Inferring Missing Entity Type Instances for Knowledge Base Completion: New Dataset and Methods

Arvind Neelakantan∗
Department of Computer Science
University of Massachusetts, Amherst
Amherst, MA, 01003
arvind@cs.umass.edu

Ming-Wei Chang
Microsoft Research
1 Microsoft Way
Redmond, WA 98052, USA
minchang@microsoft.com

Abstract

Most of previous work in knowledge base (KB) completion has focused on the problem of relation extraction. In this work, we focus on the task of inferring missing entity type instances in a KB, a fundamental task for KB competition yet receives little attention. Due to the novelty of this task, we construct a large-scale dataset and design an automatic evaluation methodology. Our knowledge base completion method uses information within the existing KB and external information from Wikipedia. We show that individual methods trained with a global objective that considers unobserved cells from both the entity and the type side gives consistently higher quality predictions compared to baseline methods. We also perform manual evaluation on a small subset of the data to verify the effectiveness of our knowledge base completion methods and the correctness of our proposed automatic evaluation method.

1 Introduction

There is now increasing interest in the construction of knowledge bases like Freebase (Bollacker et al., 2008) and NELL (Carlson et al., 2010) in the natural language processing community. KBs contain facts such as Tiger Woods is an athlete, and Barack Obama is the president of USA. However, one of the main drawbacks in existing KBs is that they are incomplete and are missing important facts (West et al., 2014). jeopardizing their usefulness in downstream tasks such as question answering. This has led to the task of completing the knowledge base entries, or Knowledge Base Completion (KBC) extremely important.

In this paper, we address an important subproblem of knowledge base completion—inferring missing entity type instances. Most of previous work in KB completion has only focused on the problem of relation extraction (Mintz et al., 2009, Nickel et al., 2011, Bordes et al., 2013, Riedel et al., 2013). Entity type information is crucial in KBs and is widely used in many NLP tasks such as relation extraction (Chang et al., 2014), coreference resolution (Ratinov and Roth, 2012, Hajishirzi et al., 2013), entity linking (Fang and Chang, 2014), semantic parsing (Kwiatkowski et al., 2013, Berant et al., 2013) and question answering (Bordes et al., 2014, Yao and Durme, 2014). For example, adding entity type information improves relation extraction by 3% (Chang et al., 2014) and entity linking by 4.2 F1 points (Guo et al., 2013). Despite their importance, there is surprisingly little previous work on this problem and, there are no datasets publicly available for evaluation.

We construct a large-scale dataset for the task of inferring missing entity type instances in a KB. Most of previous KBC datasets (Mintz et al., 2009, Riedel et al., 2013) are constructed using a single snapshot of the KB and methods are evaluated on a subset of facts that are hidden during training. Hence, the methods could be potentially evaluated by their ability to predict easy facts that the KB already contains. Moreover, the methods are not directly evaluated
on their ability to predict missing facts. To overcome these drawbacks we construct the train and test data using two snapshots of the KB and evaluate the methods on predicting facts that are added to the more recent snapshot, enabling a more realistic and challenging evaluation.

Standard evaluation metrics for KBC methods are generally type-based (Mintz et al., 2009; Riedel et al., 2013), measuring the quality of the predictions by aggregating scores computed within a type. This is not ideal because: (1) it treats every entity type equally not considering the distribution of types, (2) it does not measure the ability of the methods to rank predictions across types. Therefore, we additionally use a global evaluation metric, where the quality of predictions is measured within and across types, and also accounts for the high variance in type distribution. In our experiments, we show that models trained with negative examples from the entity side perform better on type-based metrics, while when trained with negative examples from the type side perform better on the global metric.

In order to design methods that can rank predictions both within and across entity (or relation) types, we propose a global objective to train the models. Our proposed method combines the advantages of previous approaches by using negative examples from both the entity and the type side. When considering the same number of negative examples, we find that the linear classifiers and the low-dimensional embedding models trained with the global objective produce better quality ranking within and across entity types when compared to training with negatives examples only from entity or type side. Additionally compared to prior methods, the model trained on the proposed global objective can more reliably suggest confident entity-type pair candidates that could be added into the given knowledge base.

