Discovering Missing Semantic Relations Between Entities in Wikipedia

Mengling Xu[1], *Zhichun Wang[1], Rongfang Bie[1], Juanzi Li[2], Chen Zheng[1], Wantian Ke[1], and Mingquan Zhou[1]

[1] Beijing Normal University, Beijing, China
{zcwang,rfbie,mqzhou}@bnu.edu.cn
{mengling,zc_cheney,kewantian}@mail.bnu.edu.cn
[2] Tsinghua University, Beijing, China
lijuanzi@tsinghua.edu.cn

Abstract. Wikipedia's infoboxes contain rich structured information of various entities, which have been explored by the DBpedia project to generate large scale Linked Data sets. Among all the infobox attributes, those attributes having hyperlinks in its values identify semantic relations between entities, which are important for creating RDF links between DBpedia's instances. However, quite a few hyperlinks have not been anotated by editors in infoboxes, which causes lots of relations between entities being missing in Wikipedia. In this paper, we propose an approach for automatically discovering the missing entity links in Wikipedia's infoboxes, so that the missing semantic relations between entities can be established. Our approach first identifies entity mentions in the given infoboxes, and then computes several features to estimate the possibilities that a given attribute value might link to a candidate entity. A learning model is used to obtain the weights of different features, and predict the destination entity for each attribute value. We evaluated our approach on the English Wikipedia data, the experimental results show that our approach can effectively find the missing relations between entities, and it significantly outperforms the baseline methods in terms of both precision and recall.

Keywords: Wikipedia, Infobox, Linked Data

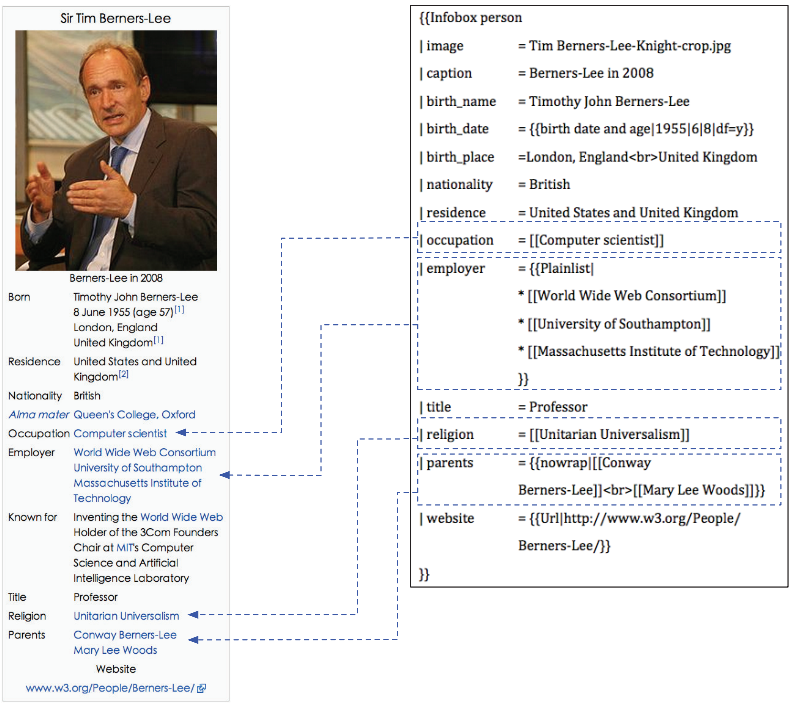
1 Introduction

Wikipedia is a free, collaborative, online encyclopedia that contains more than 20 million articles written in 285 languages by March 2013. Wikipedia articles contain rich structured information, such as infoboxes, categorization information, and links to external Web pages. Therefore, a number of projects have acquired data from Wikipedia to build large-scale machine readable knowledge bases [2, 1, 16, 3]. One of the most valuable contents in Wikipedia is its infoboxes, which display articles' most important facts as a table of attribute-value pairs,

* Corresponding author

and can be easily converted into machine-readable data. It was reported that DBpedia generated over 26 million RDF triples out of Wikipedia's infoboxes in 2009 by its generic infobox extraction algorithm. With the development of the DBpedia project these years, much more infobox RDF triples in 111 different languages have been generated.

