructured Text

Xiang Ren[†] Wenqi He[†] Ahmed El-Kishky[†] Clare R. Voss[‡] Heng Ji[♯] Meng Qu[†] Jiawei Han[†]

[†] University of Illinois at Urbana-Champaign, Urbana, IL, USA
[‡] Computational & Information Sciences Directorate, Army Research Laboratory, Adelphi, MD, USA
[♯] Computer Science Department, Rensselaer Polytechnic Institute, USA
[†]{xren7, wenqihe3, elkishk2, mengqu2, hanj}@illinois.edu [‡]clare.r.voss.civ@mail.mil [♯]jih@rpi.edu

ABSTRACT

We have been studying learning and mining graphs or networks. However, where do most real networks come from? Although some networks come from well-structured and explicitly connected nodes and links, a majority of networks come from massive unstructured text data, and it takes human efforts to extract them and build them explicitly. Unfortunately, manual data curation and extraction of structures from unstructured data can be costly, unscalable, and error-prone. We have been investigating a data-driven approach to building structured networks from unstructured text data. First, quality phrases can be mined from massive text corpus, serving as basic semantic units, mostly being entities. Second, types can be inferred for such entities from such massive text data with distant supervision and relationships among entities can be uncovered by network embedding as well. Therefore, entity typing is a critical step for mining structures from unstructured text data.

In this study, we focus on how to conduct entity typing with a data-driven approach. We show that "rough" entity types can be identified from massive text data with a distant supervision approach via some domain-independent knowledge-bases. However, for refined typing, even the type labels in a knowledge bases can be noisy (i.e., incorrect for the entity mention's local context). We propose a general framework, called PLE, to jointly embed entity mentions, text features and entity types into the same low-dimensional space where, in that space, objects whose types are semantically close have similar representations. Then we estimate the type-path for each training example in a top-down manner using the learned embeddings. We formulate a global objective for learning the embeddings from text corpora and knowledge bases, which adopts a novel margin-based loss that is robust to noisy labels and faithfully models type correlation derived from knowledge bases. Our experiments on three public typing datasets demonstrate the effectiveness and robustness of PLE, with an average of 25% improvement in accuracy compared to next best method.

1. INTRODUCTION

Converting data into knowledge is a critical research issue in the era of big data. Previous studies on structure-rich data have already achieved great success in gaining insights and knowledge by exploring and analyzing structured information. Recently, a new architecture, *heterogeneous information network*, has become a promising way of representing and organizing structured data, where nodes can be different types of entities, links can be typed, directed and weighted relationships, and both nodes and links may carry text or numeric attributes (e.g., Wikipedia and other knowledge bases, social networks like Facebook, and hyperlink networks like the World Wide Web). A variety of insightful knowledge can be uncovered by mining such semantic-rich heterogeneous information networks [29, 24, 6].

Unfortunately, the majority of massive amount of data in the real world are unstructured or loosely structured text (e.g., from news to social media, business and scientific documents, and web pages). To unlock the value of these unstructured text data, it is of great importance to uncover structures of real-world entities, such as people, products, and organizations. Identifying token spans of entity mentions in text and labeling their types enables effective structured analysis of text corpus. The extracted entity information can serve as primitives to progressively turn unstructured text corpora into heterogeneous information networks.

As intractable quantities of unstructured text data are produced, it would be infeasible to hire human editors to manually label the entities mentioned in text. Recent studies focus on automating entity recognition and typing. Traditional named entity recognition systems [20] are usually designed for several major types (*e.g.*, person, organization, location) and general domains (*e.g.*, news), and so require additional steps for adaptation to a new domain and new types. Entity linking techniques [27] map from given entity mentions detected in text to entities in KBs like Freebase, where type information can be collected. But most of such information is manually curated, and thus the set of entities so obtained is of limited coverage and freshness (*e.g.*, over 50% entities mentioned in Web documents are unlinkable [14]). The rapid emergence of domain-specific text corpora (*e.g.*, business reviews) poses significant challenges to traditional entity recognition and entity linking techniques and calls for methods of recognizing entity mentions of target types with minimal or no human supervision, and with no requirement that entities can be found in a KB.

There are broadly two kinds of efforts towards that goal: weak supervision and distant supervision. Weak supervision relies on manually-specified seed entity names in applying

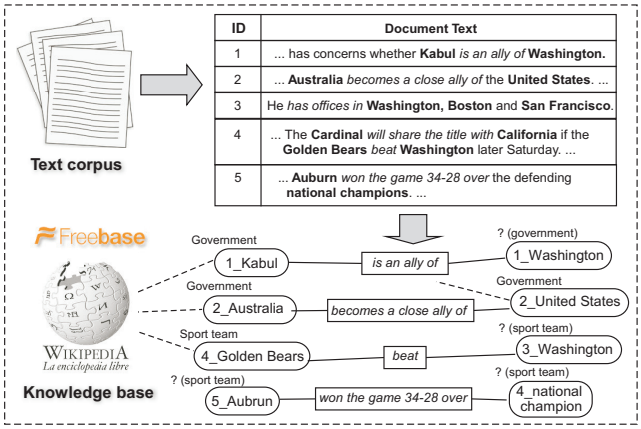

Figure 1: An example of distant supervision in entity typing.

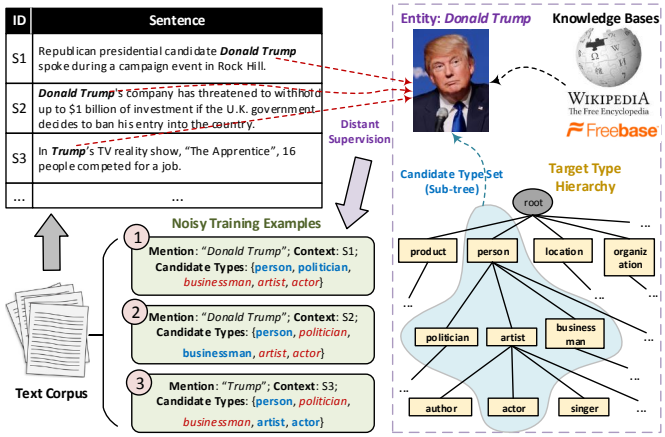

Figure 2: Current systems may find *Donald Trump* mentioned in sentences S1-S3 and assign the same types to all (listed within braces), when only some types are correct for context (blue).

