A Graph-Based Approach for Ontology Population with Named Entities

Wei Shen[1], Jianyong Wang[1], Ping Luo[2], Min Wang[2]
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]HP Labs China, Beijing, China
[1]chen-wei09@mails.tsinghua.edu.cn, jianyong@tsinghua.edu.cn
[2]{ping.luo, min.wang6}@hp.com

ABSTRACT

Automatically populating ontology with named entities extracted from the unstructured text has become a key issue for Semantic Web and knowledge management techniques. This issue naturally consists of two subtasks: (1) for the entity mention whose mapping entity does not exist in the ontology, attach it to the right category in the ontology (i.e., fine-grained named entity classification), and (2) for the entity mention whose mapping entity is contained in the ontology, link it with its mapping real world entity in the ontology (i.e., entity linking). Previous studies only focus on one of the two subtasks and cannot solve this task of populating ontology with named entities integrally. This paper proposes APOLLO, a grAph-based aPproach for pOpuLating ontoLOgy with named entities. APOLLO leverages the rich semantic knowledge embedded in the Wikipedia to resolve this task via random walks on graphs. Meanwhile, APOLLO can be directly applied to either of the two subtasks with minimal revision. We have conducted a thorough experimental study to evaluate the performance of APOLLO. The experimental results show that APOLLO achieves significant accuracy improvement for the task of ontology population with named entities, and outperforms the baseline methods for both subtasks.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Storage and Retrieval—*Information Search and Retrieval*

General Terms

Algorithms, Experimentation

Keywords

Ontology population, Named entity classification, Entity linking, Label propagation

CIKM'12, October 29–November 2, 2012, Maui, HI, USA.
Copyright 2012 ACM 978-1-4503-1156-4/12/10 ...$15.00.

1. INTRODUCTION

The trend to advance the traditional keyword-based search to the semantic entity-based search has attracted a lot of attention in recent years. A critical step to achieve this goal is to construct a comprehensive machine-understanding ontology about the world's entities, their semantic categories and their mutual relationships. Despite there exist some ontologies such as WordNet [11] which are constructed manually, they have limited coverage in various regions. Furthermore, as world evolves, new facts come into existence and are digitally expressed on the Web. Therefore, populating and enriching the existing ontology with the newly extracted facts become more and more important. Manually populating ontology requires substantial human effort and is usually time consuming. This has motivated the research on the automatic ontology population techniques.

The development of the information extraction techniques makes the automatic ontology population techniques possible. Recently, information extraction from large document collections has received a lot of attention, and a variety of information extraction problems have been considered such as named entity recognition [17, 10], named entity classification [12, 21] and relation extraction [3, 25]. Integrating the newly extracted knowledge derived from the information extraction systems with the existing ontology requires to deal with the task of populating ontology with named entities.

Ontology population with named entities is the task to locate the right place of the detected named entity in the ontology. Given a named entity mention detected from the unstructured text, if the mapping entity of the entity mention is not contained in the ontology, we should find the right category node to which the entity mention should be attached in the ontology, which is known as the task of fine-grained named entity classification. Otherwise, if the mapping entity of the entity mention exists in the ontology, the aim of this task is to link this detected entity mention with its corresponding real world entity in the ontology, which is known as the entity linking task. For example, we assume that the ontology contains the entity of NBA player named "Michael Jordan", and there is only one entity named "Michael Jordan" in the ontology. In the text "Professor Michael Jordan has a talk on machine learning.", the mapping entity of the entity mention "Michael Jordan" is the Berkeley professor whose name is also "Michael Jordan". Since the ontology does not contain such an entity, we should perform the fine-grained named entity classification task and obtain the category (i.e., *Professor*) to which the mapping entity of this entity mention belongs. Then we create a new node for this entity mention "Michael Jordan" and attach this node to the category node *Professor* in the ontology. While for the entity mention appearing in the text "Michael Jordan wins NBA champion.", we should map this mention of "Michael Jordan" to the entity of NBA player existing in the ontology, which is called the entity linking task.

Ontology population with named entities has received much attention recently, and several solutions to this task have been proposed in research [15, 14, 28, 6, 8, 7]. However, these state-of-the-art systems only focus on one of the two subtasks (i.e., fine-grained named entity classification and entity linking). It calls for a unified framework to resolve the task of populating ontology with named entities integrally.

The ontology-based fine-grained named entity classification problem has been addressed by many researchers [15, 14, 28, 6, 13]. These systems classify the named entities detected from the text into a large number of categories specified by an ontology or a multi-level taxonomy. In most of these studies, they suppose that the entity disambiguation/linking process has been completed, and all identified entity mentions have been mapped to their unique representations. Besides, some of them consider that the entity mentions identified in the data set are not ambiguous, and thus ignore the ambiguity problem of the named entity. However, in realistic scenario, the mention form of the named entity is highly ambiguous. For example, the entity mention of "Michael Jordan" can refer to the famous basketball player, the computer science professor or some other persons. Henceforth, in APOLLO, we do not ignore the ambiguity problem of the named entity, and resolve the task of populating ontology with named entities integrally.

The solutions proposed in [4, 7, 8, 24] address the entity linking task, and they all aim to link the textual mention form of the named entity with the corresponding real world entity in the existing ontology. If the matching entity of certain entity mention does not exist in the ontology, they just return NIL (denoting an unlinkable entity mention) for this mention form, and cannot attach this unlinkable entity mention to the right category in the ontology.

In this paper, we propose APOLLO, a graph-based weakly supervised framework to resolve the task of automatic ontology population with named entities integrally. Meanwhile, our proposed framework APOLLO can be directly applied to either of the two subtasks with minimal revision. APOLLO is based on the assumption that if the contexts where two named entities appear are semantically similar, they are likely to belong to the same category, which is the extension of the distributional hypothesis [16]. The only training data for APOLLO is an initial ontology, in which there are a list of labeled named entities whose categories are known to us beforehand. Therefore, APOLLO is weakly supervised and needs minimal human involvements. Given each entity mention/named entity and its associated document context, we firstly recognize all the Wikipedia concepts appearing in this context, and we consider the set of these detected Wikipedia concepts as the *semantic signature* of this entity mention/named entity. Then we construct a graph consisting of the nodes coming from all the entity mentions which need to be populated into the ontology, the named entities contained in the ontology, and the Wikipedia concepts existing in their corresponding *semantic signatures.* We weight the edges between the Wikipedia concept nodes in the graph, by leveraging the rich semantic knowledge embedded in the link structure of the Wikipedia articles. The nodes of the named entities contained in the ontology are annotated with their category labels, and other unlabeled entity mention nodes are required to be classified. Subsequently, the Adsorption label propagation algorithm [2] is applied to this constructed graph to produce a probability distribution over categories for each unlabeled entity mention node, based on the rich graph structure. Finally, for each entity mention, we have to validate whether there exists a named entity in the ontology we could link this entity mention with. Otherwise, we attach this entity mention to the category that has the largest distribution. It is noted that a very preliminary two-page version of this paper [22] has been published in WWW'12. In this paper, we make further enhancements, and give a complete and in-depth description of our proposed APOLLO framework.