Our contributions are summarized as follows:

- We develop an evaluation framework comprising of methods for dataset construction and evaluation metrics to evaluate KBC approaches for missing entity type instances. The dataset and evaluation scripts are publicly available at http://research.microsoft.com/en-US/downloads/df481862-65cc-4b05-886c-accl8lad07bb/default.aspx.

- We propose a global training objective for KBC methods. The experimental results show that both linear classifiers and low-dimensional embedding models achieve best overall performance when trained with the global objective function.

- We conduct extensive studies on models for inferring missing type instances studying the impact of using various features and models.

2 Inferring Entity Types

We consider a KB $\Lambda$ containing entity type information of the form $(e, t)$, where $e \in E$ ($E$ is the set of all entities) is an entity in the KB with type $t \in T$ ($T$ is the set of all types). For example, $e$ could be Tiger Woods and $t$ could be sports athlete. As a single entity can have multiple types, entities in Freebase often miss some of their types. The aim of this work is to infer missing entity type instances in the KB. Given an unobserved fact (an entity-type pair) in the training data $(e, t) \not\in \Lambda$ where entity $e \in E$ and type $t \in T$, the task is to infer whether the KB currently misses the fact, i.e., infer whether $(e, t) \in \Lambda$. We consider entities in the intersection of Freebase and Wikipedia in our experiments.

2.1 Information Resources

Now, we describe the information sources used to construct the feature representation of an entity to
infer its types. We use information in Freebase and external information from Wikipedia to complete the KB.

- **Entity Type Features**: The entity types observed in the training data can be a useful signal to infer missing entity type instances. For example, in our snapshot of Freebase, it is not uncommon to find an entity with the type /people/deceased_person but missing the type /people/person.

- **Freebase Description**: Almost all entities in Freebase have a short one paragraph description of the entity. Figure 1 shows the Freebase description of Jean Metellus that can be used to infer the type /book/author which Freebase does not contain as the date of writing this article.

- **Wikipedia**: As external information, we include the Wikipedia full text article of an entity in its feature representation. We consider entities in Freebase that have a link to their Wikipedia article. The Wikipedia full text of an entity gives several clues to predict its entity types. For example, Figure 2 shows a section of the Wikipedia article of Claire Martin which gives clues to infer the type /award/award_winner that Freebase misses.

3 Evaluation Framework

In this section, we propose an evaluation methodology for the task of inferring missing entity type instances in a KB. While we focus on recovering entity types, the proposed framework can be easily adapted to relation extraction as well.

First, we discuss our two-snapshot dataset construction strategy. Then we motivate the importance of evaluating KBC algorithms globally and describe the evaluation metrics we employ.

3.1 Two Snapshots Construction

In most previous work on KB completion to predict missing relation facts (Mintz et al., 2009; Riedel et al., 2013), the methods are evaluated on a subset of facts from a single KB snapshot, that are hidden while training. However, given that the missing entries are usually selected randomly, the distribution of the selected unknown entries could be very different from the actual missing facts distribution. Also, since any fact could be potentially used for evaluation, the methods could be evaluated on their ability to predict easy facts that are already present in the KB.

To overcome this drawback, we construct our train and test set by considering two snapshots of the knowledge base. The train snapshot is taken from an earlier time without special treatment. The test snapshot is taken from a later period, and a KBC algorithm is evaluated by its ability of recovering newly added knowledge in the test snapshot. This enables the methods to be directly evaluated on facts that are missing in a KB snapshot. Note that the facts that are added to the test snapshot, in general, are more subtle than the facts that they already contain and predicting the newly added facts could be harder. Hence, our approach enables a more realistic and challenging evaluation setting than previous work.

We use manually constructed Freebase as the KB in our experiments. Notably, Chang et al. (2014) use a two-snapshot strategy for constructing a dataset for relation extraction using automatically constructed NELL as their KB. The new facts that are added to a KB by an automatic method may not have all the characteristics that make the two snapshot strategy more advantageous.

We construct our train snapshot $\Lambda_0$ by taking the Freebase snapshot on 3rd September, 2013 and consider entities that have a link to their Wikipedia article. The Wikipedia full text of an entity gives several clues to predict its entity types. For example, Figure 2 shows a section of the Wikipedia article of Claire Martin which gives clues to infer the type /award/award_winner that Freebase misses.

