Label Noise Reduction in Entity Typing by Heterogeneous Partial-Label Embedding

Xiang Ren* (UIUC), Wenqi He* (UIUC), Meng Qu (UIUC), Clare R. Voss (ARL), Heng Ji (RPI), Jiawei Han (UIUC)
 Entity Typing

- **Fine-grained Entity Typing**: Type labels for a mention forms a “type-path” (not necessarily ending in a leaf node) in a given (tree-structured) type hierarchy

<table>
<thead>
<tr>
<th>ID</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Republican presidential candidate <em>Donald Trump</em> spoke during a campaign event in Rock Hill.</td>
</tr>
<tr>
<td>S2</td>
<td><em>Donald Trump’s</em> company has threatened to withhold up to $1 billion of investment if the U.K. government decides to ban his entry into the country.</td>
</tr>
<tr>
<td>S3</td>
<td>In Trump’s TV reality show, “The Apprentice”, 16 people competed for a job.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Type-path</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Person ➔ politician</td>
<td></td>
</tr>
<tr>
<td>Person ➔ businessman</td>
<td></td>
</tr>
<tr>
<td>Person ➔ artist ➔ actor</td>
<td></td>
</tr>
</tbody>
</table>

- Manually annotating training corpora with 100+ entity types
- Expensive & Error-prone
- **Current practice**: use distant supervision to *automatically labeled training corpora*
**Label Noise in Entity Typing**

<table>
<thead>
<tr>
<th>ID</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>Republican presidential candidate Donald Trump spoke during a campaign event in Rock Hill.</td>
</tr>
<tr>
<td>S2</td>
<td>Donald Trump’s company has threatened to withhold up to $1 billion of investment if the U.K. government decides to ban his entry into the country.</td>
</tr>
<tr>
<td>S3</td>
<td>In Trump’s TV reality show, “The Apprentice”, 16 people competed for a job.</td>
</tr>
</tbody>
</table>

---

**Donald Trump is mentioned in sentences S1-S3.**

- **Distant supervision**
- Assign *same* types (blue region) to all the mentions
- Does not consider *local contexts* when assigning type labels
- Introduce *label noise* to the mentions

---

The types assigned to entity Trump include *person, artist, actor, politician, businessman*, while only *{person, politician}* are correct types for the mention *“Trump” in S1*
Label Noise in Entity Typing (cont.)

- Current typing systems either ignore this issue
- assume all candidate labels obtained by supervision are “true” labels

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

- Or use simple pruning heuristics to delete mentions with conflicting types
- aggressive deletion of mentions → significant loss of training data

The larger the target type set, the more severe the loss!
Label Noise Reduction: Task Description

- Define a *new* task, called **Label Noise Reduction in Entity Typing**, to identify the correct type-path for *each mention in training set*, from its *noisy candidate type set*

- **VS. typical typing systems**: they focus on designing models for typing *unlabeled mentions*

- The first systematic study of type label noise in distant supervision

- A fundamental task for entity typing systems (the bottleneck of their performance)

### Problem Definition

**Input:**

- (1) Automatically labeled training corpus: *set of (mention, context, candidate type labels) triples*
- (2) Knowledge base, along with its entity-type facts (i.e., *set of (entity, type) tuples*)
- (3) Target type hierarchy \( T \)

**Output:** Estimate a *single type-path* (not required to end in a leaf node) in the hierarchy \( T \), based on the mention itself as well as its context in the sentence
Label Noise Reduction: Related Work

- **Fine-grained entity typing**
  - Both multi-class classification methods & embedding methods require correct type labels on each instance (entity mention) to train effective models.
  - Ignore label noise when using distant supervision.
  - Use simple pruning heuristics to delete instances with conflicting types.
    - Significant loss of training instances!

- **VS. Label noise reduction**
  - Label noise issue has been studied for other information extraction tasks such as relation extraction and slot filling. However, the form of supervision is different from that in entity typing.
Partial label learning

Consider “only of the candidate labels is correct” (1 among K is correct)

Strategy 1: Assume equal contribution of each candidate label and average the outputs from all candidate labels for prediction

Strategy 3: Treat true label as latent variable and optimize objectives such as maximum likelihood criterion and maximum margin criterion by EM procedure.