Wikipedia uses infobox templates to define the schemas of infoboxes for different types of entities. An infobox template provides important attributes that are commonly used to describe related entities. Some attributes in infobox templates are relational that their values usually contain links referring to other entities within Wikipedia, which identify semantic relations between entities. Such relational attributes can be transformed into object properties in Linked Data, which facilitate establishing typed links between instances. Since creating links of structured data on the Web is the central idea of Linked Data, relational attributes in Wikipedia are especially important for creating Linked Data. However, sometimes the relational attributes cannot really connect entities in Wikipedia because the hyperlinks from the attributes' values to the corresponding entities are not annotated by editors. This problem causes lots of valuable relations between entities being missing in Wikipedia.



Sir Tim Berners-Lee

Berners-Lee in 2008

Born	Timothy John Berners-Lee 8 June 1955 (age 57)[1] London, England United Kingdom[1]
Residence	United States and United Kingdom[2]
Nationality	British
Alma mater	Queen's College, Oxford
Occupation	Computer scientist
Employer	World Wide Web Consortium University of Southampton Massachusetts Institute of Technology
Known for	Inventing the World Wide Web Holder of the 3Com Founders Chair at MIT's Computer Science and Artificial Intelligence Laboratory
Title	Professor
Religion	Unitarian Universalism
Parents	Conway Berners-Lee Mary Lee Woods

Website

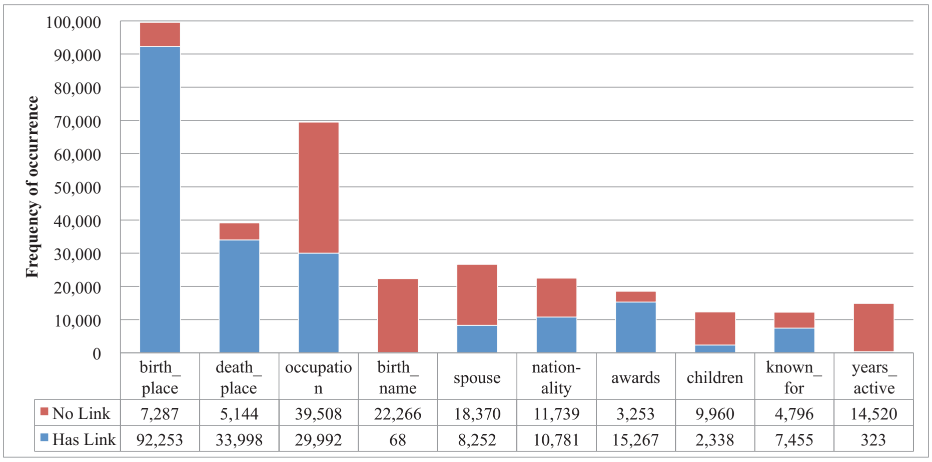
www.w3.org/People/Berners-Lee/

```
{{Infobox person
| image          = Tim Berners-Lee-Knight-crop.jpg
| caption        = Berners-Lee in 2008
| birth_name     = Timothy John Berners-Lee
| birth_date     = {{birth date and age|1955|6|8|df=y}}
| birth_place    =London, England<br>United Kingdom
| nationality    = British
| residence      = United States and United Kingdom
| occupation     = [[Computer scientist]]
| employer       = {{Plainlist|
                 * [[World Wide Web Consortium]]
                 * [[University of Southampton]]
                 * [[Massachusetts Institute of Technology]]
                 }}
| title          = Professor
| religion       = [[Unitarian Universalism]]
| parents        = {{nowrap|[[Conway
                 Berners-Lee]]<br>[[Mary Lee Woods]]}}
| website        = {{Url|http://www.w3.org/People/
                 Berners-Lee/}}
}}
```

(a) (b)

Fig. 1. Sample Wikipedia infobox: (a) display format; (b) editing format

Fig. 1 (a) and (b) show a sample infobox and its source data in editing format from the article *Tim Berners-Lee* in Wikipedia, respectively. Relational attributes such as *Occupation* and *Parents* have values with links to other entities in Wikipedia; for these attributes, we use arrow lines to connect the corresponding contents in Fig. 1 (a) and (b). The attributes *Born_place*, *Nationality* and *Residence* are supposed to be relational, but there are no links in their values. This problem does not only occur in this sample infobox. In order to get insight into the entity links in the infoboxes, we investigate all the 123,246 English person infoboxes and 1,162 Chinese person infoboxes in Wikipedia. Fig. 2 and Fig. 3 show the number of times that the value has a link and has no link of the top ten frequently used attributes in English and Chinese, respectively. In Fig. 2, it is observed that most of the top used attributes (except for the attribute *birth_name* and attribute *years_active*) can be considered as relational ones because they contain large number of links in their values; however, there are still parts of their values having no links. The percentage of values without links varies among different attributes, which ranges form 7% (*birth_place* to 81% (*children*). Similar observations can also be obtained in Chinese infoboxes; Fig. 3 shows that larger portion of attribute values have no links comparing to English infoboxes.