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 type label noise. (1): %mentions with multiple *sibling types* (*e.g.*, `actor`, `singer`); (2a)-(2c): %mentions deleted by the three pruning heuristics [7], for three experiment datasets and New York Times annotation corpus [2].

pattern-based bootstrapping methods [8, 10] or label propagation methods [30] to identify more entities of each type. Both methods require careful seed entity selection by human [13] to ensure the seed quality. Distant supervision is a more recent trend, aiming to reduce expensive human labor by utilizing entity information in KBs [21, 14] (see Fig. 1). The typical workflow is: i) detect entity mentions from a corpus, ii) map candidate mentions to KB entities of target types, and iii) use those confidently mapped {mention, type} pairs as labeled data to infer the types of remaining candidate mentions.

This paper focuses on *distantly-supervised entity recognition in a domain-specific corpus*: Given a domain-specific corpus and a set of target entity types from a KB, we aim to effectively and efficiently detect entity mentions from that corpus, and categorize each by target types or Not-Of-Interest (NOI), with distant supervision. Existing distant supervision methods encounter the following limitations when handling a large, domain-specific corpus.

• **Domain Restriction:** They assume entity mentions are already extracted by entity detectors such as noun phrase chunkers. These tools are usually trained on general-domain corpora like news articles (clean, grammatical) and make use of various linguistic features, but do not work well on specific, dynamic or emerging domains (*e.g.*, tweets).

• **Context Sparsity:** Previous methods leverage a variety of contextual clues to find sources of shared semantics across different entities. However, there are often many ways to describe even the same relation between two entities (*e.g.*, "*beat*" and "*won the game 34-28 over*" in Fig. 1). This poses challenges on typing entity mentions when they are isolated from other entities or only share infrequent context.

• **Noisy Entity Label:** Many previous studies ignore the label noise in automatically labeled training corpora—*all* candidate types obtained by distant supervision are treated as "true" types in training classifiers [36, 15] (*e.g.*, see Fig. 2). This has become an impediment to improving the performance of current entity typing systems as a majority of training mentions have noisy types (see Table. 1, row (1)).

To address the first two challenges, we develop a novel solution called ClusType. First, it extracts entity mentions by a domain-agnostic phrase mining algorithm. The proposed algorithm has minimal dependence of linguistic assumption (*e.g.*, part-of-speech (POS) tagging requires fewer assumptions of the linguistic characteristics of a domain than semantic parsing), and demonstrates great cross-domain performance. Second, ClusType mines *relation phrases* co-occurring with entity mentions and infer synonymous relation phrases which express similar types of entities as arguments. This helps form connecting bridges among entities that do not share identical context but share synonymous relation phrases. To systematically integrate the above ideas, we construct a heterogeneous graph to faithfully represent entity mentions, entity surface names and relation phrases in a unified form (see Fig. 3). With the heterogeneous graph, we formulate a semi-supervised learning of two tasks jointly: (1) type propagation on graph, and (2) relation phrase clustering. By clustering synonymous relation phrases, we can propagate types among entities bridged via these synonymous relation phrases. Conversely, derived entity argument types serve as good features for clustering relation phrases. These two tasks mutually enhance each other and lead to quality recognition of unlinkable entity mentions.

To eliminate noisy entity labels, a few systems try to denoise automatically labeled training corpora by simple pruning heuristics such as deleting mentions with conflicting types [7]. However, such strategies significantly reduce the size of training set (Table 1, rows (2a-c)) and lead to performance degradation [26]. The larger the target type set, the more severe the loss. This motivated us to define a *new* task: *Label Noise Reduction in Entity Typing* (LNR), that is, identifying the correct type labels for *each training example* from its noisy candidate type set (generated by distant supervision with a given type hierarchy). While the typical entity typing task assumes that type labels in training data are all valid and focus on designing models to predict types for *unlabeled mentions*, LNR focuses on identifying the correct types for *automatically labeled mentions*, which is related to partial label learning [22, 1].

To approach LNR, a principled framework, Heterogeneous Partial-Label Embedding (PLE), is proposed (see Fig. 5). PLE models the true types labels in a candidate type set as latent variables and require only the "*best*" type (measured under

the proposed metric) to be relevant to the mention. It extracts a variety of text features from entity mentions and their local contexts, and leverage corpus-level co-occurrences between mentions and features to model mentions' types. Moreover, it models type correlation (semantic similarity) jointly with mention-candidate type associations and mention-feature co-occurrences, to assist type-path inference, by exploiting the shared entities between two types in a KB. A heterogeneous graph is constructed to represent three kinds of objects: entity mentions, text features and entity types, and their relationships. Associations between mentions and their *true* types are kept as latent structures in the graph to be estimated. We formulate a global objective to jointly embed the graph into a low-dimensional space where, in that space, objects whose types are semantically close also have similar representations. With the learned embeddings, we can efficiently estimate the correct type-path for each entity mention in the training set in a top-down manner.

The rest of the paper is organized as follows. Sec. 2 summarizes the related work. Sec. 3 introduces the ClusType framework, and Sec. 4 describes the label noise reduction and the proposed PLE framework. Finally, we discuss the connection between entity typing and information network construction in Sec. 5 and conclude the work in Sec. 6.

2. RELATED WORK

Entity Recognition and Typing. There have been extensive studies on entity recognition and typing. In terms of the dependence on context information, existing work can be categorized into context-dependent [20, 15] and context-independent approaches [21, 14]. Work along both lines can be further categorized in terms of the type granularity that is considered. Traditional named entity recognition systems [17] focus on coarse 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) [20]. Recent work has focused on a much larger set of fine-grained types [37, 15]. As type mutual exclusion assumption no longer holds, they cast the problem as multi-label multi-class (hierarchical) classification problems [7, 37, 15], or make use of various supervised embedding techniques [36] to jointly derive feature representations in classification tasks.

Most existing fine-grained typing systems use distant supervision to generate training examples and assume that all candidate types so generated are correct. By contrast, our framework instead seeks to remove false positives, denoising the data and leaving only the correct ones for each mention based on its local context. Output of our task, *i.e.*, denoised training data, helps train more effective classifiers for entity typing. Gillick *et al.* [7] discuss the label noise issue in fine-grained typing and propose three type pruning heuristics. However, these pruning methods aggressively filter training examples and may suffer from low recall.

Partial Label Learning. Partial label learning (PLL) [22, 1] deals with the problem where each training example is associated with a set of candidate labels, where *only one is correct*. One intuitive strategy to solve the problem is to assume equal contribution of each candidate label and average the outputs from all candidate labels for prediction [1]. Another strategy is to treat true label as latent variable and optimize objectives such as maximum likelihood criterion and maximum margin criterion [22] by EM procedure.

