Wikipedia Entity Expansion and Attribute Extraction from the Web Using Semi-supervised Learning *

Lidong Bing Wai Lam
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
{ldbing, wlam}@se.cuhk.edu.hk

Tak-Lam Wong
Department of Mathematics and Information Technology
The Hong Kong Institute of Education
tlwong@ied.edu.hk

ABSTRACT

We develop a new framework to achieve the goal of Wikipedia entity expansion and attribute extraction from the Web. Our framework takes a few existing entities that are automatically collected from a particular Wikipedia category as seed input and explores their attribute infoboxes to obtain clues for the discovery of more entities for this category and the attribute content of the newly discovered entities. One characteristic of our framework is to conduct discovery and extraction from desirable semi-structured data record sets which are automatically collected from the Web. A semi-supervised learning model with Conditional Random Fields is developed to deal with the issues of extraction learning and limited number of labeled examples derived from the seed entities. We make use of a proximate record graph to guide the semi-supervised learning process. The graph captures alignment similarity among data records. Then the semi-supervised learning process can leverage the unlabeled data in the record set by controlling the label regularization under the guidance of the proximate record graph. Extensive experiments on different domains have been conducted to demonstrate its superiority for discovering new entities and extracting attribute content.

Categories and Subject Descriptors

H.4.m [**Information Systems Applications**]: Miscellaneous

Keywords

Information Extraction, Entity Expansion, Proximate Record Graph, Semi-supervised Learning

* The work described in this paper is substantially supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: CUHK413510) and the Direct Grant of the Faculty of Engineering, CUHK (Project Codes: 2050476 and 2050522). This work is also affiliated with the CUHK MoE-Microsoft Key Laboratory of Human-centric Computing and Interface Technologies.

WSDM'13, February 4–8, 2013, Rome, Italy.

1. INTRODUCTION

As a remarkable and rich online encyclopedia, Wikipedia provides a wealth of general knowledge about various aspects. Some Wikipedia articles have a structured information block known as infobox, as exemplified in Figure 1, typically on the upper right of the Wikipedia page. An infobox is composed of a set of attribute name and value pairs that summarize the key information of the entity. Infobox was extensively explored in some existing projects such as DBpedia [1], Freebase [4], and YAGO [29].

Although Wikipedia already covers a large number of popular entities, many entities have not been included. Figure 2 presents a fragment of a Web page containing a semi-structured list that describes some cartoon films. After manual checking, we find that quite a number of cartoons in this list have not been covered by Wikipedia. For example, no existing Wikipedia entities describe the cartoons numbered 2, 3, 100 and 101. On the other hand, the ones numbered 319 to 321 in the same list are well contained by Wikipedia and each of them has a description article and an infobox. Therefore, with these records that are already described by some Wikipedia entities as clues, we may infer that the remaining records in the same list shown in Figure 2 are talking about the same type of entities. For example, one can infer that, from the record 100, there exists a cartoon with title "`What's Buzzin' Buzzard`" which does not exist in Wikipedia. Furthermore, considering the infobox of cartoon entity "`One Droopy Knight`" as shown in Figure 1 and its corresponding data record, i.e. record 320 in Figure 2, one can infer that the text fragment "`Directed by Michael Lah`" in the record 320 is related to the director attribute and its value. As a result, one can further infer that the director attribute of the newly discovered entity "`What's Buzzin' Buzzard`" is associated with the value "`Tex Avery`". Some other attributes of the new entity can be obtained similarly.

Inspired by the above observation, we develop a new framework to achieve the goal of new entity discovery and attribute extraction for Wikipedia categories by mining the rich and valuable semi-structured data records on the Web as exemplified in Figure 2. Our framework makes use of a few existing seed Wikipedia entities and their infoboxes automatically extracted from a particular category as clues to discover more entities of this category. It can leverage the existing infoboxes of the seed entities to automatically harvest attribute content of the newly discovered entities. Entity attribute extraction is essential for the usability of the entities in downstream applications such as knowledge base construction.

One Droopy Knight	
Droopy series	
Directed by	Michael Lah
Produced by	William Hanna Joseph Barbera
Story by	Homer Brightman
Narrated by	Bill Thompson
Voices by	Bill Thompson Daws Butler

Figure 1: The infobox of a cartoon entity.

As noticed by previous works [6, 9, 36, 40], semi-structured data records have inherent advantages in the presentation of similar entities sharing common attributes, and these entity records are usually arranged in a layout format that has some regularities, but not following a strictly fixed template. As estimated by Elmeleegy et al., 1.4% of Web pages contain lists (formatted with <dl>, <ol>, or <ul>) that can be extracted into relational tables [9]. Another estimation shows that 1.1% of the tables (formatted with <table>) on the Web contain high-quality relational-style data [6]. Since the <table> tag is commonly used for formatting and navigation, the absolute number of relational tables is very large. Note that some semi-structured data record sets are arranged in a much more complicated format than simply with <ol>, <ul> or <table> tags. Therefore, semi-structured data exists in large amount on the Web. Such kind of data records are collected as the source from which we discover new entities and their attributes.

With respect to the goal of discovering entities from the Web given some seed entities, some existing works on entity set expansion can tackle this task to some extent. Entity set expansion takes a few user given seed entities of a particular class as input, and aims at collecting more entities of the same class. SEAL [33] explores the "list" style of data records, which can be considered as a simplified kind of semi-structured data records mentioned above, to discover more entities to expand a given entity set. Specifically, it extracts named entities with wrappers [34], each of which is a pair of character-level prefix and suffix. The work [13] by Gupta and Sarawagi also processes the "list" style of data records to extract entity names and their attributes. It takes several user input data records, including entity names and attribute values, as seeds to discover more entities as well as similar specified attributes by a trained semi-Markov Conditional Random Field (semi-Markov CRF) [27] based extractor. This method requires considerable manual effort when applying it on large number of domains. Some other works focus on a more general problem setting for the task of entity set expansion [22, 23, 25]. They first obtain a set of candidate entities by some linguistics techniques such as pattern-based method. Then the similarity of a candidate with the seeds is calculated using their context distributions on the Web or Web queries. Because of the general setting, the targeted classes have coarser granularity such as city, country, etc. Moreover, these methods are not able to conduct attribute extraction of the new entities.

With respect to the goal of entity attribute extraction, some existing works [14, 30, 37, 38] train extractors on the free text of Wikipedia articles that are automatically annotated with the corresponding articles' infoboxes. Different from their direction, we explore the semi-structured data records which exist in large quantity on the Web. For example, we can extract the entity attributes of release date, director, etc. from the semi-structured list of the page shown in Figure 2. Since each data record describes one entity, the extracted attribute information can be connected to the correct subjects (i.e. entities) automatically. Consequently, our entity discovery and attribute extraction framework eliminates some error-prone operations such as coreference resolution and disambiguation.

2: CLEANING HOUSE
Rel 2/19/38
Supervised by Robert Allen

3: BLUE MONDAY
Rel 4/2/38
Supervised by William Hanna

... ...
... ...

100: WHAT'S BUZZIN' BUZZARD?
Rel 11/27/43
Directed by Tex Avery; Animation: Ed Love, Ray Abrams, Preston Bla

101: THE STORK'S HOLIDAY
Rel 10/23/43
Directed by George Gordon; Animation: Michael Lah, Rudy Zamora, C
Al Grandmain; Story: Otto Englander, Webb Smith

... ...
... ...