	birth_ place	death_ place	occupatio n	birth_ name	spouse	nation- ality	awards	children	known_ for	years_ active
🟥 No Link	7,287	5,144	39,508	22,266	18,370	11,739	3,253	9,960	4,796	14,520
🟦 Has Link	92,253	33,998	29,992	68	8,252	10,781	15,267	2,338	7,455	323

Fig. 2. Statistics of links in person infoboxes in English Wikipedia

In order to solve the problem of missing semantic relations in Wikipedia, we need a system that can automatically add entity links in the attribute values in infoboxes. Recently, several approaches have been proposed to link entities in plain texts with Wikipedia [9, 11, 7, 15]. These approaches first identify important named entities in the given text and then link them to the corresponding entities in Wikipedia. Since infoboxes contain structured information and are quite different from plain texts, traditional entity linking approaches cannot guarantee

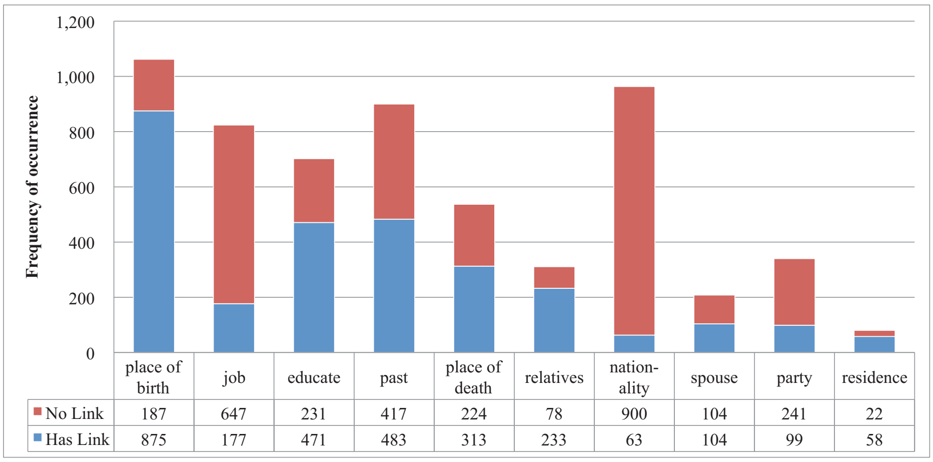


Fig. 3. Statistics of links in person infoboxes in Chinese Wikipedia

good results. Therefore, we propose an approach to automatically add entity links in infobox attribute values. Our approach first identifies all the entity mentions in a given infobox, and then decides the entity links based on 7 features of mention-entity pair. A learning model is used to obtain the appropriate weights of features, so that entity links can be predicted accurately.

The rest of this paper is organized as follows, Section 2 describes the proposed approach in detail; Section 3 presents the evaluation results; Section 4 discusses some related work and finally Section 5 concludes this work.

2 The Proposed Approach

In this section, we introduce our proposed approach in detail. Given an infobox with some missing entity links in their attribute values, our approach first automatically extracts the candidate name mentions that might refer to entities in Wikipedia, and then identifies the correct corresponding entity for each mention.

2.1 Mention Identification

To extract entity mentions in infoboxes, we build a mention dictionary that includes all the entity mentions in Wikipedia. In Wikipedia, an entity link is annotated by square brackets [[**entity**]] in the source data of articles. Here **entity** denotes the unique name of the referred entity. When the mentioned name of an entity is different from its unique name, the link is annotated by [[**entity** | **mention**]]; **mention** denotes the string tokens that actually appear in the text. In order to get all the mentions that have appeared in Wikipedia, we process all the annotated entity links in the form of [[**entity** | **mention**]] in Wikipedia. In addition, all the titles of articles in Wikipedia are also taken as mentions, which will be included in the mention dictionary. The mention dictionary also records the possible entities that each mention might refer to. Therefore, the dictionary can be represented as 2-tuple $D = (M, E)$, where $M = \{m_1, m_2, ..., m_k\}$ is the set of all mentions in Wikipedia, and $E = \{E_{m_1}, E_{m_2}, ..., E_{m_k}\}$ is the sets of entities corresponding to the mentions in M.