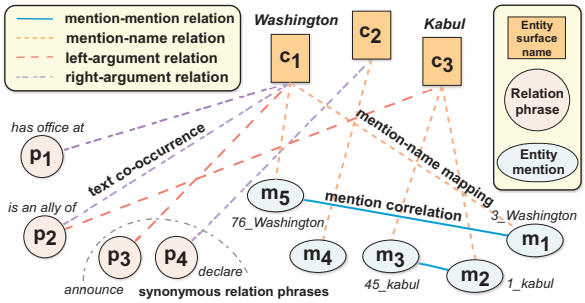

Figure 3: The constructed heterogeneous graph in ClusType.

3. THE CLUSTYPE FRAMEWORK

The input to ClusType framework is a document collection $\mathcal{D}$, a knowledge base Ψ with type schema $\mathcal{Y}_\Psi$, and a *target type set* $\mathcal{Y} \subset \mathcal{Y}_\Psi$. In this work, we use the type schema of Freebase and assume $\mathcal{Y}$ is covered by Freebase.

Entity Mention and Surface Name. An *entity mention*, m, is a token span in the text document which refers to a real-world entity e. Let c_m denote the *surface name* of m. People may use multiple surface names to refer to the same entity (*e.g.*, "*black mamba*" and "*KB*" for *Kobe Bryant*). Conversely, a surface name c could refer to different entities (*e.g.*, "*Washington*" in Fig. 1). We use a type indicator vector $\mathbf{y}_m \in \{0,1\}^T$ to denote the entity type for each mention m, where $T = |\mathcal{Y}| + 1$, *i.e.*, m has type $t \in \mathcal{Y}$ or is Not-of-Interest (NOI). By estimating $\mathbf{y}_m$, one can predict type of m as type $(m) = \text{argmax}_{1 \le i \le T}\ y_{m,i}$.

Relation Phrase. A *relation phrase* is a phrase that denotes a unary or binary relation in a sentence [4] (*e.g.*, see Fig. 4). We leverage the rich semantics embedded in relation phrases to provide type cues for their entity arguments.

Problem Description. Let $\mathcal{M} = \{m_1, ..., m_M\}$ denote the set of M candidate entity mentions extracted from $\mathcal{D}$. Suppose a subset of entity mentions $\mathcal{M}_L \subset \mathcal{M}$ can be confidently mapped to entities in Ψ. The type of a linked candidate $m \in \mathcal{M}_L$ can be obtained based on its mapping entity. This work focuses on predicting the types of *unlinkable candidate mentions* $\mathcal{M}_U = \mathcal{M} \setminus \mathcal{M}_L$.

Framework Overview. Our overall framework is as follows:

1. Perform phrase mining on a POS-tagged corpus to extract candidate entity mentions and relation phrases, and construct a heterogeneous graph G to represent available information in a unified form, which encodes our insights on modeling the type for entity mention.
2. Collect seed entity mentions $\mathcal{M}_L$ as labels by linking extracted candidate mentions $\mathcal{M}$ to the KB Ψ.
3. Estimate type indicator $\mathbf{y}$ for unlinkable candidate mention $m \in \mathcal{M}_U$ with the proposed type propagation integrated with relation phrase clustering on G.

3.1 Candidate Generation

Joint Extraction of Entity Mention and Relation Phrase. We introduce a data-driven phrase mining method to extract quality entity mentions and relation phrases. It adopts a *global significance score* to guide the filtering of low-quality phrases and relies on a set of generic POS patterns to remove phrases with improper syntactic structure [4]. By extending the methodology used in [3], the proposed phrase mining algorithm partitions sentences in the corpus into non-overlapping segments which meet a significance threshold and satisfy our syntactic constraints (see Table 2).

Over:RP the weekend the system:EP dropped:RP nearly inches of snow in:RP western Oklahoma:EP and at:RP [Dallas Fort Worth International Airport]:EP sleet and ice caused:RP hundreds of [flight cancellations]:EP and delays. ..... It is forecast:RP to reach:RP [northern Georgia]:EP by:RP [Tuesday afternoon]:EP, Washington:EP and [New York]:EP by:RP [Wednesday afternoon]:EP, meteorologists:EP said:RP.

`EP: entity mention candidate; RP: relation phrase.`

Figure 4: Example output of candidate generation.

Table 2: POS tag patterns for relation phrases.

Pattern	Example
V	disperse; hit; struck; knock;
P	in; at; of; from; to;
V P	locate in; come from; talk to;
VW*(P)	caused major damage on; come lately

`V-verb; P-prep; W-{adv | adj | noun | det | pron}`
`W* denotes multiple W; (P) denotes optional.`

Fig. 4 provides an example output of the candidate generation on New York Times (NYT) corpus. We further compare our method with a popular noun phrase chunker[1] in terms of entity detection performance, using the extracted entity mentions. Table 3 summarizes the comparison results on three datasets from different domains.

Table 3: Performance comparison on entity mention extraction.

Method	NYT		Yelp		Tweet	
	Prec	Recall	Prec	Recall	Prec	Recall
Our method	**0.469**	**0.956**	**0.306**	**0.849**	0.226	**0.751**
NP chunker	0.220	0.609	0.296	0.247	**0.287**	0.181

Heterogeneous Graph Construction. The basic idea for constructing the graph is that: the more two objects are likely to share the same label (*i.e.*, $t \in \mathcal{Y}$ or NOI), the larger the weight will be associated with their connecting edge. The constructed graph G unifies three types of links: *mention-name link*, *entity name-relation phrase link*, and *mention-mention link*, leading to three subgraphs $G_{\mathcal{M},\mathcal{C}}$, $G_{\mathcal{C},\mathcal{P}}$ and $G_{\mathcal{M}}$ (see Fig. 3 for example, and [25] for details).

Directly modeling the type indicator for each candidate mention may be infeasible due to the large number of candidate mentions (*e.g.*, $|\mathcal{M}| > 1$ million in our experiments). Intuitively, both the entity name and the surrounding relation phrases provide strong cues on the type of a candidate entity mention. We propose to model the type indicator of a candidate mention based on the type indicator of its surface name and the type signatures of its associated relation phrases. This enables our method to scale up.

