OnPerDis: Ontology-based Personal Name Disambiguation on the Web

Zhao Lu
Dept. of Computer Science and Technology
East China Normal University
Shanghai, China
Email: zlu@cs.ecnu.edu.cn

Zhixian Yan
Samsung Research America
San Jose, USA
Email: zhixian.yan@samsung.com

Liang He
Dept. of Computer Science and Technology
East China Normal University
Shanghai, China
Email: lhe@cs.ecnu.edu.cn

Abstract—With the growth of web documents, the ambiguity of personal name becomes more common and brings poor performance of web search. Identifying a correct personal entity from the a piece of or the whole document is still a very challenging problem, especially for Chinese websites. In this paper, we propose a novel Ontology-based approach for Personal Name Disambiguation (named “OnPerDis”). This approach has two main steps: first, we construct person ontology (PO) with rich conceptual modeling as well as a large set of supporting instances; second, for a given personal name on the web, we create a temporary instance and extract features from the web documents, calculate the similarity between this temporary instance and the instances in the PO. The one with the highest similarity score is chosen as the appropriate personal name. Our extensive evaluations with two rich real-life datasets (CIPS-SIGHAN 2012 NERD and Chinese web documents) shows OnPerDis’ efficacy on personal name disambiguation on the Web.

Keywords—Personal name disambiguation; Conceptual modeling; Ontology population; Instance matching;

I. INTRODUCTION

Recently, web search engines become vital in people’s daily life and are widely used to retrieve information of real-world entities including people themselves. In such cases, users enter the name of the target entity in search engines to obtain a set of Web pages that contain the name. However, the ambiguous of name (many entities share the same name or an entity has several names) typically causes ambiguous search results containing Web pages of several different entities. Such ambiguity is more common in Chinese names. For example, when search “Yao Ming”, the results are dominated by the well-known basketball player, and users have to manually fitter out these Web pages to identify the expected non-famous people who share the same name. This is the personal name ambiguity problem.

Traditional studies on personal name disambiguation focus on identifying the features of Web documents, measuring the similarity, and clustering the documents into different groups that indicates different personal entities [1][2]. For example, Ono et al. cluster the retrieved Web documents containing the names based on three mixed features, i.e., co-occurrences of named entities, key compound words and latent topic information [4]. Fleischman and Hovy propose a Maximum Entropy model to assign the probability that two names refer to the same individual [3]. Srinivasan et al. present an approach that computes a similarity between entities identified in a document with those in the knowledge base using a Vector Space Model utilizing document level entity profiles [5]. Other works pay attention on the synonym name problem using query expansions [23].

Recently, the task of name disambiguation is Entity Linking (EL), which links mentions of entities within specific contexts to their corresponding entries in an existing knowledge base (KB), such as Wikipedia and YAGO. Traditional entity linking systems typically search for candidate entities and then disambiguate them by assuming name mentions in a document to be independent, and returning either the best candidate or Nil [19]. Collective entity linking exploiting the interdependence between name mentions in the same document either in a pair-wise or a referent graph [21]. These methods highly dependent on the accuracy of named entity extraction, while named entity extractions, especially Chinese named entity extraction, currently remain a very challenging task.

Despite the wide studies of personal name disambiguation, accurately identifying Chinese people for a document with the same search Chinese name is still a very challenging task because of the following reasons:

- Most existing studies focus on personal names in English and these methods are not suitable for Chinese names. Chinese entities have less morphology variations compared to other languages, and there are no special symbols to imply boundaries between two words in Chinese, which cause the difficulty to extract person entities and their features from texts.
- These methods focus on independent features like named entities, compound keywords, translation-based language models [25], or use pair-wises and referent graphs to model relationships between two entities. The structure, semantic and relationship among personal features from texts are not used, which might be the reason for low performance.