After the dictionary D being built, our approach extracts mentions in infoboxes by matching all the n-grams of the attribute values with mentions M in the dictionary D. The result of mention identification is a set of mentions that are matched by the n-grams. Because the goal of our approach is to find the missing entity links in infoboxes, only attribute values having no links will be processed to identify entity mentions.

2.2 Features for Predicting Entity Links

Once a set of mentions are identified in an infobox, our approach computes 7 features for each mention-entity pair to assess the possibility that a link exists between them from different respects.

Before defining other features, we first introduce a metric *Semantic Relatedness* [10]. This metric is used to compute the relatedness between the candidate entity and the context of a mention in different aspects.

Definition 1. ***Semantic Relatedness.*** *Given two entities a and b in Wikipedia, the Semantic Relatedness between a and b is computed as*

$$r(a,b) = 1 - \frac{log(max(|I_a|, |I_b|)) - log(|I_a \cap I_b|)}{log(|W|) - log(min(|I_a|, |I_b|))} \tag{1}$$

where I_a and I_b are the sets of inlinks of article a and article b, respectively; and W is the set of all articles in the input wiki.

Let B be a set of entities in Wikipedia, the Semantic Relatedness between an entity a and a set of entities B is defined as

$$SR(a,B) = \frac{1}{|B|} \sum_{b \in B} r(a,b) \tag{2}$$

Given a mention m in an infobox, let E_m represent the set of candidate entities that m might link to. For each entity $e \in E_m$, the following features are computed for (e, m).

Feature 1: *Entity Occurrence*

According to the introduction of infobox given by Wikipedia, the information presented in the infobox should still be presented in the main text of the article. Therefore, if there is already a link to a certain entity in the text of article, there will be very likely a link to this entity in the infobox. Here, we define an *Entity Occurrence* feature to capture this information:

$$f_1(e,m) = \begin{cases} 1 \text{ if } e \in C_{article}(m) \\ 0 \text{ otherwise} \end{cases} \tag{3}$$

where $C_{article}(m)$ is the set of entities appearing in the main text of the current article containing m.

Feature 2: *Link Probability*

Link Probability feature approximates the probability that a mention m links to an entity e:

$$f_2(e,m) = \frac{count(m,e)}{count(m)} \tag{4}$$

where $count(m,e)$ denotes the number of times that m links to e in the whole Wikipedia, and the $count(m)$ denotes the number of times that m appears in Wikipedia.

Feature 3: *Infobox Context Relatedness*

Let $C_{infobox}(m)$ be the set of entities already be linked in the infobox where m appear, we define the infobox context relatedness between a candidate entity $e \in E_m$ and a mention m as

$$f_3(e,m) = SR(e, C_{infobox}(m)) \tag{5}$$

Feature 4: *Article Context Relatedness*

Let $C_{article}(m)$ be the set of entities already linked by mentions in the article text that m appear, we define the article context relatedness between a candidate entity $e \in E_m$ and mention m as

$$f_4(e,m) = SR(e, C_{article}(m)) \tag{6}$$

Feature 5: *Abstract Context Relatedness*

The first paragraph in the text of an article usually defines the subject of the article, and contains the most important information about the subject of the article, which is usually called the abstract or the definition of the article. Let $C_{abstract}(m)$ be the entities appear in the abstract, here we define the abstract context relatedness between a candidate entity $e \in E_m$ and a mention m as

$$f_5(e,m) = SR(e, C_{abstract}(m)) \tag{7}$$

Feature 6: *Attribute Range Context Relatedness*

Let $C_{att\_rang}(m)$ be the set of entities that appear in the value of attribute att_m, we define the attribute value context relatedness between a candidate entity $e \in E_m$ and mention m as

$$f_6(e,m) = SR(e, C_{att\_rang}(m)) \tag{8}$$

Attribute Range Context Relatedness can assess the similarity between a candidate entity and the set of entities that have already been linked in the value of a concerned attribute. Therefore, this feature can estimate what types of entities are more likely to be linked by the concerned attribute.

Feature 7: *Attribute Domain Context Relatedness*

Let $C_{att\_dom}(m)$ be the set of entities that described by the attribute att_m, we define the attribute domain context relatedness between a candidate entity $e \in E_m$ and mention m as

$$f_7(e,m) = SR(e, C_{att\_dom}(m)) \tag{9}$$

2.3 Learning to Predict New Entity Links

To predict new entity links, our approach computes the weighted sum of features between mentions and entities by the following score function:

$$s(m,e) = \omega_1 \times f_1(m,e) + ... + \omega_6 \times f_6(m,e) + \omega_7 \times f_7(m,e) \tag{10}$$

For each mention m, the entity e^* that maximizes the score function $s(m,e^*)$ is predicted as the destination entity of m. The idea of predicting entity links is simple and straight, but how to appropriately set the weights of different similarity features is a challenging problem, which highly influences the final results.