319: SCAT CATS
Rel 7/26/57
Directed by Joseph Barbera and William Hanna; Animation: Kenneth Muse, Carlo Vinci, Lewis Marshall; Story: Homer Brightman; Layout: Dick Bickenbach; Backgrounds: Robert Gentle

320: ONE DROOPY KNIGHT
Rel 12/6/57
Directed by Michael Lah; Animation: Bill Schipek, Ken Southworth, Ir
Story: Homer Brightman; Layout: Ed Benedict; Backgrounds: F. Monte

321: FEEDIN' THE KIDDIE
Rel 6/7/57
Directed by Joseph Barbera and William Hanna; Animation: Irvin Spen
Ray Patterson; Layout: Dick Bickenbach; Backgrounds: Don Driscoll; I
ORPHAN.

Figure 2: A Web page fragment of cartoon list.

Although semi-structured data is abundant on the Web and its inherent advantages are appealing for the task of entity discovery and attribute extraction, one big challenge of this task is that the number of available seed entities is typically very limited. As shown by the work [13], when the seed number is small, the trained semi-Markov CRF based extractor cannot perform well. However, it is time consuming and labor force intensive to prepare more seed entities' as well as their attributes. Semi-supervised approaches are proposed to ease the difficulty of lacking enough training data by taking the unlabeled data into account [12, 28]. In this paper, we propose a semi-supervised learning framework to extract the entities and their attribute content from the semi-structured data record sets. Our framework first derives training examples from a few data records by using the seed Wikipedia entities and their infoboxes as automatic labeling knowledge. We employ semi-Markov CRF as the basic sequence classification learning model. Due to the limited number of seed entities, the derived training examples are usually not sufficient to train a reliable model for extracting the entities and their attributes. A semi-supervised learning model with CRF is developed to solve this problem by exploiting the unlabeled data in the semi-structured data record set. To connect the derived training examples and the unlabeled records in the semi-supervised learning, a proximate record graph with each node representing one data record is constructed. Armed with the pairwise sequence alignment based similarity measure between the record nodes in the graph, the labels of derived training examples can regularize the labels of directly or indirectly aligned segments in the unlabeled records effectively.

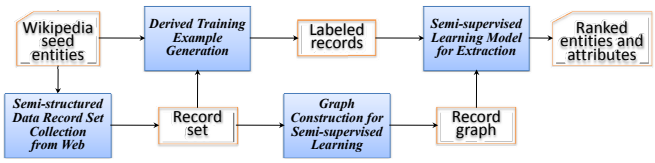

Figure 3: Architecture of our framework.

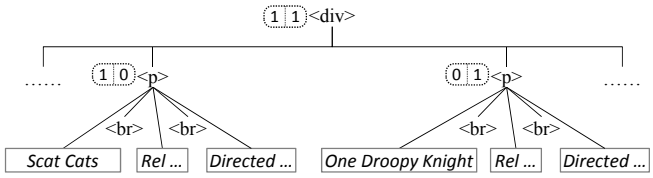

Figure 4: The DOM of the record set in Figure 2.

2. FRAMEWORK OVERVIEW

A Wikipedia category contains a set of existing entities. Taking the "`MGM animated short films`" category as an example, its existing entities include "`Scat Cats`", "`One Droopy Knight`", etc. Each single entity may have an infobox, as exemplified in Figure 1, which contains a set of attributes. Each attribute is a name-value pair (n, V), where n is the attribute name and V is the set of values. In Figure 1, an example of attribute is ("`Produced by`", {"`William Hanna`", "`Joseph Barbera`"}). Given a Wikipedia category C with several existing entities and infoboxes automatically extracted from C as clues, our framework aims at discovering more entities for the category C as well as extracting the attributes for the newly discovered entities. For discovering new entities, one example is the entity "`What's Buzzin' Buzzard`" found in the semi-structured list shown in Figure 2 and the director attribute of this new entity is "`Tex Avery`".

The architecture of our framework is depicted in Figure 3. First, the component of semi-structured data record set collection aims at automatically obtaining from the Web a collection of semi-structured data record sets which provide the sources for new entity discovery and attribute extraction. Given a Wikipedia category C, it takes several seed entities $\mathcal{S}$ automatically extracted from C as clues for constructing a synthetic query to retrieve Web pages. Then semi-structured data record sets that likely contain new entities and attributes are automatically detected from the retrieved pages. One example of such record set is given in Figure 2. Let $\mathcal{D}$ denote a record set discovered from a Web page. Some records in $\mathcal{D}$, corresponding to the seed entities in $\mathcal{S}$, are called *seed records*. The remaining records likely describe some new entities as well as their attribute values. In each record, the key information of the entity described is presented with text segments, such as cartoon name (i.e. entity name of this cartoon), release date, director, etc. The component of semi-supervised learning model for extraction in our framework aims at detecting these desirable text segments. Considering the characteristics of the task, we formulate it as a sequence classification problem. Our classification problem is different from the standard learning paradigm since we do not have manually labeled training examples. Instead, our framework automatically derives the training examples from the seed records. Such labels are known as *derived labels* and the labeled records are known as *derived training examples*. For instance, consider the record 320 in Figure 2 corresponding to the seed entity "`One Droopy Knight`" in Figure 1, our framework automatically derives labels for the text fragments corresponding to the entity name and attribute values in this record to generate a derived training example.

Let $\mathcal{D}_L$ denote the set of derived training examples in $\mathcal{D}$. Our goal is to predict the labels of text fragments for each remaining record in $\mathcal{D}$. We employ semi-Markov Conditional Random Field (semi-Markov CRF) [27] as the basic sequence classification learning model. Typically, the amount of $\mathcal{D}_L$ is quite limited, so the trained classifier by pure supervised learning would not perform well. A semi-supervised learning model is developed to solve this problem by exploiting the data records in $\mathcal{D} - \mathcal{D}_L$, which is called unlabeled data and denoted as $\mathcal{D}_U$. A proximate record graph, with each node of the graph representing one data record in $\mathcal{D}$, is proposed based on the proximate relation of records. This graph is employed to guide the regularization of the posterior label distribution of the records in $\mathcal{D}_U$ during the semi-supervised learning process. Then the labels of the records in $\mathcal{D}_U$ are inferred based on the regularized posterior. The data records in $\mathcal{D}_U$ with inferred labels are taken into account in the subsequent iteration of the training process. Finally, the new entities and their attributes extracted from different data record sets are integrated together. Then a ranking criterion is applied to return a ranked list of the entities together with their attributes.

3. SEMI-STRUCTURED DATA RECORD SET COLLECTION

As mentioned in the overview, given a Wikipedia category C, a seed entity set $\mathcal{S}$ is automatically selected from C. These seed entities are used for constructing a synthetic query to retrieve Web pages which are expected to contain semi-structured data record sets as exemplified in Figure 2. The synthetic query is composed of the seed entity names and the nouns in the category title. In addition, the query term "list" is added because this term is often used in the page titles and article titles of the desirable Web pages. One sample synthetic query for "`MGM animated short films`" category is ⟨scat cats, one droopy knight, [mgm, film, list]⟩, where the square brackets denote optional query terms. The synthetic query is then submitted to Google API to retrieve relevant documents, and the pages from Wikipedia are excluded.

In each Web page obtained above, we need to detect the semi-structured data record set describing a group of targeted entities. To do so, we design a method with two steps, namely, candidate record region identification, and record segmentation. To identify the possible record region, we make use of the seed entities as clues. We first build the DOM tree of the page and assign a flag array to each inner node of it. The number of flags in the array is equal to the number of the seed entities in $\mathcal{S}$. Each flag indicates whether the corresponding entity appears in the sub-tree rooted at the current node. After that, the flag array of each node is set following the post-order traversal. Finally, we identify the DOM node which has all 1's in its flag array, and none of its descendants meets this condition. For example, the DOM tree of the list in Figure 2 with the flag array added is shown in Figure 4, and is the DOM node that satisfies the condition.