To summarize, we make the following contributions.

- We propose APOLLO, a novel graph-based unified framework which leverages the rich semantic information derived from Wikipedia to deal with the task of ontology population with named entities integrally. Previous studies only focus on one of the two subtasks. Moreover, APOLLO can be directly applied to either of the two subtasks with minimal revision.
- APOLLO is a weakly supervised framework that requires minimal human involvements. Moreover, APOLLO is open-domain as it is independent of the underlying ontology.
- To validate the effectiveness of APOLLO, we conducted a thorough experimental study, and the experimental results demonstrate that APOLLO achieves a significant improvement in accuracy for the task of ontology population with named entities.
- We extensively evaluated the performance of APOLLO over both subtasks, and the experimental results show that APOLLO outperforms the baseline methods for both subtasks.

The remainder of this paper is organized as follows. Section 2 formulates the problem and presents the APOLLO framework. Next, the three modules of APOLLO (i.e., Graph Creation, Label Propagation and Linking Validation) are respectively introduced in Section 3, Section 4 and Section 5. Section 6 presents our experiments and Section 7 discusses the related work. We conclude this paper in Section 8.

2. ONTOLOGY POPULATION WITH NAMED ENTITIES

In this section, we will study the problem of automatically populating ontology with named entities extracted from the large text corpus. For this purpose, we will firstly give some notations and formulate the problem of ontology population with named entities in Section 2.1. Subsequently, the overall framework of APOLLO will be introduced in Section 2.2.

2.1 Notations and Problem Formulation

The only input of our framework APOLLO is a collection of documents and an initial ontology. Let D be the collection of the input documents and Ω be the initial ontology.

Let ζ be the set of all entity mentions recognized from the document set D, and each entity mention $s \in \zeta$ needs to be populated into the ontology Ω. Suppose that there are a list of labeled named entities whose categories are known within the initial ontology Ω. Let N_Ω denote the set of all named entities contained in the ontology Ω, and C_Ω be the set of all categories in the taxonomy of Ω.

Entity mention and mapping entity: An entity mention $s \in \zeta$ is a token sequence in the text document which refers to some named entity. Let n_s denote the corresponding real world named entity the entity mention s refers to. We should differentiate between the entity itself and its various entity mentions. In reality, an entity may have multiple entity mentions. For example, the entity *Hewlett-Packard* has its abbreviation "HP". On the contrary, one entity mention may also refer to several different real world entities. For instance, the entity mention of "Michael Jordan" can refer to the famous basketball player, the computer science professor or some other persons. Henceforth, the mapping entity n_s of the entity mention s depends on the context where the entity mention s occurs.

Document context: We define the document context η_s of the entity mention $s \in \zeta$ as a window of words around the occurrence of the entity mention s. Assume that the entity mention s of length $|s|$ words appears in a document d at position p. The size-k document context η_s of entity mention s with respect to d is the window $w_{p-k}, \ldots, w_{p-1}, w_{p+|s|}, \ldots, w_{p+|s|+k-1}$ of words around the occurrence of s (w_i represents the word at position i). For instance, the entity mention of "Michael Jordan" occurs in a document containing such a sentence, "In the NBA Final of 1991, Michael Jordan shot 12 free throws." When the size k is set to 5, the size-k document context η_s is "the NBA Final of 1991 shot 12 free throws". On the other hand, for each named entity $n \in N_\Omega$, we define the document context η_n of the named entity n as the description context for n in the ontology. As both the entity mention $s \in \zeta$ and the named entity $n \in N_\Omega$ have document contexts, we use η to denote the document context corresponding to an entity mention or a named entity.

Semantic signature: To capture the semantic information existing in the document context η, we recognize all the Wikipedia concepts γ appearing in η, and consider the set of these detected Wikipedia concepts as the *semantic signature* δ. Here, Wikipedia concept means the concept which has its corresponding descriptive article in Wikipedia, and each Wikipedia concept is represented by the title of its Wikipedia article. For the general textual document, we utilize the open source toolkit Wikipedia-Miner[1] to detect the Wikipedia concepts appearing in the context. The Wikipedia-Miner toolkit takes the general unstructured text as input and uses the machine learning approach to detect the Wikipedia concepts in the input document [20]. For the document context in the example mentioned above, this Wikipedia-Miner toolkit returns two Wikipedia concepts, i.e., *NBA Final* and *Free throw*. Therefore, it can be seen that these detected Wikipedia concepts are highly semantically related to the NBA player *Michael Jordan*, and we can leverage this semantic information contained in this *semantic signature* to populate this entity mention "Michael Jordan" into the ontology Ω effectively. As we know, the document from the Wikipedia has its special layout to organize its content, i.e., *Wiki markup*[2]. The references to other Wikipedia concepts in the Wikipedia document are within pairs of double square brackets. Henceforth, for a Wikipedia document, we can identify the Wikipedia concepts appearing in it directly and accurately by leveraging the characteristic of *Wiki markup*.

Now we can formulate the problem of ontology population with named entities.

Ontology population with named entities: *Given a collection of documents D, an initial ontology Ω and a set of entity mentions ζ detected from D, the task of ontology population with named entities is to locate the right place for each entity mention $s \in \zeta$ in the ontology Ω. For each $s \in \zeta$, if the mapping entity $n_s \notin N_\Omega$, the entity mention s has to be attached to the proper category $c_s \in C_\Omega$; If the mapping entity $n_s \in N_\Omega$, the goal of this task is to return this mapping entity n_s.*

2.2 The APOLLO framework

Based on the problem definition, we propose a framework called APOLLO, to address the task of ontology population with named entities using three modules as follows:

- **Graph Creation**
  To represent all the available information about the relationships between the entity mentions ζ and the named entities N_Ω in a unified way, this module constructs a graph G consisting of the nodes which come from all the entity mentions ζ, the named entities N_Ω and the Wikipedia concepts in their *semantic signatures* δ. Meanwhile, this module embeds the rich semantic information derived from the link structure of the Wikipedia articles into the graph, in the form of weighting the edges between these Wikipedia concept nodes.

- **Label Propagation**
  In this module, we assign each entity mention $s \in \zeta$ to the proper category $c_s \in C_\Omega$ via graph label propagation. Each named entity node $n \in N_\Omega$ is annotated with its corresponding category label in the graph G, and other unlabeled entity mention nodes $s \in \zeta$ are required to be classified. We then present the Adsorption label propagation algorithm [2], which is applied to the graph G to ultimately produce the predicted category $c_s \in C_\Omega$ for each unlabeled entity mention node $s \in \zeta$, based on the rich graph structure.

- **Linking Validation**
  If the mapping entity n_s of the entity mention $s \in \zeta$ exists in the ontology Ω, we have to link this entity mention s with its mapping entity n_s, i.e., the entity linking task. Otherwise, we return c_s as the category for entity mention s. Henceforth, we add this module to validate whether its mapping entity $n_s \in N_\Omega$.