Using the entire set of negative examples in the test data is impractical due to the large number of negative examples. To avoid this we only add the negative types

1Note that some of the negative instances used in training could be positive instances in test but we do not remove them during training.
of entities that have at least one new fact in the test data. Additionally, we add a portion of the negative examples for entities which do not have new fact in the test data and that were unused during training. This makes our dataset quite challenging since the number of negative instances is much larger than the number of positive instances in the test data.

It is important to note that the goal of this work is not to predict facts that emerged between the time period of the train and test snapshot. For example, we do not aim to predict the type award_winner for an entity that won an award after 3rd September, 2013. Hence, we use the Freebase description in the training data snapshot and Wikipedia snapshot on 3rd September, 2013 to get the features for entities.

One might worry that the new snapshot might contain a significant amount of emerging facts so it could not be an effective way to evaluate the KBC algorithms. Therefore, we examine the difference between the training snapshot and test snapshot manually and found that this is likely not the case. For example, we randomly selected 25 award_winner instances that were added to the test snapshot and found that all of them had won at least one award before 3rd September, 2013.

Note that while this automatic evaluation is closer to the real-world scenario, it is still not perfect as the new KB snapshot is still incomplete. Therefore, we also perform human evaluation on a small dataset to verify the effectiveness of our approach.

2 In this work, we also do not aim to correct existing false positive errors in Freebase

3.2 Global Evaluation Metric

Mean average precision (MAP) (Manning et al., 2008) is now commonly used to evaluate KB completion methods (Mintz et al., 2009; Riedel et al., 2013). MAP is defined as the mean of average precision over all entity (or relation) types. MAP treats each entity type equally (not explicitly accounting for their distribution). However, some types occur much more frequently than others. For example, in our large-scale experiment with 500 entity types, there are many entity types with only 5 instances in the test set while the most frequent entity type has tens of thousands of missing instances. Moreover, MAP only measures the ability of the methods to correctly rank predictions within a type.

To account for the high variance in the distribution of entity types and measure the ability of the methods to correctly rank predictions across types we use global average precision (GAP) (similarly to micro-F1) as an additional evaluation metric for KB completion. We convert the multi-label classification problem to a binary classification problem where the label of an entity and type pair is true if the entity has that type in Freebase and false otherwise. GAP is the average precision of this transformed problem which can measure the ability of the methods to rank predictions both within and across entity types.

Prior to us, Bordes et al. (2013) use mean reciprocal rank as a global evaluation metric for a KBC task. We use average precision instead of mean reciprocal rank since MRR could be biased to the top predictions of the method (West et al., 2014).

While GAP captures global ordering, it would be...
beneficial to measure the quality of the top $k$ predictions of the model for bootstrapping and active learning scenarios (Lewis and Gale, 1994; Cucerzan and Yarowsky, 1999). We report G@k, GAP measured on the top $k$ predictions (similarly to Precision@k and Hits@k). This metric can be reliably used to measure the overall quality of the top $k$ predictions.

4 Global Objective for Knowledge Base Completion

We describe our approach for predicting missing entity types in a KB in this section. While we focus on recovering entity types in this paper, the methods we develop can be easily extended to other KB completion tasks.

4.1 Global Objective Framework

During training, only positive examples are observed in KB completion tasks. Similar to previous work (Mintz et al., 2009; Bordes et al., 2013; Riedel et al., 2013), we get negative training examples by treating the unobserved data in the KB as negative examples. Because the number of unobserved examples is much larger than the number of facts in the KB, we follow previous methods and sample few unobserved negative examples for every positive example.

Previous methods largely neglect the sampling methods on unobserved negative examples. The proposed global objective framework allows us to systematically study the effect of the different sampling methods to get negative data, as the performance of the model for different evaluation metrics does depend on the sampling method.