The sub-tree identified above is the candidate region in the page that may contain the desirable record set. Next, we need to conduct data record segmentation to segment the sub-trees of the identified DOM node (e.g.) into separate data records. Several methods may be adopted to handle this task such as MDR [18], DEPTA [39], and our previous work RST [3]. Different from MDR and DEPTA which assume a fixed length of generalized nodes, RST provides a unified search based solution for record region detection and segmentation. In this framework, we employ and modify the RST method so that it only detects the records within the identified DOM tree (e.g. the tree in Figure 4). Meanwhile, the top down traversal search is not needed here since we already know that the record region's root should be the root of this sub-tree (e.g. in Figure 4). Note that this method is able to eliminate regions that cannot be segmented into records since these regions are probably not record regions.

4. SEMI-SUPERVISED LEARNING MODEL FOR EXTRACTION

In our framework, the extraction of new entities and their attributes from a particular data record set is formulated as a sequence classification problem. Recall that $\mathcal{D}$ is a semi-structured data record set identified from a Web page. Each data record $\mathbf{x}_i$ in $\mathcal{D}$ is composed of a sequence of text tokens and HTML tags. Hence, it can be represented by a token sequence $\mathbf{x}_i = x_i^1 \cdots x_i^{|\mathbf{x}_i|}$. Since some records in $\mathcal{D}$ correspond to the seed entities known as seed records, we attempt to automatically identify the text fragments corresponding to entity names and attribute values based on the infobox information. For example, consider the sample data record 320, i.e., the cartoon "`One Droopy Knight`" in Figure 2. By using the infobox of the seed Wikipedia entity as given in Figure 1, the text fragment "`ONE DROOPY KNIGHT`" is identified as the entity name and labeled with "ENTITY_NAME" label. The text fragment "`Michael Lah`" is identified as the director attribute value of the cartoon and labeled with "DIRECTED_BY" label. Unlike traditional labels that are provided by human, the above labels are automatically derived from the infobox and called derived labels. The label "OTHER" is used to label the tokens that do not belong to the entity names and attribute values.

After the seed records are automatically labeled with the derived labels, we obtain a set of derived training examples denoted as $\mathcal{D}_L = \{(\mathbf{x}_i, \mathbf{s}_i)\}$, where $\mathbf{s}_i = s_i^1 \cdots s_i^{|\mathbf{s}_i|}$ and $s_i^q = \langle t_i^q, u_i^q, y_i^q \rangle$ is a token segment with the beginning index t_i^q, the ending index u_i^q, and the derived label y_i^q. Our goal is to predict the labels of the text fragments for each record in the remaining part of $\mathcal{D}$, namely $\mathcal{D}_U = \mathcal{D} - \mathcal{D}_L$, so as to extract new entity names and their attribute values described by the record. The details of the component for generating the derived training examples will be presented in Section 5.

We adopt the semi-Markov CRF [27] as the basic sequence classification learning model. As mentioned, the amount of $\mathcal{D}_L$ is quite limited since only a few seed entities are given. As a result, the performance of the trained classifier with the ordinary supervised learning is limited. To tackle this problem, we develop a semi-supervised learning model which exploits the data records in $\mathcal{D}_U$. To better utilize $\mathcal{D}_U$, a graph-based component is designed to guide the semi-supervised learning process. Specifically, we propose

1: **input**: record set $\mathcal{D}$ ($\mathcal{D}_L \cup \mathcal{D}_U$)
2: $\mathcal{D}_U^{(0)} \leftarrow \mathcal{D}_U$, $n \leftarrow 0$
3: $\Lambda^{(0)} \leftarrow$ train semi-Markov CRF on $\mathcal{D}_L$
4: $\mathcal{G} \leftarrow$ construct proximate record graph on $\mathcal{D}$
5: $\hat{\mathbf{P}} \leftarrow$ calculate empirical distribution on $\mathcal{D}_L$
6: **while** *true* **do**
7: $\quad\mathbf{P} \leftarrow$ estimate label distribution for each record $\mathbf{x}_i$ in $\mathcal{D}$ with $\Lambda^{(n)}$
8: $\quad\mathbf{P}^* \leftarrow$ regularize $\mathbf{P}$ with $\hat{\mathbf{P}}$ according to graph $\mathcal{G}$
9: $\quad\tilde{\mathbf{P}} \leftarrow$ interpolate distributions of $\mathbf{P}^*$ and $\mathbf{P}$
10: $\quad\mathcal{D}_U^{(n+1)} \leftarrow$ inference label of $\mathcal{D}_U$ with $\tilde{\mathbf{P}}$
11: $\quad$**if** $\mathcal{D}_U^{(n+1)}$ same as $\mathcal{D}_U^{(n)}$ **or** $n = maxIter$ **then**
12: $\quad\quad$**goto** Line 17
13: $\quad$**end if**
14: $\quad\Lambda^{(n+1)} \leftarrow$ train semi-Markov CRF on $\mathcal{D}_L \cup \mathcal{D}_U^{(n+1)}$
15: $\quad n \leftarrow n + 1$
16: **end while**
17: **return** $\mathcal{D}_U^{(n+1)}$

Figure 5: The algorithm of our Semi-supervised Learning Model.

a graph called proximate record graph where each node of the graph represents a record in $\mathcal{D}$. Each edge represents the connected data records with high degree of similarity in both of text content and HTML format. The high-level pseudo-code of our semi-supervised learning algorithm is given in Figure 5. At the beginning, we train the initial parameters of the semi-Markov CRF on $\mathcal{D}_L$ in Line 3. Before performing the semi-supervised learning, the construction of the proximate record graph $\mathcal{G}$ is conducted using the records in $\mathcal{D}$ in Line 4. The details of the construction will be discussed in Section 4.1. In Lines 7 and 8, the proximate record graph $\mathcal{G}$ guides the regularization of the posterior label distribution $\mathbf{P}$ of the records. The details are presented in Section 4.3. In Lines 9 and 10, the regularized distribution $\mathbf{P}^*$ is interpolated with the original posterior distribution $\mathbf{P}$ to produce an updated label distribution $\tilde{\mathbf{P}}$ which is used in the inference to get the predicted labels for the records in $\mathcal{D}_U$. The details are presented in Section 4.4. Then, if the stopping conditions in Line 11 are not met, the algorithm proceeds to the next iteration, and the records in $\mathcal{D}_U$ with the inferred labels in the current iteration are involved in the semi-Markov CRF training in Line 14 as discussed in Section 4.5. Finally the labeling results of $\mathcal{D}_U$ in the last iteration are returned as the output.

After all record sets corresponding to category C are processed, the extracted entities and attribute values are integrated together. The details are described in Section 4.6.

4.1 Proximate Record Graph Construction

A proximate record graph $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ is an undirected weighted graph with each record in $\mathcal{D}$ as a vertex and $\mathcal{E}$ is the edge set. Recall that each record $\mathbf{x}_i$ in $\mathcal{D}$ is represented by a token sequence $\mathbf{x}_i = x_i^1 \cdots x_i^{|\mathbf{x}_i|}$. Each edge $e_{ij} = (\mathbf{x}_i, \mathbf{x}_j)$ connecting the vertices $\mathbf{x}_i$ and $\mathbf{x}_j$ is associated with a weight w_{ij} calculated as:

$$w_{ij} = \begin{cases} \mathbb{A}(\mathbf{x}_i, \mathbf{x}_j) & if\ \mathbf{x}_i \in \mathbb{K}(\mathbf{x}_j)\ or\ \mathbf{x}_j \in \mathbb{K}(\mathbf{x}_i) \\ 0 & otherwise \end{cases}, \quad (1)$$

where $\mathbb{A}$ is a pairwise sequence alignment function returning a score in $[0, 1]$ which indicates the proximate relation

between the two record sequences of $\mathbf{x}_i$ and $\mathbf{x}_j$. $\mathbb{K}(\cdot)$ is a function that can output the top k nearest neighbors. In the proximate record graph, only the vertices (i.e., records) sharing similar format and content will be connected by the edges. Taking the record set given in Figure 2 as an example, $\mathbf{x}_i$ denotes the token sequence obtained from the data record number i. There are edges such as $(\mathbf{x}_{319}, \mathbf{x}_{320})$ and $(\mathbf{x}_{319}, \mathbf{x}_{321})$. However, it is unlikely that there is an edge between $\mathbf{x}_2$ and $\mathbf{x}_{319}$.