VS. Label noise reduction

Leverage label correlation in target type hierarchy
Label Noise Reduction: Challenges

Presence of incorrect type labels in a mention’s candidate type set

- Supervised/semi-supervised techniques both assume “all labels are correct/reliable labels”

- How to accurately estimate the relatedness between mentions and types?

- **Aspect I**: How to model the *noisy associations between mention and its candidate labels*, to indicate the “truth status” of the candidate labels

- **Aspect II**: How to incorporate the *semantic similarity between types*, as we are estimating the type-path holistically for a mention
  - vs. estimating individual labels independently
Label Noise Reduction: Solution Ideas

- Propose a weakly-supervised (unsupervised) approach, where the end goal is to estimate the relatedness between mentions and types

  1. \( \text{sim}(\text{mention, true candidate label}) > \text{sim}(\text{mention, false candidate label}) \)
  2. \( \text{sim}(\text{mention, fine-grained true label}) > \text{sim}(\text{mention, coarse-grained true label}) \)

1. Model the “truth status” of candidate labels as “latent values” using a novel **partial-label loss** \( \rightarrow \) progressively estimate them by incorporating multiple signals:
   - **Co-occurrences between text features and mentions** in the large corpus
   - **Collective associations between type labels and mentions** in the large corpus

2. Model **semantic similarity between types** (i.e., type correlation) derived from KB, to ensure holistic type-path estimation
1. Generate text features and construct a heterogeneous graph
2. Perform joint embedding of the constructed graph $G$ into the same low-dimensional space
3. For each mention, search its candidate type sub-tree in a top-down manner and estimate the true type-path from learned embedding
Text Features for Fine-grained Typing

- **Features are extracted from:**
  - (1) mention’s name string: *e.g.*, head token, POS tags, Brown Cluster of head token
  - (2) mention’s context in the sentence: *e.g.*, n-grams, dependency roles

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Head</td>
<td>Syntactic head token of the mention</td>
<td>“HEAD_Turing”</td>
</tr>
<tr>
<td>Token</td>
<td>Tokens in the mention</td>
<td>“Turing”, “Machine”</td>
</tr>
<tr>
<td>POS</td>
<td>Part-of-Speech tag of tokens in the mention</td>
<td>“NN”</td>
</tr>
<tr>
<td>Character</td>
<td>All character trigrams in the head of the mention</td>
<td>“:tu”, “tur”, ..., “ng:”</td>
</tr>
<tr>
<td>Word Shape</td>
<td>Word shape of the tokens in the mention</td>
<td>“Aa” for “Turing”</td>
</tr>
<tr>
<td>Length</td>
<td>Number of tokens in the mention</td>
<td>“2”</td>
</tr>
<tr>
<td>Context</td>
<td>Unigrams/bigrams before and after the mention</td>
<td>“CXT_B:Maserati ,”, “CXT_A:and the”</td>
</tr>
<tr>
<td>Brown Cluster</td>
<td>Brown cluster ID for the head token (learned using ( D ))</td>
<td>“4_1100”, “8_1101111”, “12_111011111111”</td>
</tr>
</tbody>
</table>

- **“Turing Machine”** is used as an example mention from the sentence:
  - “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.”
Construction of Heterogeneous Graphs

- With three types of objects extracted from corpus: entity mentions, target types, and text features

### Three types of links:

1. **Mention-type link**: represents each mention’s candidate type assignment

2. **Mention-feature link**: captures corpus-level co-occurrences between mentions and text features

3. **Type correlation link**: encodes the type correlation derived from KB or target type hierarchy
Mention-Type Association Subgraph

- Forms a bipartite graph between entity mentions and target types
- Each mention is linked to its candidate types with binary weight
- Some links are “false” links in the constructed mention-type subgraph
- The likelihood of a mention-type link is measured by the relevance between the corresponding mention and type

Example: In sentence S1, context words *democratic* and *presidential* infer that type *politician* is more relevant than type *actor* for mention “Hillary Clinton”

**Hypothesis 1 (Partial Label Association):**

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.