To address these challenges, in this paper, we propose an ontology based approach for personal name disambiguation
on the Web, called “OnPerDis”. The OnPerDis approach has two main parts, i.e., (1) build a rich person ontology (PO) and construct a large scale of instances from Baike celebrity\(^1\) a well-known Chinese Wikipedia; (2) apply PO to evaluate the similarity of Web documents, and identify the most similar personal entity in the PO instances. This paper has the following four main contributions:

- We construct a comprehensive person ontology (PO). The PO conceptually models discriminative features of the people including a set of classes and their properties. The semantics are also captured in terms of relationships and constraints.
- We design a person ontology population algorithm to automatically construct PO instances: construct a large set of person instances using Baike celebrity or other KBs, with two suggested personal feature extractions, pattern-based feature extraction from Baike infobox and feature-aware extraction from texts.
- To identify the latent person instance from the web documents, we design a Recursive Evaluation of Ontology Instances Similarity algorithm to compare the personal name instances at both the concept-level and the property-level.
- We evaluate OnPerDis with three experiments on two datasets, one is suggested by CIPS-SIGHAN 2012 Chinese Named Entity Recognition and Disambiguation (NERD), and the other is a rich Web document dataset from Baikepedia, Wikipedia and Yahoo. The results show the efficacy of OnPerDis on Chinese personal name disambiguation.

The remaining sections of this paper are organized as follows. Section 2 describes related works about personal name disambiguation. Section 3 presents the main framework of our OnPerDis approach. Section 4 introduces the person ontology modeling and instance population. Section 5 discusses the details of using ontology to make personal name disambiguation. Section 6 describes the experimental evaluations. Finally, we conclude this paper and discuss the future work.

II. RELATED WORKS

The literature related to OnPerDis can be divided into four main categories, i.e., personal name disambiguation, entity lining, ontology population and ontology matching.

**Personal name disambiguation** – is generally conducted through clustering web documents to identify different namesakes for a given name. Ikeda et al.\(^2\) explain an approach for improving the recall values of NE (Name-Entity) based approaches by using two-stage clustering. Hereinafter, topic-related information is applied to improve the clustering\(^3\), which requires that the data already indexed in search engines. A key-phrased clustering method is designed with the combination of a second step reclassification to improve the clustering performance\(^4\). Bunescu et al.\(^5\) use Wikipedia knowledge to disambiguate NEs. Another similar approach extends existing bag-of-words features with Wikipedia concepts\(^6\). Our approach extracts personal features from Baike celebrity and further populates ontology to better model the features.

**Entity Linking** – recently provides new thoughts to personal name disambiguation. The traditional entity linking methods usually link name mentions in a document by assuming them to be independent\(^7\). However, the entities in the same document should be semantically related to each other. Collective Entity Linking (CEL), in which the name mentioned in the same document are linked jointly by exploiting the interdependence between them in a pair-wise fashion\(^8\). Another graph-based CEL method is suggested that models and exploits the global interdependence between different EL decisions using referent graph\(^9\). However, all of these EL methods pay attention only on name mentioned in a text; they do not extract entity features and have no analysis on relationships and structure of the features. On the contrary, in this paper we study on populating personal instances of the person ontology PO, which models both features and relationships between personal name entities.

**Ontology population** – is generally performed by means of some kinds of ontology-based information extraction\(^10\). This consists of (1) identifying the key terms in the text (such as named entities and technical terms) and (2) relating them to concepts in the ontology. Ontology population can be classified into two main paradigms according to their extraction methods: pattern-based methods and contextual features methods. Pattern-based approaches search for phrases that explicitly show that there is an “is-a” relationship between two words. However, such phrases do not always appear in a text corpus. The second ones use contextual features, such as using a corpus to extract features from the context in which a semantic class tends to appear. A hybrid approach integrates pattern, term structure and contextual features to build automatic ontology population from texts\(^11\). Our approach is a hybrid approach that combines the suggested pattern-based and feature-aware extraction for automatic ontology population from Baike infoboxes and from texts.