Here, we use the already existing entity links $L = \{< m_i, e_i >\}_{i=1}^{k}$ in infoboxes as training data, and train a logistic regression model to get the weights of different features. Given a mention m and its corresponding entity e, the learned weights should ensure

$$\boldsymbol{\omega} \cdot (\boldsymbol{f}(m,e^*) - \boldsymbol{f}(m,e)) > 0, (e \in E_m, e \neq e^*) \tag{11}$$

where $\boldsymbol{\omega} =< \omega_1, ..., \omega_7 >$ and $\boldsymbol{f}(\cdot) =< f_1(\cdot), ..., f_7(\cdot) >$.

Therefore, we can use the sigmoid function to compute the probability that an entity e_1 is better than another entity e_2 (denoted as $e_1 \succ e_2$) as the destination for a mention.

$$P((e_1 \succ e_2) = true) = \frac{1}{1 + e^{-\boldsymbol{\omega} \cdot (\boldsymbol{f}(m,e_1) - \boldsymbol{f}(m,e_2))}} \tag{12}$$

If $s(m,e_1) > s(m,e_2)$, $P((e_1 \succ e_2) = true) > 0.5$; otherwise $P((e_1 \succ e_2) = true) < 0.5$. In this case, the weights $\boldsymbol{\omega}$ can be determined by the MLE (maximum likelihood estimation) technique for logistic regression.

Therefore, we generate a new dataset $D = \{(\boldsymbol{x_j}, y_j)\}_{j=1}^{m}$ based on the known entity links $L = \{< m_i, e_i >\}_{i=1}^{k}$ to train a logistic regression model; $\boldsymbol{x_j}$ is the input vector and y_j represents the class label (positive or negative). For each mention m_i, a positive example $(\boldsymbol{f}(m_i, e_i) - \boldsymbol{f}(m_i, e^{'}), positive)$ or a negative example $(\boldsymbol{f}(m_i, e^{'}) - \boldsymbol{f}(m_i, e_i), negative)$ is generated for each entity $e^{'} \in (E_{m_i} - \{e_i\})$. We make the number of positive examples and negative examples be the same, which avoids the imbalanced classification problem. After the logistic regression model being trained, the learned weights $\boldsymbol{\omega} =< \omega_1, ..., \omega_7 >$ will be used in Equation 10 to predict new entity links.

For some identified mentions, there might not be its corresponding entities in Wikipedia. Therefore, a threshold δ is set to filter out entity links with low scores. In the learning process, when the optimal weights of features are obtained, the threshold δ is determined by optimizing the overall performance on the training dataset.

3 Experiment

3.1 Datasets

We use the datasets of English Wikipedia to evaluate the proposed approach. We downloaded the English Wikipedia XML dump from Wikipedia's download site[3], which was archived in August 2012, and has 4 million articles. 100 infoboxes are randomly chosen from the whole dataset for the evaluation. There are 630 already existing entity links in the selected infoboxes, 50% of these links are randomly selected as the ground truth for the evaluation, which are removed from the infoboxes before the infoboxes are fed to our approach. After the execution of our approach, we collect the new discovered entity links and compare them against the ground truth links. In the experiments, 40% of the selected ground truth links were used for training the prediction model, the rest of 60% selected links were used as testing data in the evaluation.

3.2 Evaluation Metrics

We use precision, recall, and F1-score to evaluate the performance of the proposed approach. These measures are computed as follows:

Precision (p): It is the percentage of correctly discovered entity links in all the discovered entity links.

$$p = \frac{|A \cap T|}{|A|} \tag{13}$$

where T is the set of ground truth entity links, A is the set of discovered entity links.