By exploiting the aggregated co-occurrences between entity surface names and their surrounding relation phrases across multiple documents *collectively*, we weight the importance of different relation phrases for an entity name, and use their connected edge as bridges to propagate type information between different surface names by way of relation phrases. For each mention candidate, we assign it as the *left (right, resp.) argument* to the closest relation phrase appearing on its right (left, resp.) in a sentence. The *type signature* of a relation phrase refers to the two type indicators for its left and right arguments, respectively. If surface name c often appears as the left (right) argument of relation phrase p, then c's type indicator tends to be similar to the corresponding type indicator in p's type signature.

Finally, an entity mention candidate may have an ambiguous name as well as associate with ambiguous relation phrases. We propose to propagate the type information between candidate mentions of *each* entity name based on the hypothesis that: If there exists a strong correlation (*i.e.*, within sentence, common neighbor mentions) between two candidate mentions that share the same name, then their type indicators tend to be similar.

[1]TextBlob: `http://textblob.readthedocs.org/en/dev/`

3.2 Clustering-Integrated Type Propagation

In our solution, we formulate a joint optimization problem to minimize both a graph-based semi-supervised learning error and a multi-view relation phrase clustering objective.

Seed Mention Generation. We utilize a state-of-the-art entity name disambiguation tool [18] to map each candidate mention to Freebase entities. Only the mention candidates which are mapped with high confidence scores are considered as valid output. The linked mentions which associate with multiple target types are discarded to avoid type ambiguity. This leads to a set of labeled (seed) mentions $\mathcal{M}_L$.

Relation Phrase Clustering. We propose a general relation phrase clustering method to incorporate different features for clustering, which can be integrated with the graph-based type propagation in a mutually enhancing framework. Our hypothesis is that: two relation phrases tend to have similar cluster memberships, if they have similar (1) strings; (2) context words; and (3) left and right argument type indicators. In particular, type signatures of relation phrases have proven very useful in clustering of relation phrases which have infrequent or ambiguous strings and contexts. Our solution uses the features for multi-view clustering of relation phrases based on joint non-negative matrix factorization [25].

The Joint Optimization Problem. Our goal is to infer the label (type $t \in \mathcal{Y}$ or NOI) for *each* unlinkable entity mention candidate $m \in \mathcal{M}_U$, *i.e.*, estimating $\mathbf{Y}$. We propose an optimization problem to unify two different tasks to achieve this gold: (i) type propagation over both the type indicators of entity names $\mathbf{C}$ and the type signatures of relation phrases on the heterogeneous graph G by way of graph-based semi-supervised learning, and (ii) multi-view relation phrase clustering. The seed mentions $\mathcal{M}_L$ are used as initial labels for the type propagation. Details of the joint optimization problem and the algorithm for solving it are introduced in [25].

3.3 Results

We test the proposed method on three real-world datasets[2]: (1) **NYT:** constructed by crawling 2013 news articles from New York Times; (2) **Yelp:** We collected 230,610 reviews from the 2014 *Yelp dataset challenge*; and (3) **Tweet:** We randomly selected 10,000 users in Twitter and crawled at most 100 tweets for each user in May 2011.

Compared Methods: We compared the proposed method (ClusType) with its variants which only model part of the proposed hypotheses. Several state-of-the-art entity recognition approaches were also implemented (or tested using their published codes): For ClusType, besides the proposed full-fledged model, **ClusType**, we compare (1) **ClusType-NoWm**: This variant does not consider mention correlation subgraph; (2) **ClusType-NoClus**: It performs only type propagation on the heterogeneous graph; and (3) **ClusType-TwoStep**: It first conducts multi-view clustering to assign each relation phrase to a single cluster, and then performs ClusType-NoClus between entity names, candidate entity mentions and relation phrase clusters.

1. Comparing ClusType with the other methods on entity recognition. Table 4 summarizes the comparison results on

[2]Code and datasets used in the ClusType paper [25] can be found at: `https://github.com/shanzhenren/ClusType`.

Table 4: Performance comparisons on three datasets in terms of Precision, Recall and F1 score.

Data sets	NYT			Yelp			Tweet		
Method	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Pattern [8]	0.4576	0.2247	0.3014	0.3790	0.1354	0.1996	0.2107	0.2368	0.2230
FIGER [15]	0.8668	0.8964	0.8814	0.5010	0.1237	0.1983	**0.7354**	0.1951	0.3084
SemTagger [11]	0.8667	0.2658	0.4069	0.3769	0.2440	0.2963	0.4225	0.1632	0.2355
APOLLO [28]	0.9257	0.6972	0.7954	0.3534	0.2366	0.2834	0.1471	0.2635	0.1883
NNPLB [14]	0.7487	0.5538	0.6367	0.4248	0.6397	0.5106	0.3327	0.1951	0.2459
ClusType-NoClus	0.9130	0.8685	0.8902	0.7629	0.7581	0.7605	0.3466	0.4920	0.4067
ClusType-NoWm	0.9244	0.9015	0.9128	0.7812	0.7634	0.7722	0.3539	**0.5434**	0.4286
ClusType-TwoStep	0.9257	0.9033	0.9143	0.8025	0.7629	0.7821	0.3748	0.5230	0.4367
ClusType	**0.9550**	**0.9243**	**0.9394**	**0.8333**	**0.7849**	**0.8084**	0.3956	0.5230	**0.4505**

Table 5: Example output of ClusType and the compared methods on the Yelp dataset.

ClusType	SemTagger	NNPLB
The best **BBQ:Food** I've tasted in **Phoenix:LOC** ! I had the **[pulled pork sandwich]:Food** with **coleslaw:Food** and **[baked beans]:Food** for lunch. ...	The best BBQ I've tasted in **Phoenix:LOC** ! I had the pulled **[pork sandwich]:LOC** with **coleslaw:Food** and **[baked beans]:LOC** for lunch. ...	The best **BBQ:Loc** I've tasted in **Phoenix:LOC** ! I had the pulled pork **sandwich:Food** with coleslaw and baked **beans:Food** for **lunch:Food**. ...
I only go to **ihop:LOC** for **pancakes:Food** because I don't really like anything else on the menu. Ordered **[chocolate chip pancakes]:Food** and a **[hot chocolate]:Food**.	I only go to ihop for pancakes because I don't really like anything else on the menu. Ordered **[chocolate chip pancakes]:LOC** and a **[hot chocolate]:LOC**.	I only go to ihop for pancakes because I don't really like anything else on the menu. Ordered chocolate chip pancakes and a hot chocolate.

the three datasets. Overall, ClusType and its three variants outperform others on all metrics on NYT and Yelp and achieve superior Recall and F1 scores on Tweet.