**Ontology matching** – mostly focuses on schema level, while, at the time being, instance-level matching techniques are only marginally considered\(^12\). A method that accomplishes comparing instances of tourism ontology concepts in two phases is presented in\(^13\). SERIMI matches instances between a source dataset and a target dataset without prior knowledge of the data, or the domain or schema of these dataset\(^14\), which has a two-phase instance-matching solution: the selection phase and the disambiguation phase. In our study, we design an ontology instance similarity measurement that compares two personal instances at both concept-level and property-level.

---

\(^1\)Baike celebrity, [http://baike.baidu.com/renwu/](http://baike.baidu.com/renwu/)

---
III. OUR OnPerDis APPROACH

In this section, we first formulate the problem, and then describe our approach in general.

The personal name disambiguation problem can be mainly considered to classify a set of personal names \( N = \{n_1, n_2, ..., n_k\} \) in a web message \( d \), whether they are related to a given personal entity \( e \), here \( n_i \) is a name string.

We assume that each message contains personal names as sub-strings. We say that the personal name \( n_i \) is related to the person \( e \), \( related(n_i, e) \), if and only if the related features of the personal name \( n_i \) are high similar to the features of the person \( e \). We use \( E = \{e_1, e_2, ..., e_j, ..., e_m\} \) to represent the person set. Here, \( e_j \) is a person. So, the problem of personal name disambiguation can be converted into the function of mapping an element \( n_i \) in the set \( N \) to an element \( e_j \) in the set \( E \), that is, \( f : n_i \rightarrow e_j \).

To tackle the above problem, we design the OnPerDis approach. Our approach includes two main parts: (1) the step of personal ontology modeling and instances population, and (2) the step of personal name disambiguation. The general structure of the suggested approach is shown in Figure 1.

![Figure 1. The general structure of the OnPerDis approach](image)

For each instance \( IE \) with the same name \( n_i \) in PO, we conduct two-level similarity computation between two instances (\( IS \) and \( IE \)), concept-level similarity and property-level similarity. We select the instance \( IE \) with the highest value as the particular person the name \( n_i \) refers to. If there is no instance in PO with the same name \( n_i \), then OnPerDis return empty (i.e., Nil).

IV. PERSON ONTOLOGY MODELING AND POPULATION

We now describe the details of the first part of OnPerDis, that is, building the person ontology.

A. Extract Personal Features

An entity based twitter message classification is proposed in [12], determining whether the messages are related to a given company name. Considering the tweet messages are typically short and contain very little information, the method constructs a couple of profiles for each company, which contain richer information, e.g., homepage profile using company homepage URL, metadata profile using meta keywords extracted from homepages, category profile from the category to which the company belongs, UserFeedback positive profile and UserFeedback negative profile.

Similar to their observations about classifying company names, we summarize features of personal names in terms of four aspects: (1) various personal registration tables, (2) a considerable amount of personal profile pages, (3) the Attribute Extraction Subtask (AES) of WePS-2 for the reason that WePS evaluation campaign reports some personal features are very useful for name disambiguation [14], and (4) concepts, relationships, and attributions defined in the SUMO top-level ontology [2]. We summarize more than 40 attributes of the People. Examples of attributes of the People are shown in the second column of Table I. We use “Yao Ming”, a famous Basketball players in China, as an example in Table I. We view the relevant values of an attribute as one personal features of a person. In the third column of Table I, we show some personal features of Yao Ming. The personal features are extracted from his page in Baike celebrity.