To design the alignment function $\mathbb{A}$, we employ a modified Needleman-Wunsch algorithm [21] which performs a global alignment on two sequences using the dynamic programming technique. In particular, we use unit penalty per gap, zero penalty per matching, and infinite penalty per mismatching. The overall alignment penalty of two sequences is converted and normalized into a similarity score in $[0, 1]$. Considering data records $\mathbf{x}_2$ and $\mathbf{x}_3$ in Figure 2, their token sequences are "*<p> DIGIT : cleaning house
 rel DATE* $\cdots$", and "*<p> DIGIT : blue monday
 rel DATE* $\cdots$" respectively. The alignment result is illustrated as below:

$$\begin{array}{lllllll} <p> DIGIT: & cleaning\ house & - & - &
 & rel & \cdots \\ <p> DIGIT: & - & - & blue\ monday &
 & rel & \cdots \end{array}$$

where "-" represents a gap token. After sequence alignment, we obtain a set of aligned token pairs $\mathcal{A}^t_{ij} = \{(x^m_i, x^n_j)\}$, where (x^m_i, x^n_j) indicates that x^m_i from $\mathbf{x}_i$ and x^n_j from $\mathbf{x}_j$ are aligned. Furthermore, we can also define the set of aligned segment pairs of $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$\begin{aligned} \mathcal{A}^s_{ij} \quad &= \quad \{(s^q_i, s^r_j)\} \\ &s.t. \quad (a).\neg\exists(x^m_i, x^n_j) \in \mathcal{A}^t_{ij} \wedge x^m_i \in s^q_i \wedge x^n_j \in s^r_j, and \\ &\qquad (b).(x_i^{t^q_i-1}, x_j^{t^r_j-1}) \in \mathcal{A}^t_{ij} \wedge (x_i^{u^q_i+1}, x_j^{u^r_j+1}) \in \mathcal{A}^t_{ij}, and \\ &\qquad (c).u^q_i - t^q_i \leq maxL \wedge u^r_j - t^r_j \leq maxL. \end{aligned}$$

The first condition constrains that an aligned segment pair does not contain any aligned token pairs. The second condition constrains that the left (right) neighboring tokens of the segment pair are aligned. The last condition constrains that the length of both segments should be less than or equal to the maximum length $maxL$. In the above example, one aligned segment pair is ("*cleaning house*", "*blue monday*"). Note that the alignment relation between segments is transitive. Therefore, although $\mathbf{x}_2$ and $\mathbf{x}_{319}$ are not directly connected by an edge, the segments "*cleaning house*" and "*scat cats*" can still be aligned indirectly with the paths from $\mathbf{x}_2$ to $\mathbf{x}_{319}$ in $\mathcal{G}$.

As shown by the above example, it is beneficial to make the labels of aligned segments favor towards each other. The reason is that the text fragments having the same label in different records often share the same or similar context tokens. Meanwhile, they are often presented in the similar relative locations in their own token sequences. The aligned segments generated from our pairwise sequence alignment algorithm can capture the above two characteristics at the same time. The context tokens of some "OTHER" segments may also be similar to that of the desirable segments. The "OTHER" segments in the unlabeled records are often directly or indirectly aligned to the "OTHER" segments in the labeled records. The label regularization in the semi-supervised learning can help predict the correct labels for such situations.

4.2 Semi-Markov CRF and Features

As mentioned in the overview of our framework in Section 2, our semi-supervised extraction model employs semi-Markov CRFs [27] as the basic classification learning model. In particular, a linear-chain CRF model is used. Let $\mathbf{s}_i = s^1_i \cdots s^{|\mathbf{s}_i|}_i$ denote a possible segmentation of $\mathbf{x}_i$, where $s^q_i = \langle t^q_i, u^q_i, y^m_i \rangle$ as defined above. Then the likelihood of $\mathbf{s}_i$ can be expressed as:

$$P(\mathbf{s}_i|\mathbf{x}_i; \Lambda) = \frac{1}{Z(\mathbf{x}_i)} \exp\{\Lambda^T \cdot \sum_q \mathbf{f}(y^{q-1}_i, s^q_i, \mathbf{x}_i)\}, \quad (2)$$

where $\mathbf{f}$ is a feature vector of the segment s^q_i and the state of the previous segment. Λ is the weight vector that establishes the relative importance of all the features. $Z(\mathbf{x}_i)$ is a normalizing factor.

To avoid the risk of overfitting with only a few derived training examples in our problem setting, the only feature template used in our framework is the separator feature:

$$f_{j,j',t,t',d}(y^{q-1}_i, s^q_i, \mathbf{x}_i) = \mathbf{1}_{\{y^q_i = d\}} \mathbf{1}_{\{x_i^{t^q_i - j} = t\}} \mathbf{1}_{\{x_i^{u^q_i + j'} = t'\}}, \quad (3)$$

where $j, j' \in \{1, 2, 3\}$. t and t' vary over the separators such as delimiters and tag tokens in the sequence $\mathbf{x}_i$. d varies over the derived labels of the record set from where $\mathbf{x}_i$ originates. This class of features are unrelated to the previous state, so the feature vector $\mathbf{f}$ can be simplified to $\mathbf{f}(s^q_i, \mathbf{x}_i)$ in our framework.

4.3 Posterior Regularization

In Lines 7 and 8 of the semi-supervised learning model shown in Figure 5, the posterior label distribution is regularized with the guidance of the proximate graph $\mathcal{G}$. Once the parameters of the CRF model are trained, the posterior probability of a particular segment s^q_i in the sequence $\mathbf{x}_i$, denoted by $P(s^q_i|\mathbf{x}_i; \Lambda)$, can be calculated as:

$$P(s^q_i|\mathbf{x}_i; \Lambda) = \frac{1}{Z'} \exp\{\Lambda^T \cdot \mathbf{f}(s^q_i, \mathbf{x}_i)\}, \quad (4)$$

where $Z' = \Sigma_y \exp\{\Lambda^T \cdot \mathbf{f}(\langle t^q_i, u^q_i, y\rangle, \mathbf{x}_i)\}$. Let the vector $\mathbf{P}_{s^q_i} = (P(\langle t^q_i, u^q_i, y\rangle|\mathbf{x}_i; \Lambda))^T$ denote the posterior label distribution of s^q_i. To regularize this distribution with the proximate graph $\mathcal{G}$, we minimize the following function:

$$O(\mathbf{P}) = O_1(\mathbf{P}) + \mu O_2(\mathbf{P}), \quad (5)$$

where μ is a parameter controlling the relative weights of the two terms. $O_1(\mathbf{P})$ and $O_2(\mathbf{P})$ are calculated as in Equations 6 and 7:

$$O_1(\mathbf{P}) = \sum_{\mathbf{x}_i \in \mathcal{D}_L} \sum_{b=1}^{|\mathbf{x}_i|} \sum_{s:\langle b, b+l-1, y\rangle}^{s.t. 1 \leq l \leq maxL, \; b+l-1 \leq |\mathbf{x}_i|} \|\hat{\mathbf{P}}_s - \mathbf{P}_s\|, \quad (6)$$

$$\begin{aligned} O_2(\mathbf{P}) \quad = \quad & \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} w_{ij} \Big(\sum_{(x^m_i, x^n_j) \in \mathcal{A}^t_{ij}} \|\mathbf{P}_{x^m_i} - \mathbf{P}_{x^n_j}\| \\ & + \sum_{(s^q_i, s^r_j) \in \mathcal{A}^s_{ij}} \|\mathbf{P}_{s^q_i} - \mathbf{P}_{s^r_j}\| \Big), \end{aligned} \quad (7)$$

where $\|\cdot\|$ is the Euclidean norm. $\hat{\mathbf{P}}_s$ is the empirical label distribution of the segment s in $\mathbf{x}_i$. When $l > 1$, $\hat{P}(\langle b, b+$