<table>
<thead>
<tr>
<th>ID</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>New York City Mayor Bill de Blasio is heading to Iowa on Friday for four days to campaign for Democratic presidential candidate Hillary Clinton</td>
</tr>
<tr>
<td>S2</td>
<td>Republican presidential candidate Donald Trump spoke during a campaign event in Rock Hill.</td>
</tr>
<tr>
<td>S3</td>
<td>Trump’s company has threatened to withhold up to $1 billion of investment if the U.K. government decides to ban his entry into the country.</td>
</tr>
<tr>
<td>S4</td>
<td>... Trump announced the leaders of his presidential campaign in Louisiana on Tuesday.</td>
</tr>
</tbody>
</table>
Mention-Feature Co-occurrence Subgraph

- Intuition
  - Mentions sharing many text features tend to have close type semantics
  - Text features which co-occur with many entity mentions in the corpus likely represent similar entity types.

**Example**: mentions “Donald Trump” in S2 and “Trump” in S4 share multiple features (e.g., *Trump, presidential* and *campaign*), and thus are likely of the same type *politician*. Conversely, features *campaign* and *presidential* likely represent the same type politician since they co-occur with similar sets of mentions in the corpus.

**Hypothesis 2 (Mention-Feature Co-occurrences)**: 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 Subgraph

- **Simple way**: Use distance in the target type hierarchy
- In target type hierarchy, types closer to each other tend to be more related
- **Example**: actor is more related to artist than to person in the left column

- **Advanced way**: Exploit entity-type facts in KB
- Given two target types, the correlation between them is proportional to the number of entities they share in the KB

**Hypothesis 3 (Type Correlation):**
If high correlation exists between two target types based on either type hierarchy or KB, they should be embedded close to each other.
Heterogeneous Partial-Label Embedding (PLE): The Joint Optimization Problem

\[
\mathcal{O} = \mathcal{O}_{MY} + \mathcal{O}_{MF} + \mathcal{O}_{YY}
\]

\[
\mathcal{O}_{MY} = \sum_{i=1}^{N} \ell_i + \frac{\lambda}{2} \sum_{i=1}^{N} \|u_i\|^2 + \frac{\lambda}{2} \sum_{k=1}^{K} \|v_k\|^2.
\]

\[
\ell_i = \max \left\{ 0, 1 - \left[ \max_{y \in \mathcal{Y}_i} s(m_i, y) - \max_{y' \in \mathcal{Y}_i} s(m_i, y') \right] \right\}
\]

Partial label loss between mentions and types (Hypo 1)

\[
\mathcal{O}_{MF} = - \sum_{(m_i, f_j) \in G_{MF}} w_{ij} \cdot \log p(f_j | m_i)
\]

Model mention-feature links using second-order skip-gram objective (Hypo 2)

\[
\mathcal{O}_{YY} = - \sum_{(y_k, y_{k'}) \in G_{YY}} w_{kk'} \left[ \log p(y_{k'} | y_k) + \log p(y_k | y_{k'}) \right]
\]

Type correlation based on KB (Hypo 3)
PLE: Partial-Label Loss

\[ \ell_i = \max \left\{ 0, 1 - \left[ \max_{y \in \mathcal{Y}_i} s(m_i, y) - \max_{y' \in \overline{\mathcal{Y}_i}} s(m_i, y') \right] \right\} \]

- **Intuition**
  - 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 \( \overline{\mathcal{Y}_i} \), where the scores are measured using current embedding vectors.