<table>
<thead>
<tr>
<th>ID</th>
<th>Attributes</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Chinese name</td>
<td>Yao Ming</td>
</tr>
<tr>
<td>2</td>
<td>Alias</td>
<td>Little giant</td>
</tr>
<tr>
<td>3</td>
<td>Gender</td>
<td>Male</td>
</tr>
<tr>
<td>4</td>
<td>Nationality</td>
<td>Han</td>
</tr>
<tr>
<td>5</td>
<td>Date of birth</td>
<td>Sep. 12, 1980</td>
</tr>
<tr>
<td>6</td>
<td>Affiliates</td>
<td>NBA Rockets</td>
</tr>
<tr>
<td>7</td>
<td>Place of birth</td>
<td>Shanghai</td>
</tr>
<tr>
<td>8</td>
<td>Height</td>
<td>226 cm</td>
</tr>
<tr>
<td>9</td>
<td>Specialty</td>
<td>Accurate jumper from 20 feet away</td>
</tr>
<tr>
<td>10</td>
<td>Social relations</td>
<td>Ye Li</td>
</tr>
<tr>
<td>11</td>
<td>Important events</td>
<td>Special film “Year of Yao Ming” issued</td>
</tr>
<tr>
<td>12</td>
<td>Achievements</td>
<td>ESPN the world’s most potential athletes Award (2000)</td>
</tr>
<tr>
<td>13</td>
<td>Graduated university</td>
<td>Shang Hai Jiao Tong University</td>
</tr>
<tr>
<td>14</td>
<td></td>
<td>Lawrence the world new excellent athletes Award (2003)</td>
</tr>
<tr>
<td>15</td>
<td></td>
<td>The Chinese Basketball Outstanding Contribution Award</td>
</tr>
<tr>
<td>16</td>
<td></td>
<td>.....</td>
</tr>
</tbody>
</table>

We classify all collected attributes of the People into five sections, i.e., Name, Basic info., Introduction, Contact and Personal relationship. Name section includes Chinese name, alias and other name. Examples include in Basic info. section are simple facts (such as, gender, date of birth and place of birth) and nationality. Examples include in Introduction section are work, honor, education, resident and important events. Contact section refers to email, mail address, telephone and others. Personal relationship section

\[^2\]SUMO ontology portal, [http://www.ontologyportal.org/](http://www.ontologyportal.org/)
include relatives, parties, communities and others.

Till now, Baike celebrity annotates more than 470 thousand persons, each person has a relevant page with various biographical data layouts, such as with (or without) a Baike infobox and texts. After investigating various layouts of person pages, we classify persons in Baike celebrity into three categories: (1) The first category refers to great persons whose names are not shared in Baike, such as Mao zhedong. Baike annotates this type persons more comprehensive, includes using a Baike infobox and some texts; (2) The second category means famous persons, a name is shared from 2 up to 9 persons, e.g., Yao Ming. Baike describes the type person more completely using a Baike infobox together with some description sentences; (3) The third category is allowing to ordinary persons, their names are shared more than 10 persons, for example, Wang ke. There are some sentences to describe their special personal features.

Considering the distinctions of extracting personal features from infoboxes and extracting personal features from texts, we suggest both pattern-based extraction and feature-aware extraction to extract personal features from infoboxes and texts respectively.

Pattern-based extraction from infobox.

Baike infobox is a semi-structured, which has following characters: Firstly, an infobox contains several A:B formats, each format relates to an attribute A and its corresponding values B. For example, a “Chinese name:Yao Ming” format, here “Chinese name” is an attribute of a person, “Yao Ming” is the value of the attribute. Secondly, for an attribute A, its values B maybe some simple sentences, which means these sentences are used to describe some composite information of the attribute. As the last row of Table I shown, the “Achievements” attribution of Yao Ming has three values, each value is a simple sentence.

In our approach, we view all A:B formats as patterns, and our pattern-based extraction is that taking all values after : as the values of an attribute of a person.

Feature-aware extraction from texts.

Baike also use some sentences to describe a person. Compared with extracting personal features from infobox, extracting personal features from texts are relatively difficult.

After investigation examples of values of attributes, we further classify values of attributes into four types: Traditional named entity, Special entity, Noun phrases and Description. Then, we suggest feature-aware extraction for the four types of values from three aspects: (1) using pre-trigger-words and post-trigger-words which indicating values of attributes; (2) considering the characteristics of attributes, e.g., the values of the attribute “Data of Birth” are Date type; and (3) employing left-borders and right-borders of an attribution indicating the starts and the ends of values of an attribution to extract values of attributes without special characteristics, such as the values of the attribute “Important events” have not significant characteristic. We list the suggested feature-aware extraction from texts and some examples in Figure 2.