In the following sections, we will introduce those three modules in details.

3. GRAPH CREATION

To represent all the available information in a unified form, we need a representation capable of encoding efficiently all the complicated relationships between the entity

[1] http://wikipedia-miner.cms.waikato.ac.nz/

[2] http://en.wikipedia.org/wiki/Wiki_markup

mentions ζ and the named entities N_Ω. To achieve this goal, we select the graph as the representation, since the graph can encode different types of objects (i.e., entity mentions, named entities and Wikipedia concepts) as the nodes in the graph, and represent various relationships between these objects as the edges between these nodes. Furthermore, the graph makes the potential label propagation paths explicit, and the label information can be propagated along these connecting paths from the labeled named entity nodes to the unlabeled entity mention nodes. For example, if the *semantic signatures* of the entity mention “Michael Jordan” and the named entity *Yao Ming* both contain the Wikipedia concept *NBA Final*, then this can be treated as an evidence that this entity mention “Michael Jordan” may have the same category as the named entity *Yao Ming*. Henceforth, the path connecting the two nodes (i.e., “Michael Jordan” and *Yao Ming*) via the Wikipedia concept node (i.e., *NBA Final*) may help to forward the label information of the labeled named entity node *Yao Ming* to the unlabeled entity mention node “Michael Jordan”.

Specifically, we construct a single graph $G = (V, E, W)$ to represent all the information available for this task, where V denotes the set of nodes, E is the set of edges and $W : E \rightarrow \mathbf{R}$ is the weight function which gives positive weight for each edge in E. It is noted that we define the graph G as an undirected graph. The node set V consists of the nodes which come from all the entity mentions ζ, the named entities N_Ω and the Wikipedia concepts in their *semantic signatures* δ. Specifically, for each entity mention $s \in \zeta$, we pair it with each Wikipedia concept $\gamma \in \delta_s$ where δ_s denotes the *semantic signature* of s, to create the triple (s, γ, w), and the weight w could be $1/dist(s, \gamma)$ where $dist(s, \gamma)$ is the distance between the positions of s and γ in the context. In the experiment, we set this weight w to 1.0 for the purpose of simplicity. For each triple (s, γ, w), s and γ are added to V and the edge (s, γ) is added to E, with $W(s, \gamma) = w$. And for each named entity $n \in N_\Omega$, we also pair it with each Wikipedia concept $\gamma \in \delta_n$ where δ_n denotes the *semantic signature* of n, to create the triple (n, γ, w), where the weight w could be the degree of importance for γ in the description context of entity n. The degree of importance for γ could be calculated as its average semantic relatedness to all other Wikipedia concepts in δ_n. However, we set this weight w to 1.0 in the experiment for simplicity as well. For each triple (n, γ, w), n and γ are added to V and the edge (n, γ) is added to E, with $W(n, \gamma) = w$. After the triple (s, γ, w) for each entity mention $s \in \zeta$ and the triple (n, γ, w) for each named entity $n \in N_\Omega$ are all added into the graph G, two nodes of the entity mention or the named entity in the graph G are just connected via the Wikipedia concept nodes which co-occur in both of their *semantic signatures*. Therefore, we define the current status of the graph G as G_{co}, denoting that this graph just contains the co-occurrence information between the entity mentions or named entities and the Wikipedia concepts in their *semantic signatures*.

To forward the label information over the graph more effectively, the semantically related Wikipedia concept nodes should be connected by some edges to enrich the information propagation paths. For instance, if the *semantic signature* of the entity mention “Michael Jordan” contains the Wikipedia concept *Free throw*, and the *semantic signature* of the named entity *Yao Ming* contains the Wikipedia concept *Technical foul*, this entity mention “Michael Jordan” is expected to have the same category as the named entity *Yao Ming*, since the two Wikipedia concepts are highly semantically related. Thus, we incorporate the semantic relatedness between the Wikipedia concepts into the graph.

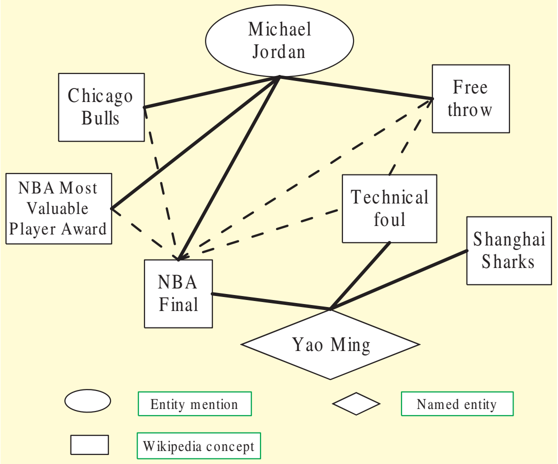

Figure 1: An example of the created graph

Since the link structure of the Wikipedia articles expresses the rich semantic relations, two Wikipedia concepts are considered to be semantically related if there are many Wikipedia concepts that link to both. In order to measure the strength of the semantic relatedness, we adopt the Wikipedia Link-based Measure (WLM) described in [19] to calculate the semantic relatedness between Wikipedia concepts. The WLM modeled from the Normalized Google Distance [5] is based on the Wikipedia's hyperlink structure. Given two Wikipedia concepts u_1 and u_2, we define the semantic relatedness between them as follows:

$$SR(u_1, u_2) = 1 - \frac{log(max(|U_1|, |U_2|)) - log(|U_1 \bigcap U_2|)}{log(|WP|) - log(min(|U_1|, |U_2|))} \quad (1)$$

where U_1 and U_2 are the sets of Wikipedia concepts that link to u_1 and u_2 respectively, and WP is the set of all concepts in Wikipedia. This definition gives higher value to more related concept pair and the value of $SR(u_1, u_2)$ is varied from 0.0 to 1.0. For each pair of Wikipedia concept nodes (γ_1, γ_2) in the graph, if the semantic relatedness $SR(\gamma_1, \gamma_2)$ is greater than some threshold τ, we add an edge (γ_1, γ_2) to E, with $W(\gamma_1, \gamma_2) = SR(\gamma_1, \gamma_2)$.

Figure 1 shows an example of the created graph, in which there are one entity mention node (i.e., “Michael Jordan”) and one named entity node (i.e., *Yao Ming*). We assume that the *semantic signature* of the entity mention “Michael Jordan” has four Wikipedia concepts (i.e., *NBA Most Valuable Player Award*, *NBA Final*, *Chicago Bulls* and *Free throw*), while the *semantic signature* of the named entity *Yao Ming* has three Wikipedia concepts (i.e, *NBA Final*, *Technical foul* and *Shanghai Sharks*). In Figure 1, we add a real line between each entity mention/named entity node and each Wikipedia concept node in its *semantic signature*. From Figure 1, we can see that the Wikipedia concept node *NBA Final* is connected with both the entity mention node “Michael Jordan” and the named entity node *Yao Ming*. The dash lines added between the semantically related Wikipedia concept nodes in Figure 1 make the paths connecting the entity mention node “Michael Jordan” and the named entity node *Yao Ming* more abundant, which also demonstrate that the entity mention "Michael Jordan" is likely to have the same category as the named entity *Yao Ming*.