- **vs. multi-label learning**
  - A large margin is enforced between all candidate types and non-candidate types without considering noisy types.
PLE: Second-Order Proximity Model

- **Intuition**
  - Nodes with similar distributions over neighbors are similar to each other

- Define the probability of feature \( f_j \) generated by mention \( m_i \) for each link \((m_i, f_j)\) in the mention-feature subgraph as follows

\[
p(f_j | m_i) = \frac{\exp(c_j^T u_i)}{\sum_{f_j' \in \mathcal{F}} \exp(c_{j'}^T u_i)}
\]

- Enforce the conditional distribution specified by embeddings, i.e., \( p(\cdot | m_i) \), to be close to the empirical distribution (i.e., link distribution of \( m_i \) over all features in the mention-feature subgraph)
Learning Algorithm for PLE

Can be efficiently solved by alternative minimization algorithm based on block coordinate descent schema

Algorithm complexity is linear to \#links in the heterogeneous graph

Mini-batch stochastic sub-gradient descent can also be applied for our problem

Algorithm 1: Model Learning of PLE

Input: $G = \{G_{MY}, G_{MF}, G_{YY}\}$, regularization parameter $\lambda$, learning rate $\alpha$, number of negative samples $Z$

Output: entity mention embeddings $\{u_i\}_{i=1}^N$, feature embeddings $\{c_j\}_{j=1}^M$, type embeddings $\{v_k\}_{k=1}^K$

1. Initialize: $\{u_i\}$, $\{c_j\}$, and $\{v_k\}$ as random vectors
2. while $\mathcal{O}$ in Eq. (7) not converge do
   3. for each link in $G_{MF}$ and $G_{YY}$ do
      4. Draw $Z$ negative links from noise distribution $P_n(\cdot)$
   end
   5. for $m_i \in \mathcal{M}$ do
      6. $u_i \leftarrow u_i - \alpha \cdot \partial \mathcal{O} / \partial u_i$ with $\partial \mathcal{O} / \partial u_i$ defined in Eq. (9)
   end
   7. for $f_j \in \mathcal{F}$ do
      8. $c_j \leftarrow c_j - \alpha \cdot \partial \mathcal{O} / \partial c_j$ using $\partial \mathcal{O} / \partial c_j$ defined in Eq. (10)
   end
   9. for $y_k \in \mathcal{Y}$ do
      10. $v_k \leftarrow v_k - \alpha \cdot \partial \mathcal{O} / \partial v_k$ based on $\partial \mathcal{O} / \partial v_k$ in Eq. (11)
      11. $v'_k \leftarrow v'_k - \alpha \cdot \partial \mathcal{O} / \partial v'_k$ using $\partial \mathcal{O} / \partial v'_k$ in Eq. (12)
   end
12. end
Top-Down Type Inference

- Perform top-down search in the candidate type sub-tree to estimate the correct type-path

```
Algorithm 2: Type Inference
```

```
Input: candidate type sub-tree \( \mathcal{Y}_i \), mention embeddings \( \{u_i\} \), type embeddings \( \{v_k\} \), threshold \( \eta \)
Output: estimated type-path \( \mathcal{Y}_i^* \) for \( m_i \in \mathcal{M} \)
1 for \( m_i \in \mathcal{M} \) do
2     Initialize: \( \mathcal{Y}_i \) as \( \emptyset \), \( r \) as the root of \( \mathcal{Y} \)
3     while \( C_i(r) \neq \emptyset \) do
4         \( r \leftarrow \arg \max_{y_k \in C_i(r)} s(u_i, v_k) \)
5         if \( s(u_i, v_r) > \eta \) then
6             Update the type-path: \( \mathcal{Y}_i^* \leftarrow \mathcal{Y}_i^* \cup \{r\} \)
7         else
8             \( \text{return } \mathcal{Y}_i^* \) as the estimated type-path for \( m_i \)
9     end
10 end
```
Experiment Setting

- Datasets:
  - (1) Wiki: 1.5M sentences sampled from ~780k Wikipedia articles
  - (2) OntoNotes: 13,109 news
  - (3) BBN: 2,311 Wall Street Journal articles

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

- **Compared Methods**
  1. **Sib**: removes siblings types;
  2. **Min**: removes types that appear only once in the document;
  3. **All**: first performs Sib pruning then Min pruning;
  4. **DeepWalk**: embedding a homogeneous graph with binary edges;
  5. **LINE**: second-order LINE;
  6. **WSABIE**: adopts WARP loss with kernel extension;
  7. **PTE**: applied PTE joint training algorithm on subgraphs $G_M$ and $G_Y$.
  8. **PL-SVM**: uses a margin-based loss to handle label noise.
  9. **CLPL**: uses a linear model to encourage large average scores for candidate types.