We consider a training snapshot of the KB $\Lambda_0$, containing facts of the form $(e, t)$ where $e$ is an entity in the KB with type $t$. Given a fact $(e, t)$ in the KB, we consider two types of negative examples constructed from the following two sets: $N_E(e, t)$ is the “negative entity set”, and $N_T(e, t)$ is the “negative type set”. More precisely,

$$N_E(e, t) \subset \{e' | e' \in E, e' \neq e, (e', t) \notin \Lambda_0\},$$

and

$$N_T(e, t) \subset \{t' | t' \in T, t' \neq t, (e, t') \notin \Lambda_0\}.$$
Algorithm 1 The training algorithm for Linear.Adagrad.

1: Initialize $w_t = 0, \forall t = 1 \ldots |T|$
2: for $(e, t) \in A_0$ do
3: for $e' \in N_E(e, t)$ do
4: if $w_t^T \Phi(e) - w_t^T \Phi(e') < 0$ then
5: AdaGradUpdate($w_t, \Phi(e') - \Phi(e)$)
6: end if
7: end for
8: for $t' \in N_T(e, t)$ do
9: if $w_t^T \Phi(e) - w_t^T \Phi(e') < 0$ then
10: AdaGradUpdate($w_t, -\Phi(e)$)
11: AdaGradUpdate($w_{t'}, \Phi(e)$).
12: end if
13: end for
14: end for

Algorithm 2 The training algorithm for the embedding model.

1: Initialize $V$, $U$ randomly.
2: for $(e, t) \in A_0$ do
3: for $e' \in N_E(e, t)$ do
4: if $s(e, t) - s(e', t) < 0$ then
5: $\mu \leftarrow V^T \Phi(t)$
6: $\eta \leftarrow U^T (\Phi(e') - \Phi(e))$
7: for $i \in 1 \ldots d$ do
8: AdaGradUpdate($U_i, \mu[i](\Phi(e') - \Phi(e))$
9: end for
10: end if
11: end for
12: end for
13: for $t' \in N_T(e, t)$ do
14: if $s(e, t) - s(e, t') < 0$ then
15: $\mu \leftarrow V^T (\Phi(t') - \Phi(t))$
16: $\eta \leftarrow U^T \Phi(e)$
17: for $i \in 1 \ldots d$ do
18: AdaGradUpdate($U_i, \mu[i](\Phi(e))$
19: end for
20: end for
21: end if
22: end for
23: end for

4.2 Algorithms

We propose three different algorithms based on the global objective framework for predicting missing entity types. Two algorithms use the linear model and the other one uses the embedding model.

**Linear Model** The scoring function in this model is given by $s(e, t | \theta = \{w_t\}) = w_t^T \Phi(e)$, where $w_t \in R^{d_{x}}$ is the parameter vector for target type $t$. The regularization term in Eq. (1) is defined as follows: $R(\theta) = 1/2 \sum_{t=1}^{T} w_t^T w_t$. We use $k = 2$ in our experiments. Our first algorithm is obtained by using the dual coordinate descent algorithm (Hsieh et al., 2008) to optimize Eq. (1), where we modified the original algorithm to handle multiple weight vectors. We refer to this algorithm as **Linear.DCD**.

While DCD algorithm ensures convergence to the global optimum solution, its convergence can be slow in certain cases. Therefore, we adopt an online algorithm, Adagrad (Duchi et al., 2011). We use the hinge loss function $(k = 1)$ with no regularization $(Reg(\theta) = \emptyset)$ since it gave best results in our initial experiments. We refer to this algorithm as **Linear.Adagrad**, which is described in Algorithm 1. Note that AdaGradUpdate$(x, g)$ is a procedure which updates the vector $x$ with the respect to the gradient $g$.

**Embedding Model** In this model, vector representations are constructed for entities and types using linear projection matrices. Recall $\Psi(t) \rightarrow R^{d_{x}}$ is the feature function that maps a type to its feature representation. The scoring function is given by $s(e, t | \theta = (U, V)) = \Psi(t)^T V U^T \Phi(e)$, where $U \in R^{d_{x} \times d}$ and $V \in R^{d_t \times d}$ are projection matrices that embed the entities and types in a $d$-dimensional space. Similarly to the linear classifier model, we use the 11-hinge loss function $(k = 1)$ with no regularization $(Reg(\theta) = \emptyset)$. $U_i$ and $V_i$ denote the $i$-th column vector of the matrix $U$ and $V$, respectively. The algorithm is described in detail in Algorithm 2.