It is noted that in our framework, the semantic relatedness can be computed by other methods such as the type hierarchy based similarity and distributional context similarity introduced in [23] or combination of them, which gives flexibility to APOLLO in an efficient and simple way.

4. LABEL PROPAGATION

The aim of this section is to assign each entity mention $s \in \zeta$ to the proper category $c_s \in C_\Omega$. Firstly, we annotate each named entity node $n \in N_\Omega$ with its corresponding category label in the graph G. In this paper, the named entity category is used as the label for the node, and we assume that each named entity just belongs to one category for the purpose of simplicity. The remaining question is how to propagate the category labels present on the labeled named entity nodes to the unlabeled entity mention nodes in the graph. To solve this problem, we apply the Adsorption label propagation algorithm introduced in [2] to the graph G, to produce the predicted category $c_s \in C_\Omega$ for each unlabeled entity mention node $s \in \zeta$ based on the rich graph structure. Furthermore, the Adsorption algorithm supports incremental updates and can be easily parallelized, which are important for large scale ontology population task. The Adsorption algorithm works on the graph G, and ultimately produces for each unlabeled entity mention node $s \in \zeta$ a label distribution L_s, representing which category labels are appropriate for the unlabeled entity mention s. Our framework APOLLO assumes that named entities that occur in semantically similar contexts belong to the same category. Specifically, we consider that named entities that co-occur with semantically related Wikipedia concepts may have the same category. Therefore, the label propagation algorithm is to forward the category label between the related named entity nodes and entity mention nodes.

The Adsorption algorithm has three different but equivalent interpretations, whose details are introduced in [2]. However, in this paper, we use two interpretations to classify the unlabeled entity mention into the proper category.

Adsorption via Averaging: In this view of the algorithm, the labels are propagated from one node to all its neighbors. Thus each node in the graph has two roles, forwarding labels and collecting labels, and each node keeps track of the history of all labels it receives. For the sake of presentation, we preprocess the original graph G to generate the augmented graph $G' = (V', E', W')$ in the way that, for each labeled named entity node $n \in N_\Omega$, we create a "shadow" node $\tilde{n}$ which has just one neighbor n in G', with an edge $(\tilde{n}, n)$ connecting them with $W'(\tilde{n}, n) = 1$. Let $\tilde{N_\Omega}$ denote the set of "shadow" nodes, $\tilde{N_\Omega} = \{\tilde{n} | n \in N_\Omega\}$, $\tilde{N_\Omega} \subset V'$. Thus, $V' = V \bigcup \tilde{N_\Omega}$, $E' = E \bigcup \{(\tilde{n}, n) | n \in N_\Omega\}$ and $W'(\tilde{n}, n) = 1$ for $n \in N_\Omega$, $W'(v_1, v_2) = W(v_1, v_2)$ for $v_1, v_2 \in V$. Meanwhile, we give the label distribution L_n of each $n \in N_\Omega$ to its "shadow" node $\tilde{n}$ in G', and leave n in graph G' with no label distribution. We define ϕ as the label which represents lack of information about the actual labels. Then, at the beginning of the algorithm, we define the initial label distribution I_v for all $v \in V'$. Specifically, for each "shadow" node $\tilde{n} \in \tilde{N_\Omega}$, $I_{\tilde{n}} = L_n$, and for all other nodes $v \in V', v \notin \tilde{N_\Omega}$, $I_v = L^\phi$ where L^ϕ represents that we have no information about the label distribution of the node v. Subsequently, the algorithm proceeds as follows: for each node $v \in V'$, we compute the label distribution as the weighted average of the label distributions of all its neighbors, i.e., $L_v = \sum_u W'(u, v) L_u$.

Adsorption via Random Walks: This view takes random walks over the edge-reversed version of the graph G' to find the label distribution for each node, which has been proved to be equivalent with the Averaging view, as described in [2]. As the graph G' is undirected, its edge-reversed version of G' is the same as itself. Therefore, to estimate the label distribution L_v for each node $v \in V'$, we perform a random walk on graph G' starting from node v. When the random walk reaches a node t, there are three choices: (a) continue the random walk to the neighbors of t; (b) abandon the random walk; (c) stop the random walk and inject the initial label distribution I_t. We assume the probabilities of these three events are $P_c(t)$, $P_a(t)$ and $P_i(t)$ respectively. Finally, L_v is set to be the expectation of all labels injected from random walks starting from node v.

Algorithm 1 *Adsorption Algorithm*

Input: $G' = (V', E', W')$, $\{I_v | v \in V'\}$.
Output: $\{L_v | v \in V'\}$.

```
1: for all $v \in V'$ do
2:     $L_v = I_v$
3: end for
4: repeat
5: for all $v \in V'$ do
6:     $M_v = \Sigma_u W'(u, v) L_u$
7: end for
8: Normalize $M_v$ to have unit $M_1$ norm
9: for all $v \in V'$ do
10:     $L_v = P_c(v) * M_v + P_i(v) * I_v + P_a(v) * L^\phi$
11: end for
12: until convergence
```

We combine these two interpretations of the Adsorption algorithm to generate the label distribution L_v for each node $v \in V'$ with Algorithm 1 like [27]. Algorithm 1 firstly initializes the label distributions for all nodes in the graph (line 1-line 3). Then, for each node $v \in V'$, the algorithm iteratively computes the weighted average of the label distributions of all its neighbors (line 5-line 7), and normalizes the computed label distribution M_v to have unit norm (line 8). Next, we use the random walk probabilities to estimate the new label distribution L_v for each node $v \in V'$ (line 9-line 11). Until convergence, each node $v \in V'$ carries a label distribution and outputs L_v as the final results. Convergence occurs if the label distributions of all nodes do not change in a round. However, in practice, we run the algorithm for a fixed number of iterations alternatively. In Algorithm 1, via using the variable M_v in line 10, we compute the label distribution for node v in the i^{th} iteration entirely based on its neighbors' label distributions from the $(i-1)^{th}$ iteration. Therefore, Algorithm 1 has the memoryless property and can be easily parallelized, which is beneficial for large scale ontology population task.