$l-1, y\rangle|\mathbf{x}_i)$ is calculated as:

$$\hat{P}(\langle b, b+l-1, y\rangle|\mathbf{x}_i) = \sum_{b\le m\le b+l-1} \hat{P}(\langle m, m, y\rangle|\mathbf{x}_i)/l, \quad (8)$$

and $\hat{P}(\langle m, m, y\rangle|\mathbf{x}_i)$ is obtained from the derived label sequence of $\mathbf{x}_i$. In Equation 5, the term $O_1(\mathbf{P})$ regularizes the estimated posterior distribution of the derived training examples in $\mathcal{D}_L$ with the original empirical labeling. The term $O_2(\mathbf{P})$ regularizes the estimated posterior distribution of the aligned token and segment pairs.

To achieve efficient computation, we employ the iterative updating method to obtain the suboptimal solution $\mathbf{P}^*$ of Equation 5. The updating formulae are:

$$\mathbf{P}'_{s_i^q} = \hat{\mathbf{P}}_{s_i^q}\mathbf{1}_{\{\mathbf{x}_i\in\mathcal{D}_L\}} + \mu \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{E}} w_{ij} \sum_{(s_i^q, s_j^r)\in\mathcal{A}_{ij}^s} \mathbf{P}_{s_j^r}, \quad (9)$$

and

$$\mathbf{P}'_{x_i^m} = \hat{\mathbf{P}}_{x_i^m}\mathbf{1}_{\{\mathbf{x}_i\in\mathcal{D}_L\}} + \mu \sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{E}} w_{ij} \sum_{(x_i^m, x_j^n)\in\mathcal{A}_{ij}^t} \mathbf{P}_{x_j^n}, \quad (10)$$

where $\mathbf{P}$ is the old estimation from the last iteration. Note that after each iteration, $\mathbf{P}'$ should be normalized so as to meet the condition $\|\mathbf{P}'\|_1 = 1$, where $\|\cdot\|_1$ is the ℓ_1 norm. Then $\mathbf{P}'$ will be regarded as the old estimation for the next iteration.

4.4 Inference with Regularized Posterior

In Lines 9 and 10 of the algorithm shown in Figure 5, the obtained $P^*(s_i^q|\mathbf{x}_i)$ in the regularization is employed to adjust the original $P(s_i^q|\mathbf{x}_i;\Lambda)$ which is calculated with the parameters obtained from the training in the last iteration. The interpolation function is given in Equation 11:

$$\tilde{P}(s_i^q|\mathbf{x}_i;\Lambda) = (1-\upsilon)P^*(s_i^q|\mathbf{x}_i) + \upsilon P(s_i^q|\mathbf{x}_i;\Lambda), \quad (11)$$

where υ is a parameter controlling the relative weights of P^* and P. Then we calculate the new feature value related to s_i^q on $\mathbf{x}_i$ as $\tilde{P}(s_i^q|\mathbf{x}_i;\Lambda) * Z'$. This value is utilized in the Viterbi decoding algorithm to infer new label sequences for the records in $\mathcal{D}_U$.

Recall that the proximate graph $\mathcal{G}$ captures the alignment relation between similar segments and this relation is transitive. Therefore, the decoded labels with the interpolated posterior distribution favor towards the labels of their aligned neighboring segments.

4.5 Semi-supervised Training

As shown in Line 11 of the algorithm in Figure 5, if the inferred labels of the records in $\mathcal{D}_U$ from the previous step are the same as the ones obtained in the last iteration, the semi-supervised learning process terminates. Otherwise, if n is still smaller than the maximum iteration number, the records in $\mathcal{D}_U$ with new labels will be taken into consideration in the next iteration of the training process. In this training, we maximize the penalized log likelihood function as shown below:

$$\begin{aligned}\ell(\Lambda^{(n)}) &= \eta \sum_{(\mathbf{x}_i,\mathbf{s}_i)\in\mathcal{D}_L} \log P(\mathbf{s}_i|\mathbf{x}_i;\Lambda^{(n)}) \quad (12)\\ &+ (1-\eta) \sum_{(\mathbf{x}_i,\mathbf{s}_i^*)\in\mathcal{D}_U^{(n)}} \log P(\mathbf{s}_i^*|\mathbf{x}_i;\Lambda^{(n)}) + \gamma\|\Lambda^{(n)}\|,\end{aligned}$$

where η is a parameter controlling the relative weights of the contribution from the records in $\mathcal{D}_L$ and $\mathcal{D}_U$. The term $\gamma\|\Lambda^{(n)}\|$ is the Euclidean norm penalty factor weighted by γ. $\mathbf{s}_i^*$ is the segmentation of $\mathbf{x}_i$ inferred from the previous step. Obviously, the above objective function is still concave and it can thus be optimized with the efficient gradient descent methods such as L-BFGS [19].

4.6 Result Ranking

In our semi-supervised leaning model for extraction as shown in Figure 5, each data record set is processed separately. Thus, the same entity may be extracted from different record sets where it appears with its different variant names. Before ranking the result entities, we first conduct entity deduplication. The number of occurrence of the same entity is counted during the deduplication process. Then the entities are ranked according to their number of occurrence. After entity deduplication, the attributes of the same entity collected from different record sets are integrated together, and they are also deduplicated and ranked similarly. The details of variant collecting and approximate string matching utilized in the deduplication will be presented in Section 5 since they are also used in the generation of derived training examples.

5. DERIVED TRAINING EXAMPLE GENERATION

As mentioned before, in a semi-structured data record set $\mathcal{D}$, the seed records refer to the data records that correspond to the seed entities in $\mathcal{S}$. The goal of derived training example generation is to automatically identify seed records in $\mathcal{D}$ and determine the sequence classification labels for these records using the information of the seed infoboxes. Since such labels are not directly provided by human as in the standard machine learning paradigm, we call them derived labels. Moreover, the records, after determining the derived labels, are called derived training examples. The generation task can be decomposed into two steps, namely, seed record finding and attribute labeling.

To find the seed record in $\mathcal{D}$ for a seed entity E, we first find the records that contain E's name or its variants as a sub-sequence. The name variants are obtained from synonym in WordNet and the redirection relation in Wikipedia. If the entity name is a person name detected by YagoTool[1], we also collect its variants by following the name conventions, such as middle name removal, given name acronym, etc. In addition, we allow an approximate string matching as supplement in case that the collected variants are not sufficient. The found records are regarded as candidate seed records of E, and the matching sub-sequences are regarded as the candidate record name segments. If multiple record name segments are found in one candidate seed record, the one that has the smallest index in the record is retained and the others are discarded. We adopt this strategy because the subject of a record, i.e. the record name segment, is usually given in the very beginning of the record [8]. When multiple candidate seed records are found for E, the one whose record name segment has the smallest index in its own token sequence is returned as the seed record. This treatment can handle the case that the entity name of E appears as an attribute value or plain text in other non-seed records.

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

The found record name segment of the seed record is labeled with the derived label "ENTITY_NAME".