The embedding model has more expressive power than the linear model, but the training unlike in the linear model, converges only to a local optimum solution since the objective function is non-convex.

4.3 Relationship to Existing Methods

Many existing methods for relation extraction and entity type prediction can be cast as a special case under the global objective framework. For example, we can consider the work in relation extraction (Mintz et al., 2009; Bordes et al., 2013; Riedel et al., 2013) as models trained with $N_T(e, t) = \emptyset$. These models are trained only using negative entities which we refer to as Negative Entity (NE) objective. The entity type prediction model in Ling and Weld (2012) is a linear model with $N_E(e, t) = \emptyset$ which
we refer to as the Negative Type (NT) objective. The embedding model described in Weston et al. (2011) developed for image retrieval is also a special case of our model trained with the NT objective.

While the NE or NT objective functions could be suitable for some classification tasks (Weston et al., 2011), the choice of objective functions for the KBC tasks has not been well motivated. Often the choice is made neither with theoretical foundation nor with empirical support. To the best of our knowledge, the global objective function, which includes both $N_E(e, t)$ and $N_T(e, t)$, has not been considered previously by KBC methods.

5 Experiments

In this section, we give details about our dataset and discuss our experimental results. Finally, we perform manual evaluation on a small subset of the data.

5.1 Data

First, we evaluate our methods on 70 entity types with the most observed facts in the training data. We also perform large-scale evaluation by testing the methods on 500 types with the most observed facts in the training data.

Table 1 shows statistics of our dataset. $\Lambda_0$ is our training snapshot and $\Lambda$ is our test snapshot. An example is an entity-type pair.

Table 1: Statistics of our dataset. $\Lambda_0$ is our training snapshot and $\Lambda$ is our test snapshot. An example is an entity-type pair.

|                | 70 types | 500 types |
|----------------|----------|-----------|
| Entities       | 2.2M     | 2.2M      |
| Training Data Statistics ($\Lambda_0$) |
| positive example | 4.5M   | 6.2M      |
| max #ent for a type | 1.1M   | 1.1M      |
| min #ent for a type | 6732   | 32        |
| Test Data Statistics ($\Lambda - \Lambda_0$) |
| positive examples | 163K   | 240K      |
| negative examples | 17.1M  | 132M      |
| negative/positive ratio | 105.22 | 554.44    |

5.2 Automatic Evaluation Results

Table 2 shows automatic evaluation results where we give results on 70 types and 500 types. We compare different aspects of the system on 70 types empirically.

Adagrad Vs DCD We first study the linear models by comparing Linear.DCD and Linear.AdaGrad. Table 2a shows that Linear.AdaGrad consistently performs better for our task.

Impact of Features We compare the effect of different features on the final performance using Linear.AdaGrad in Table 2b. Types are represented by boolean features while Freebase description and Wikipedia full text are represented using tf-idf weighting. The best MAP results are obtained by using all the information (T+D+W) while best GAP results are obtained by using the Freebase description and Wikipedia article of the entity. Note that the features are simply concatenated when multiple resources are used. We tried to use *idf* weighting on type features and on all features, but they did not yield improvements.

The Importance of Global Objective Table 2c and 2d compares global training objective with NE and NT training objective. Note that all the three methods use the same number of negative examples. More precisely, for each $(e, t) \in \Lambda_0$, $|N_E(e, t)| + |N_T(e, t)| = m + n = 2$. The results show that the global training objective achieves best scores on both MAP and GAP for classifiers and low-dimensional embedding models. Among NE and NT, NE performs better on the type-based metric while NT performs better on the global metric.

Linear Model Vs Embedding Model Finally, we compare the linear classifier model with the embedding model in Table 2e. The linear classifier model performs better than the embedding model in both MAP and GAP.

We perform large-scale evaluation on 500 types with the description features (as experiments are expensive) and the results are shown in Table 2f.
One might expect that with the increased number of types, the embedding model would perform better than the classifier since they share parameters across types. However, despite the recent popularity of embedding models in NLP, linear model still performs better in our task.