To set the random walk probabilities, we used the following heuristics from [27]. Let $c_v = \frac{log\beta}{log(\beta + exp(H(v)))}$, where $H(v) = -\Sigma_u p_{uv} * log(p_{uv})$ with $p_{uv} = \frac{W'(u,v)}{\sum_{u'} W'(u',v)}$. There-

Table 1: A part of the dictionary DT

K (Surface form)	$K.value$ (Mapping entity)
Yao Ming	*Yao Ming*
Microsoft Corporation	*Microsoft*
Michael Jordan	*Michael Jordan* *Michael I. Jordan* *Michael Jordan (mycologist)* *Michael Jordan (footballer)* ...
Kobe Bryant	*Kobe Bryant*

fore, if node v has many neighbors, c_v is low. In the experiment, we set $\beta = 2$. If node $v \in N_\Omega$, we set $i_v = (1 - c_v) * \sqrt{H(v)}$; otherwise, $i_v = 0$. Then let $z_v = max(c_v + i_v, 1)$. Lastly, we computed the random walk probabilities for each node $v \in V'$ as follows:

$$P_c(v) = c_v / z_v \tag{2}$$

$$P_i(v) = i_v / z_v \tag{3}$$

$$P_a(v) = 1 - P_c(v) - P_i(v) \tag{4}$$

According to Formula 2, we can see that for the high-degree node, the continue probability $P_c(v)$ is low. Thus we can decrease the probability of the random walk running into the unrelated regions in the graph, and make the random walk stay relatively close to its source node.

5. LINKING VALIDATION

For each entity mention $s \in \zeta$, we obtain the label distribution L_s over the categories C_Ω in the Label Propagation module. We consider the category which has the largest distribution in L_s as the predicted category c_s for the entity mention s. According to the task definition of ontology population with named entities, if the mapping entity n_s of the entity mention s exists in the ontology Ω, we have to link this entity mention s with its mapping entity n_s. Henceforth, we add this module to validate whether its mapping entity $n_s \in N_\Omega$.

As stated in Section 2.1, one entity mention may refer to several different real world entities. Thus, given an entity mention s, we firstly retrieve the set of entities that may be referred by this entity mention s, and we denote this set of entities as the candidate entity set CN_s for s. Intuitively, the candidate entities in CN_s should have the name of the entity mention of s. To solve this problem, we need to build a dictionary DT that contains vast amount of information about various mention forms of the named entities, like name variations, abbreviations, confusable names, spelling variations, nicknames, etc. In our paper, the dictionary DT is a <key, value> mapping, where the column of the key K is a list of entity mentions and the column of the mapping value $K.value$ is the set of named entities which are referred by the key K. We construct the dictionary DT by leveraging the following four structures of Wikipedia: **Entity page**, **Redirect page**, **Disambiguation page** and **Hyperlink in Wikipedia article**. The detailed construction method is introduced in [24, 23]. A part of the dictionary DT is shown in Table 1.

For each entity mention $s \in \zeta$, we look up the dictionary DT and search for s in the column of the key K. If a hit is found, i.e., $s \in K$, we add the set of the mapping entities $s.value$ to the candidate entity set CN_s. Suppose that any two entities belonging to the same category do not have the same name. For example, there is only one entity named "Michael Jordan" belonging to the category of NBA basketball player. Therefore, if two entity instances having the same name belong to the same category, we can predict that these two entity instances are the instances of the same entity. Thus, if there exists some entity $n \in CN_s$ whose category is also c_s, the same category as the predicted category for the entity mention s, then we can predict that this entity n is the mapping entity n_s of the entity mention s, and we should link this entity mention s with this entity n; otherwise, we can predict that the mapping entity of the entity mention s does not exist in the ontology Ω, that is to say, $n_s \notin N_\Omega$.

6. EXPERIMENTS

To evaluate the effectiveness of APOLLO, we conducted a thorough experimental study in this section. Firstly, we tested APOLLO over both of the two subtasks, i.e., fine-grained named entity classification task (described in Section 6.1) and the entity linking task (introduced in Section 6.2), respectively. Subsequently, we demonstrate the experimental results of APOLLO for the task of ontology population with named entities in Section 6.3.

6.1 Fine-grained named entity classification task

6.1.1 Experimental setting

To the best of our knowledge, there is no publicly available data set for the fine-grained named entity classification task, and the page for accessing the data set provided in [9] is also unavailable. Thus, we constructed the data set for the fine-grained named entity classification task, from the May 2011 version of Wikipedia and YAGO(1)[3] of version 2009-w10-5. To generate the training and test data, we chose 20 categories which are the subclasses of the *person* category from YAGO. Since the numbers of the instances belonging to different categories vary much, we randomly selected at most 200 instances for each selected category by querying the YAGO ontology, and the created data set DS_{NEC} consists of 3304 distinct instances belonging to the 20 categories in total. Since person names are more ambiguous and the categories of person entities are more diverse, the person name classification is much more challenging. Moreover, the experiments of previous methods [12, 28, 15, 14, 13] are all carried out with person names for the fine-grained named entity classification task.

We compared our framework APOLLO with the classification based approach proposed in [13], which significantly outperforms the single-context rule-based extractor similar to several state-of-the-art techniques for the task of fine-grained named entity classification [13]. We refer to this baseline method as *Ganti-KDD*. The approach *Ganti-KDD* considers two types of features, which are text n-gram feature and the list-membership feature. Since these features are all extracted from the **multi-context**, the union of all contexts across multiple documents within which the entity occurs, they assume that each entity identified in the corpus has been converted to its *canonical* representation. However, in the real application, the general documents corpus cannot satisfy this assumption.

[3] http://www.mpi-inf.mpg.de/yago-naga/yago/

Table 2: Experimental results over the DS_{NEC} data set

Approach \ ρ	0.5		0.6		0.7		0.8		0.9	
	MRR	Accu.	MRR	Accu.	MRR	Accu.	MRR	Accu.	MRR	Accu.
Ganti-KDD	-	0.7036	-	0.6989	-	0.7337	-	0.7247	-	0.7112
APOLLO$_{CO}$	0.7771	0.6915	0.7901	0.7049	0.8135	0.7409	0.7914	0.7201	0.8412	0.7750
APOLLO$_{SR}$	**0.8059**	**0.7223**	**0.8184**	**0.7366**	**0.8399**	**0.7703**	**0.8225**	**0.7489**	**0.8535**	**0.7903**

According to the original experimental setting in [13], we used the support vector machine model, *libsvm*[4], as the underlying classifier. To generate the multi-context features, we employed 3.5 million Wikipedia pages as the document corpus, where each entity in the *Wikitext* has been converted to its *canonical* representation, and this characteristic of the Wikipedia document corpus satisfies the assumption of *Ganti-KDD*. To compute the n-gram features, we extracted all n-grams in the size-4 *document context* for each occurrence of an entity, using the presence/absence of the 10K most frequent n-grams among them as features [13]. For the list-membership features, we used the entire document an entity occurs in as the aggregate context of this entity, using a 10% sample of the entities in the training data for each category as the list corpus [13].

To evaluate the performance of APOLLO over the fine-grained named entity classification task, we just eliminated the Linking Validation module in APOLLO, and let the Label Propagation module output the final label distribution for each test entity. To generate the *semantic signature* for each entity in DS_{NEC}, we used its corresponding entire entity page in Wikipedia as the document context. In the experiment, we set the edge weight between a named entity and each Wikipedia concept in its *semantic signature* to 1.0 for the purpose of simplicity. In this experiment, the threshold τ is experimentally set to 0.34, which yields the best performance. The created graph G_{co} contains 82,833 nodes and 155,450 edges, and after we added edges between the semantically related Wikipedia concept nodes into the graph, the final graph G consists of about 2.5 million edges. We refer to the results of our framework APOLLO applied to the graph G_{co} as APOLLO$_{CO}$, and denote the results of APOLLO applied to G as APOLLO$_{SR}$. In the Label Propagation module, the number of iterations for the Adsorption algorithm was set to 10.