The seed records found above compose the derived training set $\mathcal{D}_L$. The procedure of labeling the attribute values in these seed records is similar to the labeling of the entity name. The attribute values for labeling are collected from the seed entities' infoboxes and their variants are obtained in a similar manner as above. In addition, we use YagoTool to normalize the different formats of date and number types. The variant set of each attribute value goes through the same procedure above for entity name except that we only search the value variant in its own seed record. The derived labels of attribute values are obtained from the seed entity's infobox, such as "DIRECTED_BY", "STORY_BY", etc. After the labeling of attribute values, the remaining unlabeled parts in the seed records are labeled with "OTHER".

6. EXPERIMENTS

6.1 Experiment Setting

We collect 16 Wikipedia categories as depicted in Table 1 to evaluate the performance of our framework. These categories are well-known so that the annotators can collect the full set of the ground truth without ambiguity. Therefore, the experimental results are free from the annotation bias. Moreover, some of these categories are also used in the experimental evaluation of existing works, such as SEAL [33] which will also be compared in our experiments.

To obtain the ground truth of entity elements, the annotators first check the official Web site of a particular category if available. Normally, the full entity list can be found from the Web site. For example, the teams of NBA can be collected from a combox in the home page. For the categories that do not have their own official Web sites such as African country, our annotators try to obtain the entity elements from other related organization Web sites, e.g. the Web site of The World Bank. In addition, the entity lists of some categories are also available in Wikipedia.

Wikipedia already contains articles for some entities in the above categories, and quite many of these articles have well maintained infoboxes. This is helpful for us to collect the ground truth attribute values with high quality for conducting the evaluation of the attribute extraction results. For each entity, the annotators collect the ground truth attribute values for each attribute that also appeared in the seed entities' infoboxes of the same category. Since these infoboxes are used to generate the derived training examples, the extracted attribute values have the same label set as the derived label set from these infoboxes. Hence, the collected ground truth attribute values can be used to evaluate the results. During the collection of attribute values, if the entity exists in Wikipedia and has a good quality infobox, our annotators use the infobox first. After that, they search the Web to collect the values for the remaining attributes.

For each category, the semi-structured data collection component randomly selects two seed entities from the existing ones in Wikipedia to generate a synthetic query. This query is issued to Google API to download the top 200 hit Web pages. The discovery of entities and the extraction of their attributes are carried out with the semi-structured record sets detected from the downloaded pages. This procedure is executed 3 times per category so as to avoid the possible bias introduced by the seed entity selection. The average of these 3 runs is reported as the performance on this category. It is worthwhile to notice that when three or more seeds are available, we can enhance the performance by generating several synthetic queries with different combinations of the seeds and aggregate the results of these synthetic queries. For example, three synthetic queries each of which involves two seeds can be generated from three seed entities.

Table 1: The details of the Wikipedia categories collected for the experiments.

Category ID	Category Name	# of entities
1	African countries	55
2	Best Actor Academy Award winners	78
3	Best Actress Academy Award winners	72
4	Best Picture Academy Award winners	83
5	Counties of Scotland	33
6	Fields Medalists	52
7	First Ladies of the United States	44
8	Leone d'Oro winners	57
9	Member states of the European Union	27
10	National Basketball Association teams	30
11	Nobel laureates in Chemistry	160
12	Nobel laureates in Physics	192
13	Presidents of the United States	44
14	Prime Ministers of Japan	66
15	States of the United States	50
16	Wimbledon champions	179

In our framework, the parameter setting is chosen based on a separate small development data set. The tuned parameters are $\mu = 0.1$, $\upsilon = 0.2$, and $\eta = 0.01$. In the proximate record graph construction, each vertex is connected to its 3 nearest neighbors. The iteration numbers are 20 and 10 for regularization updating and semi-supervised learning respectively. Following the setting in [27], $maxL$ and γ are set to be 6 and 0.01.

6.2 Entity Expansion

For entity expansion, we conduct comparison with two methods. The first one is a baseline that employs supervised semi-Markov CRF as the extraction method (called CRF-based baseline). It also makes use of the collected semi-structured record sets by our framework as the information resource and takes the derived training examples in our framework as the training data to perform supervised learning. The second comparison method is an existing work, namely SEAL [33]. SEAL also utilizes the semi-structured data on the Web to perform entity set expansion. It generates a character-level wrapper for each Web page with the seed entities as clues. We collect 200 results from the system[2] Web site of SEAL per seed set per category.

Both precision and recall are used to evaluate the ranking results of entity discovery. We first report the result of precision at K (P@K) where K is the number of entities considered from the top of the ranked output list by a method. For each method, the value of K varies over 5, 20, 50, 100, and 200. The results of entity discovery of different methods are given in Table 2. It can be observed that all three methods achieve encouraging performance. Their average P@5 values are 0.94, 0.90 and 0.90 respectively. This demon-

[2]http://www.boowa.com/

Table 2: The precision performance of entity discovery of different methods.

	Our framework					CRF-based baseline					SEAL				
#	@5	@20	@50	@100	@200	@5	@20	@50	@100	@200	@5	@20	@50	@100	@200
1	1.00	1.00	0.94	0.48	0.25	1.00	0.90	0.80	0.47	0.25	1.00	1.00	0.92	0.47	0.25
2	1.00	1.00	0.94	0.76	0.38	1.00	0.90	0.84	0.68	0.38	1.00	1.00	0.92	0.71	0.38
3	1.00	0.95	0.92	0.67	0.35	1.00	0.90	0.82	0.58	0.31	1.00	0.85	0.74	0.48	0.29
4	1.00	1.00	0.98	0.81	0.41	1.00	0.93	0.90	0.77	0.41	1.00	1.00	0.98	0.79	0.40
5	0.53	0.43	0.28	0.16	0.11	0.40	0.35	0.28	0.16	0.11	0.40	0.40	0.22	0.12	0.07
6	1.00	1.00	0.94	0.50	0.25	1.00	0.95	0.84	0.43	0.25	1.00	1.00	0.86	0.46	0.24
7	1.00	0.93	0.72	0.42	0.21	0.93	0.85	0.60	0.41	0.21	1.00	0.80	0.62	0.40	0.21
8	0.87	0.85	0.74	0.44	0.24	0.73	0.68	0.48	0.34	0.24	0.80	0.72	0.51	0.26	0.14
9	0.80	0.72	0.46	0.23	0.12	0.60	0.55	0.44	0.23	0.12	0.80	0.55	0.42	0.22	0.12
10	1.00	1.00	0.54	0.28	0.14	1.00	0.93	0.48	0.25	0.14	1.00	1.00	0.52	0.27	0.14
11	1.00	0.95	0.90	0.88	0.66	1.00	0.87	0.84	0.68	0.52	1.00	0.95	0.86	0.71	0.48
12	1.00	0.97	0.86	0.84	0.73	1.00	0.83	0.78	0.62	0.58	1.00	0.95	0.92	0.57	0.52
13	1.00	1.00	0.80	0.41	0.21	1.00	0.93	0.72	0.37	0.19	1.00	0.98	0.76	0.39	0.20
14	1.00	1.00	0.84	0.52	0.32	1.00	0.95	0.76	0.44	0.30	1.00	0.98	0.82	0.42	0.30
15	1.00	1.00	0.95	0.48	0.24	1.00	0.93	0.88	0.48	0.24	1.00	1.00	0.96	0.48	0.24
16	0.80	0.68	0.52	0.33	0.19	0.67	0.62	0.48	0.33	0.18	0.47	0.35	0.34	0.25	0.14
avg.	0.94	0.91	0.77	0.52	0.31	0.90	0.82	0.68	0.46	0.28	0.90	0.85	0.71	0.44	0.26
P-value in pairwise t-test						0.014	9.90E-08	1.25E-05	0.002	0.025	0.074	0.011	0.004	0.001	0.010

strates that semi-structured data on the Web is very useful in this task. Therefore, making use of the semi-structured Web data to enrich some categories of Wikipedia is a feasible and practical direction. On average, the performance of our framework with semi-supervised learning extraction is better than SEAL. One reason for the superior performance is that our framework detects the semi-structured data record region before the extraction. This can eliminate the noise blocks in the Web pages. The character-level wrapper of SEAL may be affected by these noise blocks. In addition, with only a few seeds, the character-level wrapper of SEAL cannot overcome the difficulty caused by the cases that the seed entity names have inconsistent contexts. For example, one seed name is embedded in a tag <font> while the other seed name is not. The wrapper induction procedure will be misled by this inconsistency. Moreover, when the text segments of the names have similar format contexts as that of other segments of the records, the obtained wrapper will also extract more false positives. The CRF-based baseline also suffers from the above context related difficulties. Our semi-supervised learning method can cope with this problem by taking the unlabeled data records into account. Specifically, the sequence alignment based proximate record graph regularizes the labels according to the alignment relation so as to overcome the difficulties brought in by the ambiguous context. We conduct pairwise t-test using a significance level of 0.05 and find that the improvements of our framework are significant in most cases. The P-values in the significance tests are given in the last row of Table 2.