2. Comparing ClusType with its variants. Comparing with ClusType-NoClus and ClusType-TwoStep, ClusType gains performance from integrating relation phrase clustering with type propagation in a mutually enhancing way. It always outperforms ClusType-NoWm on Precision and F1 on all three datasets. The enhancement mainly comes from modeling the mention correlation links, which helps disambiguate entity mentions sharing the same surface names.

3. Comparing with trained NER. Table 6 compares ours with a traditional NER method, *Stanford NER*, trained using classic corpora like ACE corpus, on three major types—person, location and organization. ClusType and its variants outperform Stanford NER on the corpora which are dynamic (*e.g.*, NYT) or domain-specific (*e.g.*, Yelp).

Table 6: F1 score comparison with trained NER.

Method	NYT	Yelp	Tweet
Stanford NER [5]	0.6819	0.2403	0.4383
ClusType-NoClus	0.9031	0.4522	0.4167
ClusType	**0.9419**	**0.5943**	**0.4717**

5. Example output on two Yelp reviews. Table 5 shows the output of ClusType, SemTagger and NNPLB on two Yelp reviews: ClusType extracts more entity mention candidates (*e.g.*, "*BBQ*", "*ihop*") and predicts their types with better accuracy (*e.g.*, "*baked beans*", "*pulled pork sandwich*").

4. LABEL NOISE REDUCTION

Knowledge Base and Target Type Hierarchy. A KB with a set of entities $\mathcal{E}_\Psi$ contains human-curated facts on both entity-entity facts of various relationship types and entity-type facts. We denote *entity-type facts* in a KB Ψ (with type schema $\mathcal{Y}_\Psi$) as $\mathcal{T}_\Psi = \{(e, y)\} \subset \mathcal{E}_\Psi \times \mathcal{Y}_\Psi$. A *target type hierarchy* is a tree where nodes represent types of interests from $\mathcal{Y}_\Psi$ (or types which can be mapped to those in $\mathcal{Y}_\Psi$).

Automatically Labeled Training Corpora. Formally, a labeled corpus for entity typing 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* $\{\mathcal{Y}_i\}_{i=1}^N$ automatically generated for each mention. We represent the training corpus using a set of mention-based triples $\mathcal{D} = \{(m_i, c_i, \mathcal{Y}_i)\}_{i=1}^N$.

Problem Description. Since $\mathcal{Y}_i$ is annotated for entity e_i, it includes all possible types of e_i and thus may contain types that are *irrelevant* to m_i's specific context c_i. Ideally, the type labels for $m_i \in \mathcal{M}$ should form a *type-path* (not required to end at a leaf) in $\mathcal{Y}_i$ [36], which serves as a *context-dependent* type annotation for m_i. However, as discussed in [7] and shown in Fig. 2, $\mathcal{Y}_i$ may contain type-paths that are irrelevant to m_i in c_i. Even though in some cases $\mathcal{Y}_i$ is already a type-path, it may be overly specific for c_i and so insufficient to infer the whole type-path using c_i. We denote the true type-path for mention m_i as $\mathcal{Y}_i^*$. Label noise reduction task focuses on estimating $\mathcal{Y}_i^*$ from $\mathcal{Y}_i$ based on mention m_i as well as its context c_i.

Framework Overview. Our solution [26] casts the problem as a *weakly-supervised* learning task, which aims to derive the relatedness between mentions and their candidate types using both corpus-level statistics and KB facts. We propose a *graph-based partial-label embedding framework* (see also Fig. 5) as follows:

1. Generate text features for each entity mention $m_i \in \mathcal{M}$, and construct a heterogeneous graph using three kinds of objects in the corpus, namely entity mentions $\mathcal{M}$, target types $\mathcal{Y}$ and text features (denoted as $\mathcal{F}$), to encode aforementioned signals in a unified form.
2. Perform joint embedding of the constructed graph G into the same low-dimensional space where, in that space, close objects (*i.e.*, whose embedding vectors have high similarity score) tend to also share the same types.
3. For each mention m_i (in set $\mathcal{M}$), search its candidate type sub-tree $\mathcal{Y}_i$ in a top-down manner and estimate the true type-path $\mathcal{Y}_i^*$ from learned embeddings.

4.1 Construction of Graphs

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 (*e.g.*, head token) and its context c_i (*e.g.*, bigram). Details of feature generation are introduced in [26].

With entity mentions $\mathcal{M}$, text features $\mathcal{F}$ and target types $\mathcal{Y}$, we build a heterogeneous graph G to unify three kinds of links: *mention-type link* represents each mention's candidate type assignment; *mention-feature link* captures corpus-level co-occurrences between mentions and text features; and *type-type link* encodes the type correlation derived from

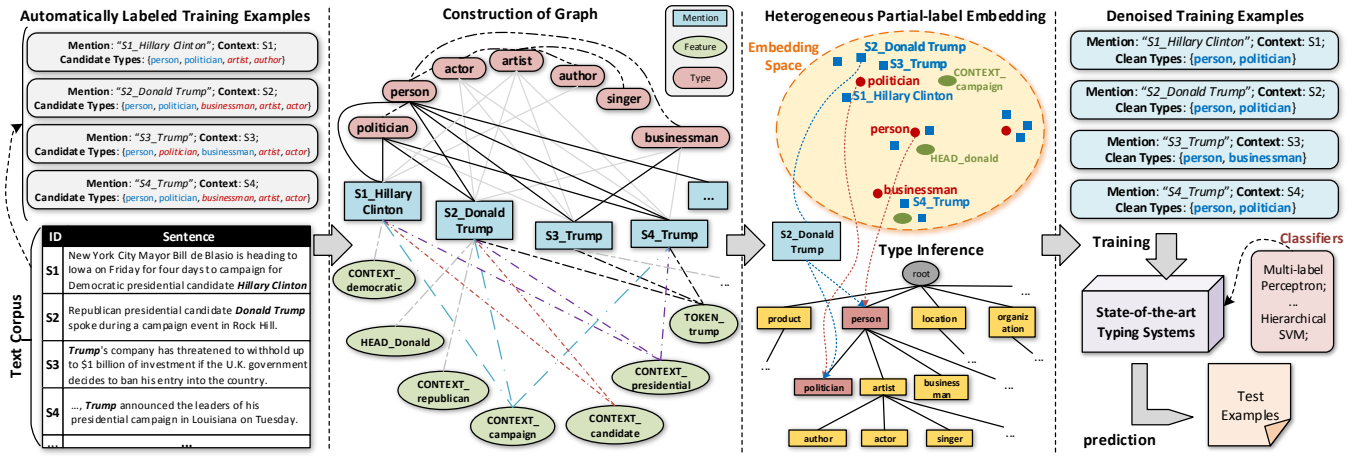

Figure 5: Framework Overview of Heterogeneous Partial-Label Embedding (PLE).