6.1.2 Experimental results

In this subsection, we present the evaluation results of APOLLO for the fine-grained named entity classification task. As the output of our framework is a label distribution, we computed the Mean Reciprocal Rank (MRR) of the test entity with respect to the gold standard target category. In addition, since the baseline method *Ganti-KDD* employs the SVM model which produces the predicted category for each test entity, we used the usual metric of *Accuracy* (Accu.) on the classification results to evaluate the performance of *Ganti-KDD*. Meanwhile, to give a fair comparison, the accuracy of the predicted category which has the largest distribution in the label distribution of each test entity is also computed for APOLLO. To demonstrate the performance of these approaches with different numbers of training entities, we made the parameter ρ denote the proportion of the training entities in the data set DS_{NEC}. To split the data set DS_{NEC} into the training and test data set with respect to ρ, for each selected category in DS_{NEC}, we randomly selected the corresponding number of entities belonging to this category as the training entities, and the remaining entities are regarded as the test entities.

Table 2 shows the experimental results of these approaches over the DS_{NEC} data set under the different settings of the parameter ρ, varying from 0.5 to 0.9. From the results, we can see that the baseline method *Ganti-KDD* and the approach APOLLO$_{CO}$ have the similar accuracy when ρ is set from 0.5 to 0.8. But when ρ equals to 0.9, the accuracy achieved by APOLLO$_{CO}$ is much higher than what can be achieved by *Ganti-KDD*. However, it is noticed that the features of the baseline method *Ganti-KDD* are all extracted from the **multi-context**, the union of all contexts across multiple documents within which the entity occurs, while the approach APOLLO$_{CO}$ only leverages the co-occurrence information of the Wikipedia concepts in the *semantic signature* extracted from the **single context**. When the features of the baseline method *Ganti-KDD* are extracted from the **single context**, the accuracy achieved by *Ganti-KDD* decreases greatly, which will be confirmed in Section 6.3.2. By leveraging the semantic knowledge embedded in Wikipedia, the approach APOLLO$_{SR}$ significantly outperforms the baseline method *Ganti-KDD* in terms of accuracy and the approach APOLLO$_{CO}$ in terms of both MRR and accuracy. Overall, the experimental results indicate that the Wikipedia concepts extracted from the document context and the semantic relations between them are quite useful for the task of fine-grained named entity classification.

The detailed results for each category of these three approaches are shown in Table 3 when the training entity proportion ρ equals to 0.8. It can be seen from the results in Table 3 that some of these categories are relatively easy to distinguish (e.g., Presidents of the United States and Chinese emperors), while some categories (e.g., American socialists and American revolutionaries) are very difficult to be classified accurately. In Table 3, for each row (category), the best accuracy is in bold, and the results show that APOLLO$_{SR}$ obtains the highest accuracy for 12 categories, while the approaches *Ganti-KDD* and APOLLO$_{CO}$ get the highest accuracy for 7 and 8 categories respectively.

6.2 Entity linking task

6.2.1 Experimental setting

Entity linking is initiated as a task in the track of Knowledge Base Population (KBP) at the Text Analysis Conference (TAC). The data set for TAC-KBP track in 2009[5] is available for us, so we used it as the test data set for APOLLO over the entity linking task. According to the problem formulation of ontology population with named en-

[4] http://www.csie.ntu.edu.tw/∼cjlin/libsvm/

[5] http://apl.jhu.edu/∼paulmac/kbp.html

Table 3: Accuracy for each category over the DS_{NEC} data set ($\rho = 0.8$)

Category	*Ganti-KDD*	APOLLO$_{CO}$	APOLLO$_{SR}$
American chief executives	0.5263	0.4211	**0.5789**
American entrepreneurs	**0.625**	0.4474	0.5405
Presidents of the United States	**1.0**	**1.0**	**1.0**
American singers	0.7027	0.7027	**0.7179**
American film actors	0.8837	0.9070	**0.9268**
American basketball coaches	**0.9**	**0.9**	0.875
American labor leaders	0.6842	0.7949	**0.8421**
England international footballers	0.9	**0.975**	0.95
Harvard Law School alumni	0.45	0.7	**0.825**
American computer scientists	0.7027	0.825	**0.9268**
Olympic athletes of the United States	0.725	0.7	**0.85**
English poets	0.775	0.825	**0.85**
American philosophers	0.8780	**0.9512**	0.875
American academics	**0.7045**	0.2439	0.2381
American socialists	**0.3333**	0.3125	0.2353
American revolutionaries	0.125	**0.25**	**0.25**
English Formula One drivers	**0.9231**	**0.9231**	**0.9231**
American diplomats	**0.6579**	0.5385	0.5385
American television journalists	0.8	**0.85**	0.8
Chinese emperors	0.7143	**1.0**	**1.0**

tities, our entity linking subtask focuses on the entity mention whose mapping entity exists in the ontology, which is called the linkable entity mention. In this TAC-KBP data set, there are total 1675 linkable entity mentions, which require to be linked with the ontology.

To perform the entity linking task, APOLLO firstly produces the candidate entity set CN_s for each entity mention s using the dictionary DT introduced in Section 5. To generate the *semantic signature* for each entity mention in the TAC-KBP data set, we used the entire document where the entity mention occurs as the document context. Next, for each entity mention s, APOLLO creates a graph consisting of the nodes coming from the entity mention s itself, the Wikiepdia concepts in its *semantic signature* δ_s and the candidate entities in CN_s. In the experiment, we also set the edge weight between the entity mention s and each Wikipedia concept in δ_s to 1.0. Since all candidate entities in CN_s are Wikipedia concepts, so we computed the semantic relatedness between each candidate entity in CN_s and each Wikipedia concept in δ_s according to Formula 1. If the semantic relatedness is greater than the threshold τ, APOLLO creates an edge between them and weights the edge using the relatedness measure. In this experiment, the threshold τ is experimentally set to 0.086, which yields the best performance. Next, APOLLO annotates each candidate entity node in the graph with a unique label, and after running the Adsorption algorithm, the candidate entity whose corresponding label has the largest distribution in L_s is regarded as the mapping entity for the entity mention s. In addition, due to many spelling errors existing in the set of entity mentions, we also tried to correct them using the query spelling correction supplied by Google.