B. Conceptual Model of Person Ontology

Personnel features can contain some intrinsic properties and social relationships with other individuals. We construct an person ontology to organize these features well.

The person ontology is described as a four-tuple $PO = (C, P, R, PV)$. Here, $C$ is the concept set, $P$ is the property set, $R$ represents the set of relationships among concepts, or properties, or concepts with their properties, or properties with their property-values. There are three types of relationships in PO, i.e., kind-of, part-of and instance-of. $PV$ refers to the property-value set.

Further, we distinguish data properties and object properties. Data properties are used to describe simple personal features, such as Chinese name and Education, while object properties are used to describe the associations between a person and other named entities, e.g., Affiliate.

We organize all concepts and properties of PO into a hierarchy [18]. In PO, a node represents a concept, and attached data of a concept node has three means, properties, property-values, and instances respectively. Edges between two nodes describe relationships between concepts and properties. Using PO, we can have a formal and explicit specification of concepts and properties related with person entities, and express semantic relationships between persons and their personal features. A fragment of the person ontology is shown in Figure 3.

<table>
<thead>
<tr>
<th>Types of attributes</th>
<th>Notes</th>
<th>Examples of attributes</th>
<th>Extraction rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional named entities</td>
<td>The values of attributes are named entities with special characters</td>
<td>Alias, Social relations, Graduated schools, Place of Birth, Date of Birth, etc.</td>
<td>A. Part of Speech Tagging matching B. A+ Pre-trigger-words and Post-trigger-words</td>
</tr>
<tr>
<td>Special entities</td>
<td>The values of attributes are strong composition characters</td>
<td>Email, Telephone number, Personal homepage, etc.</td>
<td>Regular expression matching</td>
</tr>
<tr>
<td>Noun phrases</td>
<td>The values of attributes are enumeration data or records in dictionaries</td>
<td>Job-occupation, Nationality, Awards, Specialty, etc.</td>
<td>A. Dictionary matching B. A+ pre-trigger-words and post-trigger-words</td>
</tr>
<tr>
<td>Descriptions</td>
<td>The values of attributes are sentences</td>
<td>Achievements, Research directions, Important events, etc.</td>
<td>Pre-trigger-words and post-trigger-words + Left and right borders</td>
</tr>
</tbody>
</table>

Figure 2. Feature-aware extraction from texts and some examples

Figure 3. A fragment of the person ontology
C. Person Ontology Population

The main steps of the person ontology population algorithm are: First, parse the person ontology PO using an ontology parsing tool, the Jena API. Then take the person pages corresponding to various personal names in Baike celebrity as the dataset to automatically create the instances using pattern-based extraction and feature-aware extraction.

For example, we query the name “Yao Ming” in Baike celebrity, the person page which contains the personal profile of Yao Ming is returned, thus we firstly create a new instance named “Yao Ming”. As Yao Ming is famous person, his page includes an infobox, we extract his personal features using the way of personal features extracting from Baike infobox. After, we assign these extracted personal features to the corresponding properties of the new person instance.

V. PERSONAL NAME DISAMBIGUATION PROCEDURE

With the previous person ontology, we can now present the second part of OnPerDis, i.e., the detailed procedure for personal name disambiguation. It has four main steps:

Step 1. According to the concepts and properties defined in PO, extract personal features related to the name mention in a document, and organize these personal features together with the name mention as a temporary instance IS according to the structure of PO.

Step 2. Use the ontology query language SPARQL to query all instances IEs of PO sharing the same name.

Step 3. Calculate the similarities between IS and the queried instances IEs both from concept-level and property-level using a recursive evaluation of ontology instances similarity algorithm.

Step 4. After ranking the similarities, we select the top one instance related person as the particular person.

A. Instance Similarity Measurement

Given two instances IS and IE, we measure the similarity between them both from concept-level and property-level.

Concept-level similarity computation.