We manually check some categories with low performance. One major type of error for Category 9 is due to the non-EU European countries, such as "Ukraine" and "Switzerland". When we retrieve the semi-structured data records from the Web, the seeds of EU member countries also serve as the seeds for non-EU European countries in a counterproductive manner. For the category of Scotland county, many noisy place names are extracted. The main reason is that the entity names in this category are also widely used to name other places all over the nations of British Commonwealth. Consequently, the collected large amount of noisy semi-structured data records affect the performance.

We also report the recall of different methods in Table 3. Since the number of the ground truth entities in different categories varies from dozens to near 200, the recall values are calculated at different K values of the result list, namely 50, 100, and 200. If the ground truth number is no more than a particular K value, the recall for this K value is calculated. On average, our framework outperforms SEAL by about 7%. One reason is that the noise output of SEAL affects the ranking of the true positives, and some of them are ranked lower. Another possible reason is that the search engine cannot return sufficient number of semi-structured record sets. This also affects the performance of our framework.

6.3 Attribute Extraction

Another major goal of our framework is to extract attribute values for the detected entities. SEAL does not perform entity attribute extraction, therefore, we only report the performance of our framework and the CRF-based baseline. For each domain, we only examine the extracted attributes of the correct entities in its answer entity list and report the average precision at different K values from 1 to 10 in all domains. Note that about one fifth correctly extracted entities do not have detected attributes and they are excluded from this evaluation.

The result of attribute extraction performance is given in Figure 6. It can been seen that our framework can perform significantly better than the CRF-based baseline with a difference about 11% on average at different K levels. It is worth emphasizing that our framework can achieve 16% improvement on average when K is no more than 5. It indicates that our framework is rather robust. We manually check the extracted attributes of CRF-based baseline and find that besides noise segments, it sometimes wrongly identifies the name of an entity as an attribute value. The reason is that the CRF-based baseline only depends on the separator fea-

Table 3: The precision performance of entity discovery of different methods.

	Our framework			CRF baseline			SEAL		
K / #	50	100	200	50	100	200	50	100	200
1	–	0.91	0.94	–	0.89	0.94	–	0.89	0.94
2	–	0.79	0.87	–	0.78	0.84	–	0.79	0.82
3	–	0.74	0.83	–	0.68	0.83	–	0.62	0.79
4	–	1.00	1.00	–	0.95	1.00	–	0.97	0.99
5	0.45	0.52	0.71	0.44	0.52	0.71	0.35	0.39	0.45
6	–	1.00	1.00	–	0.92	1.00	–	0.92	1.00
7	0.81	1.00	1.00	0.72	0.97	1.00	0.74	0.90	0.95
8	–	0.71	0.84	–	0.62	0.80	–	0.41	0.51
9	0.84	0.92	0.96	0.81	0.92	0.96	0.81	0.88	0.96
10	0.96	1.00	1.00	0.86	0.89	1.00	0.93	0.96	1.00
11	–	–	0.81	–	–	0.66	–	–	0.61
12	–	–	0.74	–	–	0.59	–	–	0.52
13	0.93	0.97	1.00	0.86	0.88	0.90	0.89	0.90	0.93
14	–	0.83	1.00	–	0.75	1.00	–	0.76	1.00
15	0.98	1.00	1.00	0.92	1.00	1.00	1.00	1.00	1.00
16	–	–	0.19	–	–	0.18	–	–	0.16
avg.	0.83	0.88	0.87	0.77	0.85	0.84	0.79	0.80	0.79

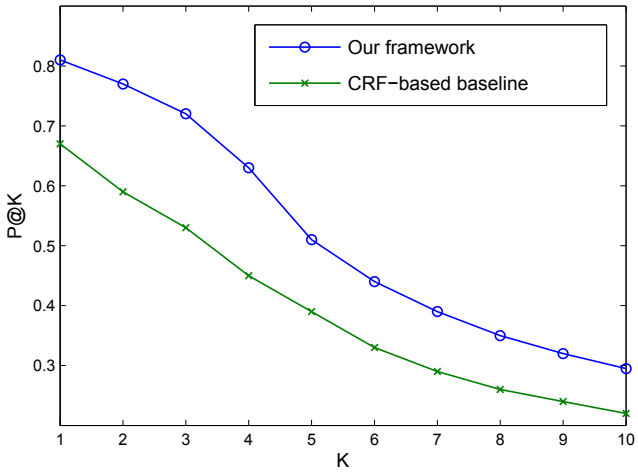

Figure 6: The attribute extraction performance of our framework and the CRF-based baseline.

tures to identity segments. Thus they cannot distinguish those segments with similar context tokens. Our framework can handle these cases by regularizing the label distribution with the proximate record graph as guidance. As a result, noise segments can be regularized to favor towards the segments labeled as "OTHER" in the derived training examples. Similarly, the segments of the entity name and the attribute value with similar context tokens can also be properly identified. In addition, it can be seen that the attributes of higher rank have better precision. Specifically, our framework can output 2 true positive attribute values among the top 3 extracted attribute results. It is because the important attributes of an entity are repeatedly mentioned in different data record sets, such as the capital and GDP of an African country, the party and spouse of a US president, etc. Therefore, they are ranked higher according to the counted occurrence time. We also find that some correct attributes are not ranked very high. One possible reason is due to the simple design of the final ranking method which only counts the number of occurrence of the extracted attributes.

7. RELATED WORK

SEAL [33] exploits "list" style of data, which can be considered as a simplified kind of semi-structured data records in this paper, to discover more entities for expanding a given entity set. They extract named entities with wrappers, each of which is composed of a pair of character-level prefix and suffix [34]. However, they do not perform record region detection, consequently the wrappers may extract noise from the non-record regions of the page. Context distribution similarity based methods [23, 25] utilize the context of the seed entities in the documents or Web search queries to generate a feature/signature vector of the targeted class. Then the candidate entities are ranked according to the similarity of their feature vectors with the class's vector. Different from the above methods that explore positive seed instances only, Li et al [16] proposed a learning method that takes both positive and unlabeled learning data as input and generates a set of reliable negative examples from the candidate entity set. Then the remaining candidates are evaluated with the seeds as well as the negative examples. Ensemble semantics methods [7, 26] assemble the existing set expansion techniques as well as their information resources to boost the performance of a single approach on a single resource. The output of named entity recognition [20] can serve as one source to perform set expansion [23, 25, 26]. All the above methods cannot extract attributes of the discovered entities.

Entity set acquisition systems [5, 10] do not need input seeds. They leverage domain independent patterns, such as "is a" and "such as", to harvest the instances of a given class. Open information extraction [2, 11] and table semantifying [17, 31, 32] focus more on extracting or annotating large amount of facts and relations. The methods of weakly-supervised attribute acquisition [22, 24] can also be applied in identifying important attributes for the categories. Thus, the existing infoboxes can be polished, and the non-infobox categories can obtain proper attributes to establish their own infobox schemata.