KB or target type hierarchy. This leads to three subgraphs G_{MY}, G_{MF}, and G_{YY}, respectively.

Mention-Type Association Subgraph. In the automatically labeled training corpus $\mathcal{D} = \{(m_i, c_i, \mathcal{Y}_i)\}$, each mention m_i is assigned a set of candidate types $\mathcal{Y}_i$ from the target type set $\mathcal{Y}$. This naturally forms a bipartite graph between entity mentions $\mathcal{M}$ and target types $\mathcal{Y}$, where each mention $m_i \in \mathcal{M}$ is linked to its candidate types $\mathcal{Y}_i$ with binary weight. However, some links are "false" links in the constructed mention-type subgraph—adopting the above assumptions may incorrectly yield mentions of different types having similar embeddings. We propose to model mention-type links based on the hypothesis that: A mention should be embedded closer to its most relevant candidate type than to any other non-candidate type, yielding higher similarity between the corresponding embedding vectors. During model learning, relevance between an entity mention and its candidate type is measured by the similarity between their current estimated embeddings. Text features, as complements to mention-candidate type links, also participate in modeling the mention embeddings, and help identify a mention's most relevant type.

Mention-Feature Co-occurrence Subgraph. Intuitively, entity mentions sharing many text features (*i.e.*, with similar distributions over $\mathcal{F}$) tend to have close type semantics; and text features which co-occur with many entity mentions in the corpus (*i.e.*, with similar distributions over $\mathcal{M}$) likely represent similar entity types. Therefore, if two entity mentions share similar features, they should be close to each other in the embedding space (*i.e.*, high similarity score). If two features co-occur with a similar set of mentions, their embedding vectors tend to be similar.

Type Correlation Subgraphs. In KB Ψ, types assigned to similar sets of entities should be more related to each other than those assigned to quite different entities [12] (*e.g.*, `actor` is more related to `director` than to `author` in the right column of Fig. 6). Thus, if high correlation exists between two target types based on either type hierarchy or KB, they should be embedded close to each other. We build a homogeneous graph G_{YY} to represent the correlation between types. Given two target types $y_k, y_{k'} \in \mathcal{Y}$, the correlation (*i.e.*, the edge weight in G_{YY}) between them is proportional to the number of entities they share in the KB.

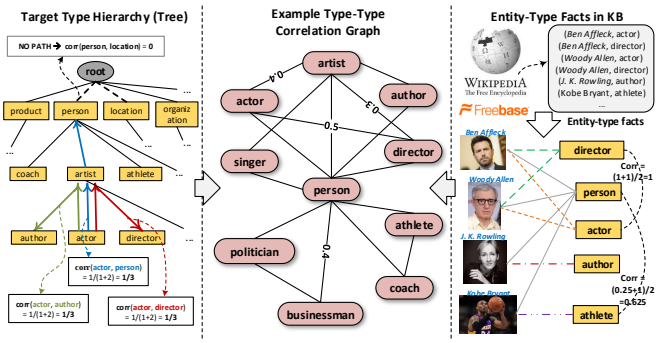

Figure 6: Example of constructing type correlation graph.

4.2 Heterogeneous Partial-Label Embedding

In our solution, we formulate a global objective [26], by extending a margin-based rank loss to model noisy mention-type links in G_{MY} and leveraging the distributional assumption [19] to model subgraphs G_{MF} and G_{YY}.

To effectively model the *noisy* mention-type links in subgraph G_{MY}, we extend the margin-based loss in [22] (used to learn linear classifiers) to enforce the hypothesis on mention-type association. The intuition of the loss is simple: for mention m_i, the maximum score associated with its candidate types $\mathcal{Y}_i$ is greater than the maximum score associated with any other non-candidate types $\overline{\mathcal{Y}}_i = \mathcal{Y} \setminus \mathcal{Y}_i$, where the scores are measured using current embedding vectors.

To model mention-feature co-occurrences represented by links in G_{MF}, we follow the idea that *nodes with similar distributions over neighbors are similar to each other*. This idea is similar to that found in Second-order Proximity model [32], and Skip-gram model [19]—it models text corpora following the distributional hypothesis [9] which says that you should know a word by the company it keeps.

Finally, type correlation links can be modeled with a method similar to that used in modeling the mention-feature subgraph—two types are similar to each other if they are correlated to the same set of types. As link (m_i, f_j) in bipartite graph G_{MF} is *directed*, we treat each *undirected* link $(y_k, y_{k'})$ in the homogeneous graph G_{YY} as two directed links [31].

The Global Optimization Objective. Our goal is to embed the heterogeneous graph G into a d-dimensional vector space, following the three proposed hypotheses in Sec. 4.1. Intuitively, one can *collectively* minimize the objectives of

Method	Wiki							OntoNotes						
	Acc	Ma-P	Ma-R	Ma-F1	Mi-P	Mi-R	Mi-F1	Acc	Ma-P	Ma-R	Ma-F1	Mi-P	Mi-R	Mi-F1
Raw	0.373	0.558	**0.681**	0.614	0.521	**0.719**	0.605	0.480	0.671	**0.793**	0.727	0.576	**0.786**	0.665
Sib [7]	0.373	0.583	0.636	0.608	0.578	0.653	0.613	0.487	0.710	0.732	0.721	0.675	0.702	0.688
Min [7]	0.373	0.561	0.679	0.615	0.524	0.717	0.606	0.481	0.680	0.777	0.725	0.592	0.763	0.667
All [7]	0.373	0.585	0.634	0.608	0.581	0.651	0.614	0.487	0.716	0.724	0.720	0.686	0.691	0.689
DeepWalk-Raw [23]	0.328	0.598	0.459	0.519	0.595	0.367	0.454	0.441	0.625	0.708	0.664	0.598	0.683	0.638
LINE-Raw [32]	0.349	0.600	0.596	0.598	0.590	0.610	0.600	0.549	0.699	0.770	0.733	0.677	0.754	0.714
WSABIE-Raw [36]	0.332	0.554	0.609	0.580	0.557	0.633	0.592	0.482	0.686	0.743	0.713	0.667	0.721	0.693
PTE-Raw [31]	0.419	0.678	0.597	0.635	0.686	0.607	0.644	0.529	0.687	0.754	0.719	0.657	0.733	0.693
PLE-NoCo	0.556	0.795	0.678	0.732	0.804	0.668	0.730	0.593	0.768	0.773	0.770	0.751	0.762	0.756
PLE-CoH	0.568	0.805	0.671	0.732	0.808	0.704	0.752	0.620	0.789	0.785	0.787	0.778	0.769	0.773
PLE	**0.589**	**0.840**	0.675	**0.749**	**0.833**	0.705	**0.763**	**0.639**	**0.814**	0.782	**0.798**	**0.791**	0.766	**0.778**