Recall (r): It is the percentage of correctly discovered entity links in the ground truth entity links.

$$r = \frac{|A \cap T|}{|T|} \tag{14}$$

F1-score ($F1$): F1-Measure considers the overall result of precision and recall.

$$F1 = \frac{2pr}{p + r} \tag{15}$$

3.3 Comparison Methods

Here we use three comparison methods as the baselines of evaluation:

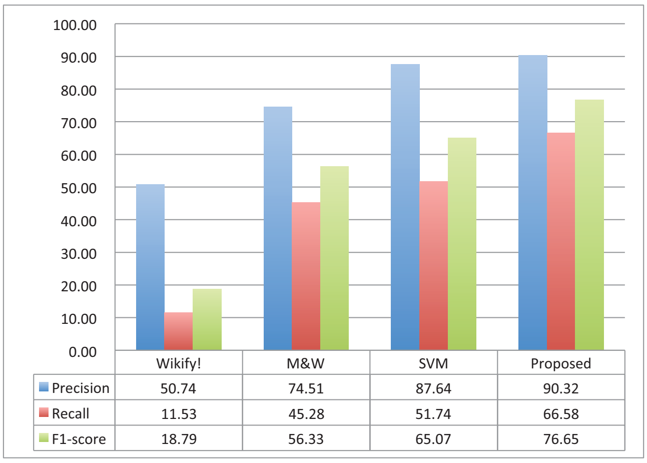
- **Wikify!**. This method was proposed by Mihalcea and Csomai [9], which is able to automatically perform the annotation task following the Wikipedia guidelines. Wikify! first uses a unsupervised extraction algorithm to identify and rank mentions, and then combines both knowledge-based approach and data-driven method to discover new entity links.

[3] http://dumps.wikimedia.org/enwiki/

- **M&W**. Milne and Witten proposed an learning based entity linking approach [11]. Their approach uses three features (Commonness, Relatedness, and Context Quality) and C4.5 classifier to predict new entity links. Here, we first use our approach to identify mentions in the infoboxes, and then employ Milne and Witten's disambiguation method to predict new entity links.
- **SVM**. This method first computes the similarities defined in Section 2.2 for each mention-entity pair, and then trains a SVM [4] classification model on the training entity links. New mention-entity pairs are predicted by the trained SVM as entity links or not entity links.

3.4 Results Analysis

Here we first compare the performance of our approach with the comparison methods, and then analysis the contribution of different features in our approach.



	Wikify!	M&W	SVM	Proposed
🟦 Precision	50.74	74.51	87.64	90.32
🟥 Recall	11.53	45.28	51.74	66.58
🟩 F1-score	18.79	56.33	65.07	76.65

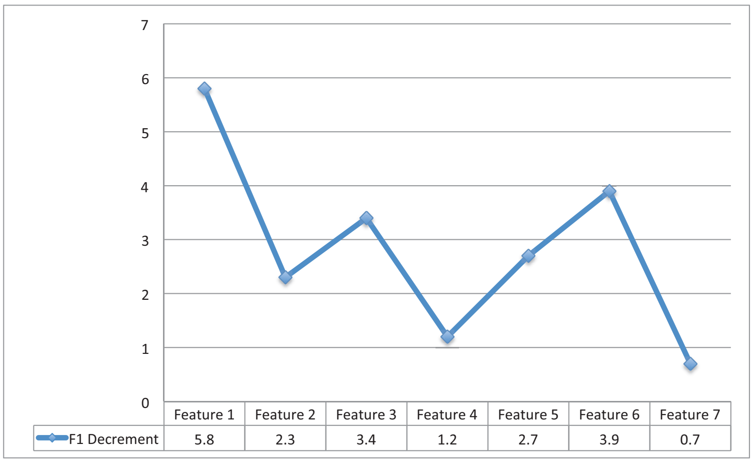
Fig. 4. Performance of different methods (%)

Performance Comparison Fig. 4 shows the performance of 4 different methods. According to the results, the Wikify! method does not perform very well on the infobox data. Wikify! only achieves 50.74% precision and 11.53% recall. It seems that Wikify! can not make good decision given only the string tokens of a infobox. The method of M&W performs better than Wikify!, but the SVM method achieves both better precision and recall than M&W. Therefore, it shows that the features defined for our approach have better discriminant ability than the features in M&W method. Compared with three baseline methods, our proposed approach achieves the best results in terms of both precision and recall.

Our proposed approach outperforms SVM method by 11.58% in terms of F1-score, which means that the learning method in our approach is more suitable for the entity linking tasks; training classifiers directly on the original features cannot get the best performance.

Feature Contribution Analysis Among 7 defined features, which one is the most important? To get insight to this question, we perform an analysis on the contribution of different features. Here, we run our approach 7 times on the evaluation data. Each time one feature is removed from the feature vectors of mention-entity pairs. We record the decrease of F1-score for each feature when it is removed; it is reasonable to evaluate the importance of each feature by comparing their corresponding F1-score decrease. Fig. 5 compares the importance of different features. According to the results, we can rank these features based on their importance in a descending order as: Feature 1, Feature 6, Feature 3, Feature 5, Feature 2 and Feature 7.