We measure the concept-level similarity (cSim) of two instances as follows,

\[ cSim(IS, IE) = \frac{\sum_{c_1 \in C_1} \sum_{c_2 \in C_2} \text{sim}(c_1, c_2)}{|C_2|} \]  

(1)

where \( C_1 \) and \( C_2 \) represent two sets of all concepts in two instances IS and IE; \( c_1 \) and \( c_2 \) represent two concepts of two sets \( C_1 \) and \( C_2 \); \( |C_2| \) denotes the size of the concept set \( C_2 \); \( \text{sim}(c_1, c_2) \) is the similarity between two concepts \( c_1 \) and \( c_2 \). Here, if \( c_1 \) and \( c_2 \) are identical or synonyms, then \( \text{sim}(c_1, c_2) = 1 \), else \( \text{sim}(c_1, c_2) = 0 \).

Property-level similarity computation.

We measure the property-level similarity (pSim) of two instances as follows,

\[ pSim(IS, IE) = \frac{\sum_{v_1 \in V_1} \sum_{v_2 \in V_2} \text{sim}(v_1, v_2)}{|V_1|} \]  

(2)

where \( V_1 \) and \( V_2 \) represent two sets of all values in two instances IS and IE; \( v_1 \) and \( v_2 \) represent two property-values containing in two sets \( V_1 \) and \( V_2 \). Here, \( \text{sim}(v_1, v_2) \) denotes the similarity between two values \( v_1 \) and \( v_2 \).

As we discussed above, there are two kinds of property-values in PO, data type and object type, we also consider the two types as follows:

1. For two data type property-values, if \( v_1 \) and \( v_2 \) are similar or the same, there is \( \text{sim}(v_1, v_2) = 1 \), else let \( \text{sim}(v_1, v_2) = 0 \).

2. For two object type property-values, there is,

\[ \text{sim}(v_1, v_2) = \frac{|SV_1 \cap SV_2|}{|SV_1| + |SV_2| - |SV_1 \cap SV_2|} \]  

(3)

where, \( SV_1 \) and \( SV_2 \) denote two sets of words contained in two property-values \( v_1 \) and \( v_2 \); \( |SV_1| \) and \( |SV_2| \) are the sizes of two sets \( SV_1 \) and \( SV_2 \); \( |SV_1 \cap SV_2| \) denotes the number of all common words in two sets \( SV_1 \) and \( SV_2 \).

Here, the common words refer to that, suppose \( A \) and \( B \) denote two sets of words contained in two long strings respectively, if two corresponding words \( a \) and \( b \) in the sets \( A \) and \( B \) are same or similar, we view the two words \( a \) and \( b \) as a pair of common words.

Measuring similarity of two instances.

For two instances, we use the following equation to compute the similarity (iSim) between two instances:

\[ iSim(IS, IE) = \lambda \times cSim(IS, IE) + (1 - \lambda) \times pSim(IS, IE) \]  

(4)

where \( \lambda \in [0, 1] \), \( cSim(IS, IE) \) measures the similarity between the instances of IS and IE, and \( pSim(IS, IE) \) measures the similarity between the properties of IS and IE. We will discuss the value of \( \lambda \) in the last paragraph of Section VI.

We conclude the characteristics of similarities between two instances here: (1) The similarity score is expressed as a number in the range \([0, 1]\). (2) The similarity value of two completely different instances is 0, while the similarity value of two almost same instances is 1. (3) The similarity value of two instances is also characterized in reflexivity (where an instance is theoretically to itself) and symmetry (if two instances \( X \) equals to \( Y \), then \( Y \) reciprocally equals to \( X \)).

Basic steps of the recursive evaluation of ontology instances similarity algorithm are depicted in Figure 4. The comparison process first acquire all classes, properties and property-values of two instances and parse the two instance
to two trees, then start from the two root nodes, compute their concept-level similarity or property-level similarity. After, access two corresponding sub-nodes in the same layer of the two root nodes, if the two sub-nodes are concept node, compute their concept-level similarity, else if the two sub-nodes are property-value node, then compute their property-level similarity. The match procedure recursively until all sub-nodes have not descendants.

Figure 4. Procedure of recursive evaluation on ontology instances similarity.