6.2.2 Experimental results

Table 4: Experimental results over the TAC-KBP data set compared with top 4 ranked systems in TAC-KBP track of 2009

System	Accuracy	# of correctly linked
Rank 1	0.7725	1294
Rank 2	0.7654	1282
Rank 3	0.7588	1271
Rank 4	0.7063	1183
APOLLO	**0.7845**	**1314**

To evaluate the performance of APOLLO over the TAC-KBP data set, we adopted the evaluation measure *Accuracy*, which is used in most work about entity linking. The accuracy is calculated as the number of correctly linked entity mentions divided by the total number of all entity mentions. The experimental results of APOLLO over the TAC-KBP data set are shown in Table 4. The results of the top 4 systems which perform best over the set of linkable entity mentions in TAC-KBP track of 2009 [18] are also shown in Table 4, for the purpose of comparison. Besides the accuracy, we also show the number of correctly linked entity mentions. The results in Table 4 show that APOLLO outperforms the best systems in TAC-KBP track of 2009, which demonstrates the effectiveness of APOLLO over the entity linking task.

6.3 Ontology population with named entities task

6.3.1 Experimental setting

To evaluate the performance of APOLLO over the task of ontology population with named entities, the test data set should contain both unlinkable entity mention that requires to be attached to the proper category, and linkable entity mention that should be linked with the entity existing in the ontology. We denote the DS_{NEC} data set used in Section 6.1.1 when the parameter ρ equals to 0.8 as $DS_{NEC\rho=0.8}$, which consists of 2643 training entities and 661 test entities. In the following experiments, we regard the training entities in the $DS_{NEC\rho=0.8}$ data set as the set of named entities contained in the ontology. In addition, besides the test entity mentions in $DS_{NEC\rho=0.8}$ which are all unlinkable, we added some new test entity mentions which can be linked with the named entities existing in the ontology. To make the newly added test entity mentions linkable, we randomly sampled 20% of the named entities for each category existing in the ontology, and regarded the names of these sampled entities as the set of newly added test entity mentions. For each newly added test entity mention, we obtained its document context via querying its name with Google. The top ranked page which is not from Wikipedia is regarded as its candidate document context, because the corresponding Wikipedia page has been regarded as the document context for its corresponding mapping entity in the ontology. Then we verified whether the entity described in the candidate document context is the same as the corresponding mapping entity by human judgments. If so, we added this test entity mention and its candidate document context to the test data set; Otherwise, we removed it. Finally, we obtained the data set for the ontology population task (which we refer to DS_{OP}), in which the set of named entities contained in the ontology is the same as the training data set of $DS_{NEC\rho=0.8}$, and the test data set consists of 661 unlinkable test entity mentions from the original test data set of $DS_{NEC\rho=0.8}$ and the newly added 372 linkable test entity mentions.

We added the Linking Validation module of APOLLO to the baseline method *Ganti-KDD* introduced in Section 6.1.1

to create the baseline method BASELINE$_{OP}$ for the ontology population task. The features for the named entities existing in the original $DS_{NEC\rho=0.8}$ data set are extracted from the multi-context in the same way as in Section 6.1.1. Whereas for the newly added linkable test entity mention whose document context is not *Wikitext*, we are only able to extract the features existing in this single context, rather than across multiple documents, since we cannot obtain the *canonical* representation of the entity mention. It is fairly common for one of the mention forms of an entity in a document to be a long and typical mention form of that entity (e.g., "Michael Jordan"), while the other mention forms of the same entity are shorter mention forms (e.g., "Jordan"). To generate richer features for each linkable test entity mention for the BASELINE$_{OP}$ method, we used a simple in-document coreference resolution method which is to map shorter mention forms to the long and typical entity mention form in the same document.

We applied APOLLO to the DS_{OP} data set to evaluate the performance of APOLLO for the task of ontology population with named entities. We used the same setting for APOLLO as described in Section 6.1.1. The final graph G contains 99,609 nodes and about 3.1 million edges.

6.3.2 Experimental results

To evaluate the performance of APOLLO and BASELINE$_{OP}$ over the DS_{OP} data set, we also adopted the evaluation measure *Accuracy* (Accu.), which is used in both of the two subtasks. For the unlinkable test entity mention, if it is attached to the gold standard category, it is regarded as correct. And for the linkable test entity mention, if it is linked with the correct entity, we consider it as correct. The overall accuracy is calculated as the number of all correctly assigned entity mentions divided by the total number of the entity mentions.

The experimental results of APOLLO and BASELINE$_{OP}$ over the DS_{OP} data set are shown in Table 5. We show the accuracy and the number of correctly assigned entity mentions for both APOLLO and BASELINE$_{OP}$, according to the different types of the test entity mentions (i.e., all, unlinkable and linkable). From the results in Table 5 we can see that APOLLO achieves significantly higher accuracy compared with the baseline method BASELINE$_{OP}$ in all aspects. For the set of linkable test entity mentions, the accuracy achieved by BASELINE$_{OP}$ (23.66%) is much lower than the accuracy achieved by APOLLO (81.72%) which leverages the rich semantic information derived from Wikipedia. Furthermore, from the experimental results shown in Table 5 and Table 2, we can see that for the same set of the unlinkable test entity mentions, APOLLO obtains higher accuracy (75.34%) over the DS_{OP} data set in comparison with the accuracy (74.89%) achieved over the $DS_{NEC\rho=0.8}$ data set. The main reason is that via adding some linkable test entity mentions and the Wikipedia concepts existing in their *semantic signatures*, the graph created from the DS_{OP} data set is more beneficial for the process of label propagation. Therefore, it can be seen that APOLLO can obtain better performance over the richer graph structure, which demonstrates the scalability of APOLLO.

Table 5: Experimental results over the DS_{OP} data set

	APOLLO		BASELINE$_{OP}$	
	Accu.	#	Accu.	#
All	**0.7764**	**802**	0.5489	567
Unlinkable	**0.7534**	**498**	0.7247	479
Linkable	**0.8172**	**304**	0.2366	88

7. RELATED WORK AND DISCUSSION

The classic named entity classification task is confined to classify named entities into coarse-grained categories, such as person, location and organization. It has been investigated for several years by means of supervised approaches, which require a large number of manually tagged texts as the training data. As ontology generally contains hundreds of entity categories, supervised methods are not directly applicable for ontology-based fine-grained named entity classification, because the amount of the training data they need is too large, and the process of manually creating such annotated data requires too much human effort. Recently, many weakly supervised systems have emerged to address the task of fine-grained named entity classification [6, 28, 15, 14, 13].

Cimiano and Völker [6] addressed the fine-grained classification of named entities based on the Harris' distributional hypothesis as well as the vector space model. This method assigns a named entity to the contextually most similar concept from the ontology. The empirical results show that the pseudo-syntactic dependencies are an interesting alternative to the word window-based approaches.

Tanev and Magnini [28] proposed a weakly supervised approach to automatically populating a part of their ontology with named entities from text. For each category in the ontology, the algorithm learns a feature vector exploiting the lexico-syntactic information extracted from the contexts where the entities belonging to this category occur. They assumed that the named entities in the test data set are not ambiguous and they did not consider the problem of entity ambiguity. However, this assumption does not remain true in the real data sets.