It seems that the occurrence of candidate entities in the main text of article is very important for identifying the correct entity links. The *Attribute Range Context Relatedness* feature is also important, it might because this feature can reflect what types of entities are possible to appear in the values of certain attributes. The *Attribute Domain Context Relatedness* feature is the least important one among all the 7 features, it might because different entities usually have different values of the same attribute; the relatedness between candidate entities and the entities described by a specific attribute is less relevant to the entity links.



	Feature 1	Feature 2	Feature 3	Feature 4	Feature 5	Feature 6	Feature 7
F1 Decrement	5.8	2.3	3.4	1.2	2.7	3.9	0.7

Fig. 5. Contribution analysis of different features (%)

4 Related Work

In this section, we review some related work.

4.1 Entity Linking

A group of closely related work is *Entity Linking*, which aims to identify entities in documents and link them to a knowledge base, such as Wikipedia and DBpedia.

Wikify! [9] is a system which is able to automatically perform the annotation task following the Wikipedia guidelines. Wikify! has two components: the keyword extraction and the link disambiguation. In the first components, Wikify! uses a unsupervised keyword extraction algorithm to identify and rank mentions. In the disambiguation component, Wikify! combines both knowledge-based approach and data-driven method to predict the links from mentions to entities in Wikipedia.

Milne et al. [11] proposed a learning based approach for linking entities in text to Wikipedia. Their approach trains a C4.5 classifier based on three features (commonness, relatedness and context quality) of entity-mention pairs for link disambiguation. A classification algorithm is also used in the candidate link detection.

Kaulkarni et al. [7] proposed a collective approach for annotating Wikipedia entities in Web text. Their approach differs from the former approaches in that it combines both local mention to entity compatibility and global document level topical coherence. The collective prediction of entity links improves the accuracy of results.

Following a similar collective decision idea, Han et al. [6] proposed a graph-based collective entity linking algorithm. Their approach first construct a referent graph, where nodes corresponds to all name mentions in a document and all possible referent entities of these name mentions, edge between a name mention and an entity represents a compatible relation between them, edge between two entities represents a sematic-related relation between them. Both the compatibility and semantic relatedness are propagate through the referent graph. The entity linking problem is solved by selecting the entity for a mention that maximizes the product of compatibility and relatedness.

Mendes et al. [8] developed a system DBpedia Spotlight for automatically annotating text documents with DBpedia URIs. DBpedia Spotlight first recognizes the phrases in a sentence that may indicate a mention of a DBpedia entity; then the recognized mention is mapped to candidate entities in DBpedia; a disambiguation stage is employed to find the most likely entities for the mention. The disambiguation task is cast as a ranking problem in DBpdia Spotlight, and Vector Space Model and a new weighting method Inverse Candidate Frequencty (ICF) are used for similarity computation.

Shen et al. [15] proposed a system LINDEN, which is a novel framework to link named entities in text with a knowledge base by leveraging the rich semantic knowledge embedded in the Wikipedia and the taxonomy of the knowledge base.

The LINDEN builds a feature vector for each entity, which includes link probability, semantic associativity, semantic similarity and global coherence. And the system uses a max-margin technique to rank the candidate entities for the entity mentions.

LIEGE [14] is another work of Shen et al., a general framework for linking entities in web lists with knowledge base. In order to find the proper entity from knowledge base as the mapping entity for the list item, LIEGE defines several metrics to measure the link quality of the candidate mapping entity, including the prior probability, coherence, type hierarchy based similarity, and distributional context similarity. A max-margin technique is used to learn the weights for different feature values to calculate the linking quality.

The above entity linking approaches mainly take plain texts as inputs, and the infoboxes are quite different from plain texts. Information in infoboxes is structured, and some existing entity links might appear in infoboxes. Based on these observations, we define more specific features to describe the relations between mentions and entities. What's more, we use a new learning method to get the weights of different features. Because there are lots of context entity links for each mention in infobox, we can still get desired results when entity links are predicted not in a collective way.

4.2 Instance Matching

Another group of related work is *Instance Matching*, which aims to find equivalent entities in different linked datasets. Instance matching tools can be used to find new RDF links between linked datasets.