VI. EVALUATION AND ANALYSIS

In this section, we design three experiments to evaluate OnPerDis on two dataset, one is Chinese web documents from Baike, Wikipedia and Yahoo, the other is the dataset suggested by CIPS-SHIHAN 2012 NERD.

During the experiment procedure, we assume that one text focuses on one person. We identify person name mention in a text using the approach suggested by the work [22], which can precise identify Chinese person name, Chinese Transliterated Person Names and Japanese person name. The average precision of person name identifications in Chinese documents are higher than 95%.

A. First Experiment on Personal Features Extraction

As we discussed above, two key steps of the OnPerDis approach are creating instances and measuring similarity between two instance. Both of the two steps need to extract personal features either from infoboxes or texts. In this section, we design the first experiment to evaluate the accuracy of extracting personal features.

We use Precision (Pa), Recall (Ra) and F-measure (Fa) to evaluate the performance of personal feature extraction as follows.

\[
Pa = \frac{\text{Number of correctly extracted personal features}}{\text{Number of all extracted personal features}} \tag{5}
\]

\[
Ra = \frac{\text{Number of correctly extracted personal features}}{\text{Number of all personal features contained in texts}} \tag{6}
\]

\[
Fa = \frac{2 \times P \times R}{P + R} \tag{7}
\]

We design the first experiment from the following issues: (1) There are two ways to describe persons in Baike celebrity, infobox and texts. Data in infoboxes are semi-structure, so we can extract personal features correctly using the pattern-based extraction. Considering the difficulty of extracting exactly personal features from texts, we suggest feature-aware extraction. (2) Both persons of the first category and the second category generally are described by infoboxes, it is not necessary to extract personal features from texts. While for persons of the third category, we need to extract personal features from texts.

Taking into account above issues, we select ten person names of the third category, examples include Li Chen and Song Jia. Each name has several Baike pages (each page refers to one person). We collect more than 100 Baike pages of the ten person names as the test dataset. We select 10 attributions of the People which cover the four attribution types discussed in Figure 2, i.e., Nationality (Nat.), Honorary title (H. T.), Place of birth (P. T.), Date of birth (D. B.), Affiliate (Aff.), Job occupation (J. O.), Graduated university (G. U.), Awards (Awa.), Important events (I. E.), and Social relations (S. R.). Values of each attribution in test texts are from 20 up to 60. The first experimental results are shown in Figure 5.

Figure 5 shows that, two attributions, Place of birth and Date of birth, have higher precision scores; Four attributions, Awards, Nationality, Graduated university and Position, have relatively higher precision rates; Other three attributions, Honorary Title, Affiliate and Important events, their precision rates are relatively lower. The main reasons of above experimental results are: (1) The first attribute type, e.g., Place of birth and Date of birth, have obvious trigger words, these trigger words help to locate the positions of values of these attributions. In addition, the values of these attributions have their own characteristics, examples are date format and place format. (2) The values of the fourth attribution types, for instance, Important events and Affiliates, are texts. It is
obvious that there are less trigger words or borders to locate their values.

B. Second Experiment on CIPS-SIGHAN 2012 Dataset

The CIPS-SIGHAN 2012 NERD\(^3\) is the task of detecting entity mentions from raw text and classifying each mention to its real world entity. A collection of web documents (the Source) and a Knowledge Base (NameKB) which contains the targets of disambiguation are provided. There are 32 XML documents in NameKB, in each document there are from 3 to 37 candidate entities with the same name, and each entity has a short description (non-structure text). Each name relates to 37 to 416 source texts. In all texts for 32 person name, the number of instances person name refer to are from 4 to 166, the person name mentions are strong random.