Giuliano and Gliozzo [15] presented an instance-based learning algorithm for fine-grained named entity classification against the People ontology, an excerpt of the WordNet ontology. This proposed approach is based on a lexical substitution technique, and the plausibility of the generated sentence is estimated using the Web data. In a similar setting, Giuliano [14] proposed a kernel-based method that implicitly maps entities, represented by aggregating all contexts in which they occur, into a latent semantic space derived from Wikipedia. However, in both of these two approaches, to collect sufficient contextual information for each named entity, these systems query the search engine with the name of the entity, and consider all snippets retrieved by the search engine referring to the same entity. Therefore, they ignored the problem of ambiguity of proper names.

The approach presented in [13] is the baseline method introduced in Section 6.1.1. The experimental results in [13] show that this method significantly outperforms the single-context rule-based extractor similar to several state-of-the-art techniques for the task of fine-grained named entity classification. In this study, the authors assumed that each entity identified in the corpus has been converted to its *canonical* representation. However, in the real application, the general documents corpus cannot satisfy this assumption.

As more and more knowledge bases like DBpedia [1] and YAGO [26] are available publicly, the entity linking task has attracted great interests of many researchers starting from Bunescu and Pasca [4], who used the bag of words model to measure the cosine similarity between the context of the entity mention and the text of the Wikipedia article. Cucerzan [7] proposed a solution which is the first system to recognize the global document-level topical coherence of the entities. The system addresses the entity linking problem through a process of maximizing the agreement between the context of the entity mention and the contextual information extracted from the Wikipedia, as well as the agreement among the categories associated with the candidate entities. The learning based solution in [8] focuses on the classification framework to resolve entity linking. It develops a comprehensive feature set based on the entity mention, the contextual document and the knowledge base entry, and then uses a SVM ranker to score each candidate entity. Our previous work [24] proposed LINDEN to deal with the entity linking task. LINDEN is a novel framework to link named entities in text with a knowledge base unifying Wikipedia and WordNet, by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledge base. Overall, the essential step of the entity linking task is to define a similarity measure between the text around the entity mention and the document associated with the entity.

8. CONCLUSION

In this paper, we have studied the problem of ontology population with named entities. We propose APOLLO, a novel unified framework to resolve the task of automatic ontology population with named entities integrally via random walks on graphs. APOLLO is a weakly supervised framework and can be easily parallelized. To evaluate the effectiveness of APOLLO, a thorough experimental study was conducted, and the experimental results demonstrate that APOLLO achieves significantly higher accuracy for the ontology population task compared with the baseline method, by leveraging the rich semantic knowledge embedded in the Wikipedia. Furthermore, we extensively evaluated the performance of APOLLO over both subtasks, and the experimental results show that APOLLO outperforms the baseline methods for both subtasks.

9. ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China under Grant No. 60833003, National Basic Research Program of China (973 Program) under Grant No. 2011CB302206, and an HP Labs Innovation Research Program award.

10. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, and Z. Ives. Dbpedia: A nucleus for a web of open data. In *ISWC'07*, pages 11–15.

[2] S. Baluja, R. Seth, D. Sivakumar, Y. Jing, J. Yagnik, S. Kumar, D. Ravich, and R. M. Aly. Video suggestion and discovery for youtube: Taking random walks through the view graph. In *WWW'08*, pages 895–904.

[3] M. Banko, M. J. Cafarella, S. Soderl, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI'07*, pages 2670–2676.

[4] R. Bunescu and M. Pasca. Using encyclopedic knowledge for named entity disambiguation. In *EACL'06*, pages 9–16.

[5] R. L. Cilibrasi and P. M. B. Vitanyi. The google similarity distance. *IEEE Trans. on Knowl. and Data Eng.*, 19:370–383, March 2007.

[6] P. Cimiano and J. Völker. Towards Large-Scale, Open-Domain and Ontology-Based Named Entity Classification. In *RANLP'05*, pages 166–172.

[7] S. Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *EMNLP and CoNLL*, pages 708–716, 2007.

[8] M. Dredze, P. McNamee, D. Rao, A. Gerber, and T. Finin. Entity disambiguation for knowledge base population. In *COLING'10*, pages 277–285.

[9] A. Ekbal, E. Sourjikova, A. Frank, and S. P. Ponzetto. Assessing the challenge of fine-grained named entity recognition and classification. In *Proceedings of the 2010 Named Entities Workshop*, pages 93–101, 2010.

[10] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Unsupervised named-entity extraction from the web: an experimental study. *Artif. Intell.*, 165:91–134, 2005.

[11] C. Fellbaum, editor. *WordNet: An Electronic Lexical Database.* MIT Press, Cambridge, MA, 1998.

[12] M. Fleischman and E. Hovy. Fine grained classification of named entities. In *COLING'02*, pages 1–7.

[13] V. Ganti, A. C. König, and R. Vernica. Entity categorization over large document collections. In *SIGKDD'08*, pages 274–282.

[14] C. Giuliano. Fine-grained classification of named entities exploiting latent semantic kernels. In *CoNLL'09*, pages 201–209.

[15] C. Giuliano and A. Gliozzo. Instance-based ontology population exploiting named-entity substitution. In *COLING'08*, pages 265–272.

[16] Z. Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[17] A. McCallum and W. Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *CONLL'03*, pages 188–191.

[18] P. McNamee, H. Simpson, and H. T. Dang. Overview of the tac 2009 knowledge base population track. In *Text Analysis Conference (TAC 2009).*

[19] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. In *WIKIAI'08.*

[20] D. Milne and I. H. Witten. Learning to link with wikipedia. In *CIKM'08*, pages 509–518.

[21] D. Nadeau and S. Sekine. A Survey of Named Entity Recognition and Classification. *Journal of Linguisticae Investigationes*, 30(1):1–20, 2007.

[22] W. Shen, J. Wang, P. Luo, and M. Wang. Apollo: a general framework for populating ontology with named entities via random walks on graphs. In *WWW'12*, pages 595–596.

[23] W. Shen, J. Wang, P. Luo, and M. Wang. Liege: Link entities in web lists with knowledge base. In *SIGKDD'12*, pages 1424–1432.

[24] W. Shen, J. Wang, P. Luo, and M. Wang. Linden: linking named entities with knowledge base via semantic knowledge. In *WWW'12*, pages 449–458.

[25] W. Shen, J. Wang, P. Luo, M. Wang, and C. Yao. Reactor: a framework for semantic relation extraction and tagging over enterprise data. In *WWW'11*, pages 121–122.

[26] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A core of semantic knowledge unifying wordnet and wikipedia. In *WWW'07*, pages 697–706.

[27] P. P. Talukdar, J. Reisinger, M. Paşca, D. Ravichandran, R. Bhagat, and F. Pereira. Weakly-supervised acquisition of labeled class instances using graph random walks. In *EMNLP'08*, pages 582–590.

[28] H. Tanev and B. Magnini. Weakly supervised approaches for ontology population. In *Proceeding of the 2008 conference on Ontology Learning and Population: Bridging the Gap between Text and Knowledge*, pages 129–143.