Among the semi-supervised CRF approaches, one class of methods consider data sequence granularity [12, 15, 35]. Precisely, these methods incorporate one more term in the objective function of CRF. This term captures the conditional entropy of the CRF model or the minimum mutual information on the unlabeled data. The extra term can be interpreted by information theory such as rate distortion theory. However, the objective function does not possess the convexity property any more. Subramanya et al. also constructed a graph to guide the semi-supervised CRF learning in part-of-speech tagging problem [28]. This method regularizes the posterior probability distribution on each single token in a 3-gram graph. Its n-gram based graph construction is not applicable to the problem tackled here. The reason is that the length of our desirable text segments cannot be fixed in advance. In contrast, our proximate record graph can capture both record level and segment level similarities at the same time. Furthermore, the proximate record graph is also able to capture the position information of a text segment in the alignment so that the aligned segments with higher chance describing the same functionality components of different records.

8. CONCLUSIONS AND FUTURE WORK

In this paper, a framework of Wikipedia entity expansion and attribute extraction is presented. This framework takes a few seed entities automatically collected from a particular category as well as their infoboxes as clues to harvest more entities as well as attribute content by exploiting the semi-structured data records on the Web. To tackle the problem of lacking sufficient training examples, a semi-supervised learning model is proposed. A proximate record graph is designed, based on pairwise sequence alignment, to guide the semi-supervised learning. Extensive experimental results can demonstrate that our framework can outperform a state-of-the-art existing system. The semi-supervised learning model achieves significant improvement compared with pure supervised learning.

Several directions are worth exploring in the future. One direction is to investigate how to perform derived training example generation with the article of the seed entity when its infobox does not exist. Another direction is to detect new attributes in the semi-structured data record sets that are not mentioned in the infoboxes. Such new attributes are valuable to populate the relations in knowledge bases.

9. REFERENCES

[1] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. Dbpedia: a nucleus for a web of open data. In *ISWC/ASWC*, pages 722–735, 2007.

[2] M. Banko, M. J. Cafarella, S. Soderl, M. Broadhead, and O. Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[3] L. Bing, W. Lam, and Y. Gu. Towards a unified solution: data record region detection and segmentation. In *CIKM*, pages 1265–1274, 2011.

[4] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, 2008.

[5] M. J. Cafarella, D. Downey, S. Soderland, and O. Etzioni. Knowitnow: fast, scalable information extraction from the web. In *HLT*, pages 563–570, 2005.

[6] M. J. Cafarella, A. Halevy, D. Z. Wang, E. Wu, and Y. Zhang. Uncovering the relational web. In *WebDB*, 2008.

[7] A. Carlson, J. Betteridge, R. C. Wang, E. R. Hruschka, and T. M. Mitchell. Coupled semi-supervised learning for information extraction. In *WSDM*, pages 101–110, 2010.

[8] E. Crestan and P. Pantel. Web-scale table census and classification. In *WSDM*, pages 545–554, 2011.

[9] H. Elmeleegy, J. Madhavan, and A. Halevy. Harvesting relational tables from lists on the web. *Proc. VLDB Endow.*, 2:1078–1089, 2009.

[10] O. Etzioni, M. Cafarella, D. Downey, S. Kok, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates. Web-scale information extraction in knowitall (preliminary results). In *WWW*, pages 100–110, 2004.

[11] O. Etzioni, A. Fader, J. Christensen, S. Soderland, and Mausam. Open information extraction: The second generation. In *IJCAI*, pages 3–10, 2011.

[12] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, pages 529–536. 2004.

[13] R. Gupta and S. Sarawagi. Answering table augmentation queries from unstructured lists on the web. *Proc. VLDB Endow.*, 2:289–300, 2009.

[14] R. Hoffmann, C. Zhang, and D. S. Weld. Learning 5000 relational extractors. In *ACL*, pages 286–295, 2010.

[15] F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans. Semi-supervised conditional random fields for improved sequence segmentation and labeling. In *ACL*, pages 209–216, 2006.

[16] X.-L. Li, L. Zhang, B. Liu, and S.-K. Ng. Distributional similarity vs. pu learning for entity set expansion. In *ACLShort*, pages 359–364, 2010.

[17] G. Limaye, S. Sarawagi, and S. Chakrabarti. Annotating and searching web tables using entities, types and relationships. *Proc. VLDB Endow.*, 3:1338–1347, 2010.

[18] B. Liu, R. Grossman, and Y. Zhai. Mining data records in web pages. In *KDD*, pages 601–606, 2003.

[19] D. C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45:503–528, 1989.

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

[21] S. Needleman and C. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[22] M. Paşca. Organizing and searching the world wide web of facts – step two: harnessing the wisdom of the crowds. In *WWW*, pages 101–110, 2007.

[23] M. Paşca. Weakly-supervised discovery of named entities using web search queries. In *CIKM*, pages 683–690, 2007.

[24] M. Paşca and B. V. Durme. Weakly-supervised acquisition of open-domain classes and class attributes from web documents and query logs. In *ACL*, pages 19–27, 2008.

[25] P. Pantel, E. Crestan, A. Borkovsky, A.-M. Popescu, and V. Vyas. Web-scale distributional similarity and entity set expansion. In *EMNLP*, pages 938–947, 2009.

[26] M. Pennacchiotti and P. Pantel. Entity extraction via ensemble semantics. In *EMNLP*, pages 238–247, 2009.

[27] S. Sarawagi and W. W. Cohen. Semi-markov conditional random fields for information extraction. In *NIPS*, pages 1185–1192, 2004.

[28] A. Subramanya, S. Petrov, and F. Pereira. Efficient graph-based semi-supervised learning of structured tagging models. In *EMNLP*, pages 167–176, 2010.

[29] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Web Semant.*, 6:203–217, 2008.

[30] F. M. Suchanek, M. Sozio, and G. Weikum. Sofie: a self-organizing framework for information extraction. In *WWW*, pages 631–640, 2009.

[31] P. Venetis, A. Halevy, J. Madhavan, M. Paşca, W. Shen, F. Wu, G. Miao, and C. Wu. Recovering semantics of tables on the web. *Proc. VLDB Endow.*, 4:528–538, 2011.

[32] J. Wang, B. Shao, H. Wang, and K. Q. Zhu. Understanding tables on the web. Technical report, 2010.

[33] R. C. Wang and W. W. Cohen. Language-independent set expansion of named entities using the web. In *ICDM*, pages 342–350, 2007.

[34] R. C. Wang and W. W. Cohen. Character-level analysis of semi-structured documents for set expansion. In *EMNLP*, pages 1503–1512, 2009.

[35] Y. Wang, G. Haffari, S. Wang, and G. Mori. A rate distortion approach for semi-supervised conditional random fields. In *NIPS*, pages 2008–2016. 2009.

[36] T.-L. Wong and W. Lam. Learning to adapt web information extraction knowledge and discovering new attributes via a bayesian approach. *IEEE Trans. on Knowl. and Data Eng.*, 22:523–536, 2010.

[37] F. Wu and D. S. Weld. Autonomously semantifying wikipedia. In *CIKM*, pages 41–50, 2007.

[38] F. Wu and D. S. Weld. Open information extraction using wikipedia. In *ACL*, pages 118–127, 2010.

[39] Y. Zhai and B. Liu. Web data extraction based on partial tree alignment. In *WWW*, pages 76–85, 2005.

[40] J. Zhu, Z. Nie, J.-R. Wen, B. Zhang, and W.-Y. Ma. Simultaneous record detection and attribute labeling in web data extraction. In *KDD*, pages 494–503, 2006.