We compute the Precision (Pre), Recall (Rec) of a name \(n\) suggested by CIPS-SIGHAN 2012 as follows,

\[
\text{Pre}(n) = \frac{\sum_{SL_{XX} \in SL} \sum_{t \in SL_{XX}} |SL_{XX} \cap L_{XX}|}{\sum_{SL_{XX} \in SL} |SL_{XX}|} \quad (8)
\]

\[
\text{Rec}(n) = \frac{\sum_{L_{XX} \in L} \sum_{t \in L_{XX}} |SL_{XX} \cap L_{XX}|}{\sum_{L_{XX} \in L} |L_{XX}|} \quad (9)
\]

For each name \(n\), there is a collection of test documents for evaluation. Evaluation is carried out on a per document basis. Let \(T\) denote the document collection for one name, for each query document \(t\) of \(T\), our approach may fall into two classes, namely: SL_{XX} or Nil, representing in KB id or not found in NameKB respectively. The gold label is L_{XX}.

For the name collection \(N\), our approach overall Precision (Pre), Recall (Rec) and F are computed as follows,

\[
\text{Pre} = \frac{\sum_{n \in N} \text{Pre}(n)}{|N|} \quad (10)
\]

\[
\text{Rec} = \frac{\sum_{n \in N} \text{Rec}(n)}{|N|} \quad (11)
\]

\[
F = \frac{2 \times \text{Pre} \times \text{Rec}}{\text{Pre} + \text{Rec}} \quad (12)
\]

We automatically create 473 instances for 32 names in NameKB\(^3\). We compare our approach with other three algorithms submitted to CIPS-SIGHAN 2012 NERD, they are (1) Feature and Keyword based algorithm (Wang’s algorithm)\(^25\), Decision Tree based category algorithm (Lin’s algorithm)\(^26\) and Support Vector Space algorithm (Fan’s algorithm)\(^27\). All the three algorithms achieve higher experimental results on the same dataset. Experimental results shown in Figure 6 show that a relative higher precision rate of our approach. Compared with other three algorithms, the F scores of our approach improves more than 4\%., 5.51\% and near to 9.8\% respectively.

Figure 6. Experimental results on CIPS-SIGHAN dataset.

C. Third Experiment on Three Categories of Person Name

To test the disambiguation effectiveness of three personal name categories, we select 62 person names, cover the first category (such as Mao zhedong, Zhou enlai), the second category (e.g., Liang sicheng, Li kaifu), the third category (such as Yang chen, Zhou rong, Zhang Men). For all 62 person names, we automatically construct more than 1,800 instances from Baidu celebrity, and we collect more than 400 test texts from Baike, Wikipedia and Yahoo for all 62 names. Each name relates to from 10 to 60 texts. The performance of the third experiment is shown in Figure 7.

Figure 7. Experimental results on three personal name categories.

Figure 7 shows that our approach gets good performance in the three categories of person name disambiguation. For the first category, both precision score (large than 80\%) and recall score (large than 88\%) are highest. For the second category, both precision and recall scores are a bit lower than the first category. For the third category, the precision are lower than that for the first type while the precision is relatively higher, near to 0.75\%.

In above three experiments, we set various values to the weight \(\lambda\) using the following tactics: If the extracted personal features from texts are few, that is to say, comparing similarity of personal features between two instances is less important, hence the influence of the concept-level similarity is higher, we set \(\lambda = 0.8\); If the extracted personal features from documents are more, which means the influence of the
property-level similarity is higher, thus we set a lower value to $\lambda$, such as let $\lambda = 0.2$.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we designed OnPerDis, a novel ontology based approach for Chinese personal name disambiguation on the Web. OnPerDis first designs a rich person ontology PO, including comprehensive features and rich set of person instances from Chinese Wikipedia. This PO can be used as a rich knowledge base in practice. Two personal features extraction methods were defined: one for infobox, and the other for text. A recursive evaluation of ontology instances similarity algorithm was designed to compare person instances. In addition, our approach also shows its efficacy in identifying a person with various names.

In future, we will enlarge OnPerDis’ knowledge base with more instances of the person ontology. We will also explore the mapping relationships between English names and associated Chinese names, and study more generic personal name disambiguation across several languages.

VIII. ACKNOWLEDGEMENT

This work is sponsored by the grant from the Shanghai Science and Technology Foundation (No. 11511504000).

REFERENCES