### 5.3 Human Evaluation

To verify the effectiveness of our KBC algorithms, and the correctness of our automatic evaluation method, we perform manual evaluation on the top 100 predictions of the output obtained from two different experimental setting and the results are shown in Table 3. Even though the automatic evaluation gives pessimistic results since the test KB is also incomplete, the results indicate that the automatic evaluation is correlated with manual evaluation. More excitingly, among the 179 unique instances we manually evaluated, 17 of them are still missing in Freebase which emphasizes the effectiveness of our approach.

---

6 This is true even with existing automatic evaluation methods.

7 at submission time.

---

Table 2: Automatic Evaluation Results. Note that $m = |\mathcal{N}_E(e, t)|$ and $n = |\mathcal{N}_T(e, t)|$. 

---
Table 3: Manual vs. Automatic evaluation of top 100 predictions on 70 types. Predictions are obtained by training a linear classifier using Adagrad with global training objective (m=1, n=1). G@100-M and Accuracy-M are computed by manual evaluation.

5.4 Error Analysis

- **Effect of training data:** We find the performance of the models on a type is highly dependent on the number of training instances for that type. For example, the linear classifier model when evaluated on 70 types performs 24.86% better on the most frequent 35 types compared to the least frequent 35 types. This indicates bootstrapping or active learning techniques can be profitably used to provide more supervision for the methods. In this case, G@k would be an useful metric to compare the effectiveness of the different methods.

- **Shallow Linguistic features:** We found some of the false positive predictions are caused by the use of shallow linguistic features. For example, an entity who has acted in a movie and composes music only for television shows is wrongly tagged with the type /film/composer since words like "movie", "composer" and "music" occur frequently in the Wikipedia article of the entity (http://en.wikipedia.org/wiki/J._J._Abrams).

6 Related Work

**Entity Type Prediction and Wikipedia Features** Much of previous work (Pantel et al., 2012; Ling and Weld, 2012) in entity type prediction has focused on the task of predicting entity types at the sentence level. Yao et al. (2013) develop a method based on matrix factorization for entity type prediction in a KB using information within the KB and New York Times articles. However, the method was still evaluated only at the sentence level. Toral and Munoz (2006), Kazama and Torisawa (2007) use the first line of an entity’s Wikipedia article to perform named entity recognition on three entity types.

**Knowledge Base Completion** Much of precious work in KB completion has focused on the problem of relation extraction. Majority of the methods infer missing relation facts using information within the KB (Nickel et al., 2011; Lao et al., 2011; Socher et al., 2013; Bordes et al., 2013) while methods such as Mintz et al. (2009) use information in text documents. Riedel et al. (2013) use both information within and outside the KB to complete the KB.

**Linear Embedding Model** Weston et al. (2011) is one of first work that developed a supervised linear embedding model and applied it to image retrieval. We apply this model to entity type prediction but we train using a different objective function which is more suited for our task.

7 Conclusion and Future Work

We propose an evaluation framework comprising of methods for dataset construction and evaluation metrics to evaluate KBC approaches for inferring missing entity type instances. We verified that our automatic evaluation is correlated with human evaluation, and our dataset and evaluation scripts are publicly available[8]. Experimental results show that models trained with our proposed global training objective produces higher quality ranking within and across types when compared to baseline methods.

In future work, we plan to use information from entity linked documents to improve performance and also explore active leaning, and other human-in-the-loop methods to get more training data.

References

[Berant et al.2013] Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, and Christopher D. Manning. 2013. Semantic parsing on freebase from question-answer pairs. In Empirical Methods in Natural Language Processing.

[Bollacker et al.2008] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In
Joint knowledge for named entity recognition. In 2007. Exploiting wikipedia as external knowledge for named entity recognition. In Advances in Neural Information Processing Systems.

[Bordes et al.2014] Antoine Bordes, Sumit Chopra, and Jason Weston. 2014. Question answering with subgraph embeddings. In Empirical Methods in Natural Language Processing.

[Carlson et al.2010] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R. Hruschka, and A. 2010. Toward an architecture for never-ending language learning. In In AAAI.

[Chang et al.2014] Kai-Wei Chang, Wen tau Yih, Bishan Yang, and Christopher Meek. 2014. Typed tensor decomposition of knowledge bases for relation extraction. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing.