- For PLE, we compare
  1. **PLE**: adopts KB-based type correlation subgraph;
  2. **PLE-CoH**: adopts type hierarchy-based correlation subgraph;
  3. **PLE-NoCo**: does not consider type correlation.
Intrinsic Experiments: Effectiveness of Label Noise Reduction

- **Goal:** compare how accurately PLE and the other methods can estimate the true types of mentions from its noisy candidate type set

<table>
<thead>
<tr>
<th>Method</th>
<th>Wiki Acc</th>
<th>Wiki Ma-P</th>
<th>Wiki Ma-R</th>
<th>Wiki Ma-F1</th>
<th>OntoNotes Acc</th>
<th>OntoNotes Ma-P</th>
<th>OntoNotes Ma-R</th>
<th>OntoNotes Ma-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw</td>
<td>0.373</td>
<td>0.558</td>
<td>0.681</td>
<td>0.614</td>
<td>0.521</td>
<td>0.719</td>
<td>0.605</td>
<td></td>
</tr>
<tr>
<td>Sib [7]</td>
<td>0.373</td>
<td>0.583</td>
<td>0.636</td>
<td>0.608</td>
<td>0.578</td>
<td>0.653</td>
<td>0.613</td>
<td></td>
</tr>
<tr>
<td>Min [7]</td>
<td>0.373</td>
<td>0.561</td>
<td>0.679</td>
<td>0.615</td>
<td>0.524</td>
<td>0.717</td>
<td>0.606</td>
<td></td>
</tr>
<tr>
<td>All [7]</td>
<td>0.373</td>
<td>0.585</td>
<td>0.634</td>
<td>0.608</td>
<td>0.581</td>
<td>0.651</td>
<td>0.614</td>
<td></td>
</tr>
<tr>
<td>DeepWalk-Raw [21]</td>
<td>0.328</td>
<td>0.598</td>
<td>0.459</td>
<td>0.519</td>
<td>0.595</td>
<td>0.367</td>
<td>0.454</td>
<td></td>
</tr>
<tr>
<td>LINE-Raw [29]</td>
<td>0.349</td>
<td>0.600</td>
<td>0.596</td>
<td>0.598</td>
<td>0.590</td>
<td>0.610</td>
<td>0.600</td>
<td></td>
</tr>
<tr>
<td>WSABIE-Raw [34]</td>
<td>0.332</td>
<td>0.554</td>
<td>0.609</td>
<td>0.580</td>
<td>0.557</td>
<td>0.633</td>
<td>0.592</td>
<td></td>
</tr>
<tr>
<td>PTE-Raw [28]</td>
<td>0.419</td>
<td>0.678</td>
<td>0.597</td>
<td>0.635</td>
<td>0.686</td>
<td>0.607</td>
<td>0.644</td>
<td></td>
</tr>
<tr>
<td>PLE-NoCo</td>
<td>0.556</td>
<td>0.795</td>
<td>0.678</td>
<td>0.732</td>
<td>0.804</td>
<td>0.668</td>
<td>0.730</td>
<td></td>
</tr>
<tr>
<td>PLE-CoH</td>
<td>0.568</td>
<td>0.805</td>
<td>0.671</td>
<td>0.732</td>
<td>0.808</td>
<td>0.704</td>
<td>0.752</td>
<td></td>
</tr>
<tr>
<td>PLE</td>
<td><strong>0.589</strong></td>
<td><strong>0.840</strong></td>
<td><strong>0.749</strong></td>
<td><strong>0.833</strong></td>
<td><strong>0.833</strong></td>
<td><strong>0.705</strong></td>
<td><strong>0.763</strong></td>
<td></td>
</tr>
</tbody>
</table>