Table 7: Performance comparisons on LNR on **Wiki** and **OntoNotes** datasets.

the three subgraphs G_{MY}, G_{MF} and G_{YY}, as mentions $\mathcal{M}$ and types $\mathcal{Y}$ are shared across them. To achieve the goal, we formulate a joint optimization problem as follows.

$$\min_{\{\mathbf{u}_i\}_{i=1}^N, \{\mathbf{c}_j\}_{j=1}^M, \{\mathbf{v}_k, \mathbf{v}'_k\}_{k=1}^K} \mathcal{O} = \mathcal{O}_{MY} + \mathcal{O}_{MF} + \mathcal{O}_{YY},$$

where objective $\mathcal{O}_{MY}$ of the subgraph G_{MY} is specified by aggregating the partial-label loss defined in [26] across all the mentions $\mathcal{M}$, along with ℓ_2-regularizations on $\{u_i\}_{i=1}^N$ and $\{v_k\}_{k=1}^K$ to control the scale of the embeddings [22]. Objectives $\mathcal{O}_{MF}$ and $\mathcal{O}_{YY}$ are specified by the second-order proximity model introduced in [32]. To solve the proposed optimization problem, we develop an alternative minimization algorithm based on block-wise coordinate descent schema [33].

4.3 Results

Our experiments use three public datasets:[3]. **Wiki** [15], **OntoNotes** [35], and **BBN** [34].

Compared Methods. We compared the proposed method (PLE) with its variants which model parts of the hypotheses, and three pruning heuristics [7]. Several state-of-the-art embedding methods and partial-label learning methods were also implemented (or tested using their published codes). For PLE, besides the proposed model, **PLE**, which adopts KB-based type correlation subgraph, we compare (1) **PLE-NoCo**: This variant does not consider type correlation subgraph G_{YY} in the objective; and (2) **PLE-CoH**: It adopts type hierarchy-based correlation subgraph.

Example output on news articles. Table 8 shows the types estimated by PLE, PTE and WSABIE on three news sentences from OntoNotes dataset.

Performance on Label Noise Reduction. We first conduct *intrinsic evaluation* on how accurately PLE and the other methods can estimate the true types of mentions from its noisy candidate type set. Table 7 summarize the comparison results on the Wiki and OntoNotes datasets. For embeddings learned on different pruned corpora, we only show the combination that yields the best result.

Fine-Grained Entity Typing. In Table 9, we further conduct *extrinsic evaluation* on fine-grained typing to study the performance gain from denoising the automatically generated training corpus $\mathcal{D}$. Two state-of-the-art fine-grained type classifiers, HYENA [37] and FIGER [15], are trained on the denoised corpus which is generated using PLE or the other compared methods. Trained classifiers are then tested on the evaluation set. We also compare with partial-label learning methods PL-SVM [22] and CLPL [1].

5. FROM TYPING TO INFORMATION NETWORK CONSTRUCTION

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

Text	***NASA*** says it may decide by tomorrow whether another space walk will be needed ...	... the board of *directors* which are composed of twelve members directly appointed by the ***Queen***.
Wiki Page	`https://en.wikipedia.org/wiki/NASA`	`https://en.wikipedia.org/wiki/Elizabeth_II`
Cand. type set	`person`, `artist`, `location`, `structure`, `organization`, `company`, `news_company`	`person`, `artist`, `actor`, `author`, `person_title`, `politician`
WSABIE	`person`, `artist`	`person`, `artist`
PTE	`organization`, `company`, `news_company`	`person`, `artist`
PLE	`organization`, `company`	`person`, `person_title`

Table 8: Example output of PLE and the compared methods on two news sentences from the **OntoNotes** dataset.

Massive text data, in the form of news, social media, industry/business/government reports, or scientific literature, are ubiquitous and are valuable sources for knowledge mining. Recent studies have shown various kinds of interesting knowledge, including rank-based clustering, classification, similarity search, relationship prediction, and personalized recommendation, can be mined from typed, structured heterogeneous information networks [29]. To bridge unstructured text to typed, semantic structures and structured heterogeneous information networks, a critical step is to uncover appropriate types for phrases in text, with minimal human training efforts, and construct heterogeneous information networks automatically in a massive scale. Therefore, our roadmap for unstructured data to structured knowledge is laid out as follows.

1. **Phrase mining:** Recently, effective methods have been developed for mining quality phrases from large text corpora with no training effort or with only minor human labeling effort or distant supervision, such as ToPMine [3] and Segphrase [16]. By integrating with some part-of-speech tagging with the phrase mining process, the quality of phrases generated can be further enhanced.

2. **Entity recognition and typing:** Taking the entities or entity candidates generated from phrase mining, the work described here will generate general or refined types for entities in the text.

3. **Mining relationships among typed entities:** A rough correlation relationship among a set of entities can be inferred from the frequent co-occurrences in massive text corpora. More refined relationships among those entities can be uncovered by in-depth analysis of the corresponding language or sentiment features associated with those entities in massive text.

4. **Construction and refinement of typed heterogeneous networks:** Typed heterogeneous information networks can be constructed based on entities and their associated relationships discovered in the text.