[Cucerzan and Yarowsky1999] Silviu Cucerzan and David Yarowsky. 1999. Language independent named entity recognition combining morphological and contextual evidence. In 1999. Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

[Duchi et al.2011] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. In Journal of Machine Learning Research.

[Fang and Chang2014] Yuan Fang and Ming-Wei Chang. 2014. Entity linking on microblogs with spatial and temporal signals. In Transactions of the Association for Computational Linguistics.

[Guo et al.2013] Stephen Guo, Ming-Wei Chang, and Emre Kiciman. 2013. To link or not to link? a study on end-to-end tweet entity linking. In The North American Chapter of the Association for Computational Linguistics., June.

[Hajishirzi et al.2013] Hannaneh Hajishirzi, Leila Zilles, Daniel S. Weld, and Luke Zettlemoyer. 2013. Joint coreference resolution and named-entity linking with multi-pass sieves. In Empirical Methods in Natural Language Learning.

[Hsieh et al.2008] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S. Sathiya Keerthi, and S. Sundararajan. 2008. A dual coordinate descent method for large-scale linear svm. In International Conference on Machine Learning.

[Kazama and Torisawa2007] Jun’ichi Kazama and Kentaro Torisawa. 2007. Exploiting wikipedia as external knowledge for named entity recognition. In Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

[Kwitkowski et al.2013] Tom Kwiatkowski, Eunsol Choi, Yoav Artzi, and Luke. Zettlemoyer. 2013. Scaling semantic parsers with on-the-fly ontology matching. In Empirical Methods in Natural Language Processing.

[Lao et al.2011] Ni Lao, Tom Mitchell, and William W. Cohen. 2011. Random walk inference and learning in a large scale knowledge base. In Conference on Empirical Methods in Natural Language Processing.

[Lewis and Gale1994] David D. Lewis and William A. Gale. 1994. A sequential algorithm for training text classifiers. In ACM SIGIR Conference on Research and Development in Information Retrieval.

[Ling and Weld2012] Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In Association for the Advancement of Artificial Intelligence.

[Manning et al.2008] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to information retrieval. In Cambridge University Press, Cambridge, UK.

[Mintz et al.2009] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In Association for Computational Linguistics and International Joint Conference on Natural Language Processing.

[Nickel et al.2011] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. 2011. A three-way model for collective learning on multi-relational data. In International Conference on Machine Learning.

[Pantel et al.2012] Patrick Pantel, Thomas Lin, and Michael Gamon. 2012. Mining entity types from query logs via user intent modeling. In Association for Computational Linguistics.

[Ratinov and Roth2012] Lev Ratinov and Dan Roth. 2012. Learning-based multi-sieve co-reference resolution with knowledge. In Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning.

[Riedel et al.2013] Sebastian Riedel, Limin Yao, Andrew McCallum, and Benjamin M. Marlin. 2013. Relation extraction with matrix factorization and universal schemas. In The North American Chapter of the Association for Computational Linguistics.

[Socher et al.2013] Richard Socher, Danqi Chen, Christopher Manning, and Andrew Y. Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In Advances in Neural Information Processing Systems.

[Toral and Munoz2006] Antonio Toral and Rafael Munoz. 2006. A proposal to automatically build
and maintain gazetteers for named entity recognition by using Wikipedia. In *European Chapter of the Association for Computational Linguistics*.

[West et al. 2014] Robert West, Evgeniy Gabrilovich, Kevin Murphy, Shaohua Sun, Rahul Gupta, and Dekang Lin. 2014. Knowledge base completion via search-based question answering. In *Proceedings of the 23rd International Conference on World Wide Web*, pages 515–526. International World Wide Web Conferences Steering Committee.

[Weston et al. 2011] Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *International Joint Conference on Artificial Intelligence*.

[Yao and Durme 2014] Xuchen Yao and Benjamin Van Durme. 2014. Information extraction over structured data: Question answering with freebase. In *Association for Computational Linguistics*.

[Yao et al. 2013] Limin Yao, Sebastian Riedel, and Andrew McCallum. 2013. Universal schema for entity type prediction. In *Proceedings of the 2013 Workshop on Automated Knowledge Base Construction*. 