Silk [17] is a link discovery engine which automatically finds RDF links between data sets. Users must specify which type of RDF links should be discovered between the data sources as well as which conditions data items must fulfill in order to be interlinked. These link conditions can apply different similarity metrics to multiple properties of an entity or related entities that are addressed using a path-based selector language.

idMesh [5] is a graph-based algorithm for online entity disambiguation based on a probabilistic graph analysis of declarative links relating pairs of entities. idMesh derives a factor-graph from the entity and the source graphs to retrieve equivalent entities.

Raimond et al.[13] propose a interlinking algorithm for automatically linking music-related data sets on the web, taking into account both the similarities of the web resources and of their neighbors. Their algorithm provides online linking function based on accessing data through SPARQL end-points.

Nikolov et al. [12] present a data integration architecture called KnoFuss and proposed a component-based approach, which allows flexible selection and tuning of methods and takes the ontological schemata into account to improve the reusability of methods.

The purpose of our approach is to add semantic relations between entities in Wikipedia, which will finally enrich RDF links in DBpedia. Although *Instance Matching* also can add RDF links between different datasets, it mainly focuses on discovering the *sameAs* links, which is different from finding arbitrary typed relations between entities.

5 Conclusion and Future Work

In this paper, we propose an approach for automatically discovering the missing typed relations between entities in Wikipedia's infoboxes. Our approach works in two steps: it first identifies entity mentions in the given infoboxes, and uses a learning model to predict new entity links based on several features of mention-entity pair. The experimental results show that our approach can accurately find missing links in infoboxes, and it performs better than the baseline methods.

Actually, there are some wrongly annotated entity links in Wikipedia's infoboxes. Besides of adding new entity links in infoboxes, we also want to discover wrong entity links in infoboxes in the future work. By the efforts of adding and refining entity links in infoboxes, more semantic relations of high quality between entities can be obtained. Our future work also includes extending our approach to solve the problem of finding missing RDF links between linked datasets.

6 Acknowledgement

The work is supported by NSFC (No. 61202246, 61035004, 61171014), NSFC-ANR(No. 61261130588), 863 High Technology Program (2011AA01A207), FP7-288342, THU-NUS NExT Co-Lab, and the Fundamental Research Funds for the Central Universities.

References

1. S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. G. Ives. DBpedia: A nucleus for a web of open data. In *Proceedings of the 6th international semantic web conference*, pages 722–735, 2007.
2. C. Bizer, J. Lehmann, G. Kobilarov, S. Auer, C. Becker, R. Cyganiak, and S. Hellmann. DBpedia - a crystallization point for the web of data. *Web Semantics: Science, Services and Agents on the World Wide Web*, 7(3):154 – 165, 2009.
3. K. D. Bollacker, R. P. Cook, and P. Tufts. Freebase: a shared database of structured general human knowledge. In *Proceedings of the 22nd national conference on Artificial intelligence*, volume 2, pages 1962–1963, 2007.
4. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 9 1995.
5. P. Cudre-Mauroux, P. Haghani, M. Jost, K. Aberer, and H. De Meer. idMesh: graph-based disambiguation of linked data. In *Proceedings of the 18th international conference on World Wide Web*, pages 591–600, 2009.
6. X. Han, L. Sun, and J. Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774, 2011.

7. S. Kulkarni, A. Singh, G. Ramakrishnan, and S. Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466, 2009.
8. P. N. Mendes, M. Jakob, A. García-Silva, and C. Bizer. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8, 2011.
9. R. Mihalcea and A. Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242, 2007.
10. D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In *Proceedings of the first AAAI Workshop on Wikipedia and Artificial Intelligence*, 2008.
11. D. Milne and I. H. Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, 2008.
12. A. Nikolov, V. S. Uren, E. Motta, and A. N. D. Roeck. Handling instance coreferencing in the knofuss architecture. In *1st international workshop on Identity and Reference on the Semantic Web*, 2008.
13. Y. Raimond, C. Sutton, and M. Sandler. Automatic interlinking of music datasets on the semantic web. In *Proceedings of the1st Linked Data on the Web Workshop.*
14. W. Shen, J. Wang, P. Luo, and M. Wang. LIEGE: link entities in web lists with knowledge base. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1424–1432, 2012.
15. W. Shen, J. Wang, P. Luo, and M. Wang. LINDEN: linking named entities with knowledge base via semantic knowledge. In *Proceedings of the 21st international conference on World Wide Web*, pages 449–458, 2012.
16. F. M. Suchanek, G. Kasneci, and G. Weikum. YAGO: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.
17. J. Volz, C. Bizer, M. Gaedke, and G. Kobilarov. Discovering and maintaining links on the web of data. In *Proceedings of the 8th International Semantic Web Conference*, pages 650–665, 2009.