Typing System	Noise Reduction Method	Wiki			OntoNotes			BBN		
		Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1	Acc	Ma-F1	Mi-F1
N/A	PL-SVM [22]	0.428	0.613	0.571	0.465	0.648	0.582	0.497	0.679	0.677
N/A	CLPL [1]	0.162	0.431	0.411	0.438	0.603	0.536	0.486	0.561	0.582
HYENA [37]	Raw	0.288	0.528	0.506	0.249	0.497	0.446	0.523	0.576	0.587
	Min [7]	0.325	0.566	0.536	0.295	0.523	0.470	0.524	0.582	0.595
	All [7]	0.417	0.591	0.545	0.305	0.552	0.495	0.495	0.563	0.568
	WSABIE-Min [36]	0.199	0.462	0.459	0.400	0.565	0.521	0.524	0.610	0.621
	PTE-Min [31]	0.238	0.542	0.522	0.452	0.626	0.572	0.545	0.639	0.650
	PLE-NoCo	0.517	0.672	0.634	0.496	0.658	0.603	0.650	0.709	0.703
	PLE	**0.543**	**0.695**	**0.681**	**0.546**	**0.692**	**0.625**	**0.692**	**0.731**	**0.732**
FIGER [15]	Raw	0.474	0.692	0.655	0.369	0.578	0.516	0.467	0.672	0.612
	Min	0.453	0.691	0.631	0.373	0.570	0.509	0.444	0.671	0.613
	All	0.453	0.648	0.582	0.400	0.618	0.548	0.461	0.636	0.583
	WSABIE-Min	0.455	0.646	0.601	0.425	0.603	0.546	0.481	0.671	0.618
	PTE-Min	0.476	0.670	0.635	0.494	0.675	0.618	0.513	0.674	0.657
	PLE-NoCo	0.543	0.726	0.705	0.547	0.699	0.639	0.643	0.753	0.721
	PLE	**0.599**	**0.763**	**0.749**	**0.572**	**0.715**	**0.661**	**0.685**	**0.777**	**0.750**

Table 9: Study of performance improvement on fine-grained typing systems **FIGER** [15] and **HYENA** [37] on the three datasets.

5. **Mining needed knowledge from structured heterogeneous networks:** Mining methods have been and will be further developed for effective mining of various kinds of knowledge in such networks, as shown in [29]. This will lead to the query-based flexible generation of different kinds of knowledge needed.

6. CONCLUSION

In this paper, we presented a data-driven approach to effective entity typing and demonstrated its power on mining different typed of massive data sets via our extensive experiments. We believe entity typing plays a critical role at turning unstructured text data into structured heterogeneous networks, which in turn can be mined systematically for user-interested knowledge. Thus, it may substantially enlarge the kinds of datasets to be examined in learning and mining graphs or networks.

Lots of interesting research frontiers are worth further exploration. Especially, mining fine-grained relationships among typed entities in massive text corpora is our current focus of study.

References

[1] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *JMLR*, 12:1501–1536, 2011.

[2] J. Dunietz and D. Gillick. A new entity salience task with millions of training examples. *EACL*, 2014.

[3] A. El-Kishky, Y. Song, C. Wang, C. R. Voss, and J. Han. Scalable topical phrase mining from text corpora. *VLDB*, 2015.

[4] A. Fader, S. Soderland, and O. Etzioni. Identifying relations for open information extraction. In *EMNLP*, 2011.

[5] J. R. Finkel, T. Grenager, and C. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL*, 2005.

[6] L. Getoor. *Introduction to statistical relational learning*. MIT press, 2007.

[7] D. Gillick, N. Lazic, K. Ganchev, J. Kirchner, and D. Huynh. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[8] S. Gupta and C. D. Manning. Improved pattern learning for bootstrapped entity extraction. In *CONLL*, 2014.

[9] Z. S. Harris. Distributional structure. *Word*, 10:146–162, 1954.

[10] Y. He and D. Xin. Seisa: set expansion by iterative similarity aggregation. In *WWW*, 2011.

[11] R. Huang and E. Riloff. Inducing domain-specific semantic class taggers from (almost) nothing. In *ACL*, 2010.

[12] J.-Y. Jiang, C.-Y. Lin, and P.-J. Cheng. Entity-driven type hierarchy construction for freebase. In *WWW*, 2015.

[13] Z. Kozareva and E. Hovy. Not all seeds are equal: Measuring the quality of text mining seeds. In *NAACL*, 2010.

[14] T. Lin, O. Etzioni, et al. No noun phrase left behind: detecting and typing unlinkable entities. In *EMNLP*, 2012.

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

[16] J. Liu, J. Shang, C. Wang, X. Ren, and J. Han. Mining quality phrases from massive text corpora. In *Proc. 2015 ACM SIGMOD Int. Conf. on Management of Data (SIGMOD'15)*, Melbourne, Australia, May 2015.

[17] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, and D. McClosky. The stanford corenlp natural language processing toolkit. *ACL*, 2014.

[18] P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. Dbpedia spotlight: shedding light on the web of documents. In *I-Semantics*, 2011.

[19] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.

[20] D. Nadeau and S. Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30:3–26, 2007.

[21] N. Nakashole, T. Tylenda, and G. Weikum. Fine-grained semantic typing of emerging entities. In *ACL*, 2013.

[22] N. Nguyen and R. Caruana. Classification with partial labels. In *KDD*, 2008.

[23] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk: Online learning of social representations. In *KDD*, 2014.

[24] S. Y. Philip, J. Han, and C. Faloutsos. *Link Mining: Models, Algorithms, and Applications*. Springer, 2010.

[25] X. Ren, A. El-Kishky, C. Wang, F. Tao, C. R. Voss, and J. Han. Clustype: Effective entity recognition and typing by relation phrase-based clustering. In *KDD*, 2015.

[26] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *KDD*, 2016.

[27] W. Shen, J. Wang, and J. Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *TKDE*, (99):1–20, 2014.

[28] W. Shen, J. Wang, P. Luo, and M. Wang. A graph-based approach for ontology population with named entities. In *CIKM*, 2012.

[29] Y. Sun and J. Han. Mining heterogeneous information networks: a structural analysis approach. *SIGKDD Explorations*, 14(2):20–28, 2013.

[30] P. P. Talukdar and F. Pereira. Experiments in graph-based semi-supervised learning methods for class-instance acquisition. In *ACL*, 2010.

[31] J. Tang, M. Qu, and Q. Mei. Pte: Predictive text embedding through large-scale heterogeneous text networks. In *KDD*, 2015.

[32] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *WWW*, 2015.

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

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

[35] R. Weischedel, E. Hovy, M. Marcus, M. Palmer, R. Belvin, S. Pradhan, L. Ramshaw, and N. Xue. Ontonotes: A large training corpus for enhanced processing. 2011.

[36] D. Yogatama, D. Gillick, and N. Lazic. Embedding methods for fine grained entity type classification. In *ACL*, 2015.

[37] M. A. Yosef, S. Bauer, J. Hoffart, M. Spaniol, and G. Weikum. Hyena: Hierarchical type classification for entity names. In *COLING*, 2012.