40.57% improvement in Accuracy and 23.89% improvement in Macro-Precision compared to the best baseline on Wiki dataset

- **vs. pruning strategies:** LNR identifies true types from the candidate type sets instead of aggressively deleting instances with noisy type labels
- **vs. other embedding methods:** PLE obtains superior performance because it effectively models the noisy type labels
- **vs. PLE variants:** (i) PLE captures type semantic similarity; (ii) modeling type correlation with entity-type facts in KB yields more accurate and complete type correlation statistics than type hierarchy-based approach
Intrinsic Experiments: Effectiveness of Label Noise Reduction

Example output on news articles

<table>
<thead>
<tr>
<th>Text</th>
<th>NASA says it may decide by tomorrow whether another space walk will be needed ...</th>
<th>... the board of directors which are composed of twelve members directly appointed by the Queen.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cand. type set</td>
<td>person, artist, location, structure, organization, company, news_company</td>
<td>person, artist, actor, author, person_title, politician</td>
</tr>
<tr>
<td>WSABIE</td>
<td>person, artist</td>
<td>person, artist</td>
</tr>
<tr>
<td>PTE</td>
<td>organization, company, news_company</td>
<td>person, artist</td>
</tr>
<tr>
<td>PLE</td>
<td>organization, company</td>
<td>person, person_title</td>
</tr>
</tbody>
</table>

- PLE predicts fine-grained types with better accuracy (e.g., person_title)
- and avoids from overly-specific predictions (e.g., news_company)
Intrinsic Experiments: Effectiveness of Label Noise Reduction

- Testing the effect of training set size
  - Performance of all methods improves as the ratio increases, and becomes insensitive as the sampling ratio > 0.7
- Testing the effect of training set size
  - Performance of PLE becomes insensitive as becomes small enough (i.e., 0.01)
Extrinsic Experiments: Fine-Grained Entity Typing

- Compare performance gain of two state-of-the-art typing systems, when using denoised training data output by different compared methods

<table>
<thead>
<tr>
<th>Typing System</th>
<th>Noise Reduction Method</th>
<th>Wiki Acc</th>
<th>Ma-F1</th>
<th>Mi-F1</th>
<th>OntoNotes Acc</th>
<th>Ma-F1</th>
<th>Mi-F1</th>
<th>BBN Acc</th>
<th>Ma-F1</th>
<th>Mi-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>PL-SVM [20]</td>
<td>0.428</td>
<td>0.613</td>
<td>0.571</td>
<td>0.465</td>
<td>0.648</td>
<td>0.582</td>
<td>0.497</td>
<td>0.679</td>
<td>0.677</td>
</tr>
<tr>
<td>N/A</td>
<td>CLPL [2]</td>
<td>0.162</td>
<td>0.431</td>
<td>0.411</td>
<td>0.438</td>
<td>0.603</td>
<td>0.536</td>
<td>0.486</td>
<td>0.561</td>
<td>0.582</td>
</tr>
<tr>
<td>HYENA [35]</td>
<td>Raw</td>
<td>0.288</td>
<td>0.528</td>
<td>0.506</td>
<td>0.249</td>
<td>0.497</td>
<td>0.446</td>
<td>0.523</td>
<td>0.576</td>
<td>0.587</td>
</tr>
<tr>
<td></td>
<td>Min [7]</td>
<td>0.325</td>
<td>0.566</td>
<td>0.536</td>
<td>0.295</td>
<td>0.523</td>
<td>0.470</td>
<td>0.524</td>
<td>0.582</td>
<td>0.595</td>
</tr>
<tr>
<td></td>
<td>All [7]</td>
<td>0.417</td>
<td>0.591</td>
<td>0.545</td>
<td>0.305</td>
<td>0.552</td>
<td>0.495</td>
<td>0.495</td>
<td>0.563</td>
<td>0.568</td>
</tr>
<tr>
<td></td>
<td>WSABIE-Min [34]</td>
<td>0.199</td>
<td>0.462</td>
<td>0.459</td>
<td>0.400</td>
<td>0.565</td>
<td>0.521</td>
<td>0.524</td>
<td>0.610</td>
<td>0.621</td>
</tr>
<tr>
<td></td>
<td>PTE-Min [28]</td>
<td>0.238</td>
<td>0.542</td>
<td>0.522</td>
<td>0.452</td>
<td>0.626</td>
<td>0.572</td>
<td>0.545</td>
<td>0.639</td>
<td>0.650</td>
</tr>
<tr>
<td></td>
<td>PLE-NoCo</td>
<td>0.517</td>
<td>0.672</td>
<td>0.634</td>
<td>0.496</td>
<td>0.658</td>
<td>0.603</td>
<td>0.650</td>
<td>0.709</td>
<td>0.703</td>
</tr>
<tr>
<td></td>
<td>PLE</td>
<td>0.543</td>
<td>0.695</td>
<td>0.681</td>
<td>0.546</td>
<td>0.692</td>
<td>0.625</td>
<td>0.692</td>
<td>0.731</td>
<td>0.732</td>
</tr>
<tr>
<td>FIGER [14]</td>
<td>Raw</td>
<td>0.474</td>
<td>0.692</td>
<td>0.655</td>
<td>0.369</td>
<td>0.578</td>
<td>0.516</td>
<td>0.467</td>
<td>0.672</td>
<td>0.612</td>
</tr>
<tr>
<td></td>
<td>Min</td>
<td>0.453</td>
<td>0.691</td>
<td>0.631</td>
<td>0.373</td>
<td>0.570</td>
<td>0.509</td>
<td>0.444</td>
<td>0.671</td>
<td>0.613</td>
</tr>
<tr>
<td></td>
<td>All</td>
<td>0.453</td>
<td>0.648</td>
<td>0.582</td>
<td>0.400</td>
<td>0.618</td>
<td>0.548</td>
<td>0.461</td>
<td>0.636</td>
<td>0.583</td>
</tr>
<tr>
<td></td>
<td>WSABIE-Min</td>
<td>0.455</td>
<td>0.646</td>
<td>0.601</td>
<td>0.425</td>
<td>0.603</td>
<td>0.546</td>
<td>0.481</td>
<td>0.671</td>
<td>0.618</td>
</tr>
<tr>
<td></td>
<td>PTE-Min</td>
<td>0.476</td>
<td>0.670</td>
<td>0.635</td>
<td>0.494</td>
<td>0.675</td>
<td>0.618</td>
<td>0.513</td>
<td>0.674</td>
<td>0.657</td>
</tr>
<tr>
<td></td>
<td>PLE-NoCo</td>
<td>0.543</td>
<td>0.726</td>
<td>0.705</td>
<td>0.547</td>
<td>0.699</td>
<td>0.639</td>
<td>0.643</td>
<td>0.753</td>
<td>0.721</td>
</tr>
<tr>
<td></td>
<td>PLE</td>
<td>0.599</td>
<td>0.763</td>
<td>0.749</td>
<td>0.572</td>
<td>0.715</td>
<td>0.661</td>
<td>0.685</td>
<td>0.777</td>
<td>0.750</td>
</tr>
</tbody>
</table>

- vs. other noise reduction methods: the effectiveness of the proposed margin-based loss in modeling noisy candidate types

- vs. partial-label learning methods: PLE obtains superior performance because it jointly models type correlation derived from KB and feature-mention co-occurrences in the corpus
Case Analyses

- Testing at different type levels
  - It is more difficult to distinguish among deeper (more fine-grained) types.
  - PLE always outperforms the other two method, and achieves a 153% improvement in Accuracy.

![Accuracy on different type levels](chart)

(a) Test at different type levels

![Micro-F1 w.r.t. Re-training Iteration](chart)

(b) Iterative Re-training

- Iterative re-training of PLE
  - Analyze the effect of boostrapping PLE
  - The performance gain becomes marginal after 3 iterations of re-training