LODDO: Using Linked Open Data Description Overlap to Measure Semantic Relatedness Between Named Entities

Wenlei Zhou, Haofen Wang, Jiansong Chao,
Weinan Zhang, and Yong Yu

APEX Data & Knowledge Management Lab
Department of Computer Science Engineering
Shanghai Jiao Tong University, Shanghai, China
{wenlei.zhouwl,whfcarter,jiansong.chao,
wnzhang,yyu}@apex.sjtu.edu.cn

Abstract. Measuring semantic relatedness plays an important role in information retrieval and Natural Language Processing. However, little attention has been paid to measuring semantic relatedness between named entities, which is also very significant. As the existing knowledge based approaches have the entity coverage issue and the statistical based approaches have unreliable result to low frequent entities, we propose a more comprehensive approach by leveraging Linked Open Data (LOD) to solve these problems. LOD consists of lots of data sources from different domains and provides rich a priori knowledge about the entities in the world. By exploiting the semantic associations in LOD, we propose a novel algorithm, called LODDO, to measure the semantic relatedness between named entities. The experimental results show the high performance and robustness of our approach.

Keywords: Named Entity, Semantic Relatedness, Linked Open Data

1 Introduction

Semantic relatedness measuring plays an important role in the area of natural language processing (e.g., word sense disambiguation [14]) and information retrieval. With the advance of Semantic Web, more and more documents are annotated with real world entities. Hence, measuring semantic relatedness between these named entities can be regarded as an effective mean to capture semantic associations between documents, which can be further used for semantic search.

In recent years, there are abundant research studies on measuring semantic relatedness between words. They tried to solve the following two challenges:

- Word Ambiguity. A word might refer to different meanings or can represent different entities.
- Different Representations of a Single Entity. Even for a unique entity, it may have different representations, which requires us to collect all synonyms of a given word.

The existing work can be divided into two types: knowledge based approaches and statistical based approaches. The former ones basically leverage a high-quality knowledge source like WordNet [12] or Wikipedia[1]. The main limitation of this kind of work is the coverage issue. While Wikipedia is the world largest domain independent knowledge base, it misses a number of entities in some specific domain. On the other hand, statistical based approaches mainly exploit the Web for this task. However, they fail to provide reliable semantic relatedness between words of low frequencies.

In this paper, we propose a novel approach to overcome the previous problems by leveraging Linked Open Data [1] (LOD). LOD is an abundant Web of data which contains a vast number of named entities. It is constructed by linking diverse data sources. While the openness of the Web might involve data noise, we assume LOD as high-quality data sources since they are published from existing structural or qualified databases. As the data sources cover many domains, given a named entity, it is highly possible that there is some description about it in LOD. Thus entity coverage problem can be eased by using LOD. On the other hand, while the statistical based approaches regard named entities which have the same name in all documents as the same entity, LOD represents them as different entities. As a result, each entity in LOD has its own description and it is distinguished from other entities of the same name.

The contributions of this paper are threefold. First, we build an efficient LOD index mechanism to solve the two challenges: word ambiguity and different representations of a single entity. Second, we propose a novel approach LODDO to accurately measure the semantic relatedness between named entities by exploiting the semantic associations in LOD. Third, the experiments result shows that our approach outperforms the existing semantic relatedness measuring approaches by at least 39.6%.

The remainder of the paper is organized as follows. In section 2 we discuss previous work related to named entities semantic relatedness measuring. The methodology is presented in section 3. The conducted experiments and the benchmark dataset with the evaluation results are presented in section 4. In section 5 we conclude the paper and discuss the future work.

2 Related Work

The existing semantic relatedness measuring approaches can be grouped into two types according to the sources they use: knowledge based approaches and statistical based approaches. The knowledge based approaches take advantage of a high-quality knowledge source such as WordNet, Roget or Wikipedia. The statistical based approaches calculate the statistical information of words by using Web corpus as their source.

Regarding the knowledge source as a graph of concepts connected with others, a straightforward approach to calculate semantic relatedness between two

[1] http://www.wikipedia.org/

words (concepts) is to find the length of the path connecting the two words in the graph [17, 10, 9, 22]. Based on the intuition that the relatedness of two words (concepts) can be measured by the amount of information they share, Strube and Ponzetto [20, 16] applied intrinsic information content to Wikipedia category graph. Resnik [18] used information content based on WordNet to measure semantic similarity. Hypothesizing that the higher word overlap in two concepts' glosses, the stronger semantic relatedness of these two concepts, Lesk [11] and Banerjee [3] introduced a measure based on the amount of word overlap in the glosses of two concepts. Strube [20] regarded the first paragraph of the concept's Wikipedia article as the concept's glosses. Patwardhan [15] calculated the cosine of the second-order gloss vectors which represented the corresponding words by using WordNet glosses. Gabrilovich [7] introduced ESA which constructed concept vectors from Wikipedia articles where each vector element represented an article. Milne [13] constructed the vectors by using the interlink articles.

For abstract concepts semantic relatedness measuring, single domain independent knowledge source may be enough to cover all the concepts. However, when dealing with hundreds of millions named entities in our real life, the coverage problem arises. Research [23] has also shown that the accuracy differs depending on the choice of the knowledge sources, and there is no conclusion which knowledge source is superior to others. It seems that different knowledge source may have its own preference in describing data, and thus it is unreliable to just use single knowledge source when measuring semantic relatedness.

The statistical based approaches calculate the statistical information of words by using Web corpus as their source. Bollegala [5] used four popular co-occurrence measures to calculate page-count-based similarity metrics for the pairs of single words and automatically extracts lexico-syntactic patterns about the pairs of single words based on the title, snippet and URL of the Web search results. Spanakis [19] modified Bollegala's method by adding consideration of the "Bag of Words" representation to the Web search results text for each single word. Since a named entity usually contains more than one word, the lexico-syntactic patterns extraction cannot be used directly. Gracia [8] proposed a transformation of the Normalized Google Distance [6] into a word relatedness measure based on Web search engine.

Some shortcomings of statistical based approaches are as follows. Without the help of human knowledge, the statistical based approaches actually regard the words in all documents as the same meaning when calculating one word's statistical information. This will lead to the ineffectiveness when measuring two low frequent words' semantic relatedness. In addition, these approaches also depend on the effectiveness and efficiency of the Web search engine.

3 Methodology

In recent years, the amount of structured data available on the Web has been increasing rapidly, making it possible to propose new ways to address complex information needs by combining data from different sources. LOD is aimed to

link the existing data sources using RDF, and by September 2010, 203 data sources in different domains consisting of over 25 billion RDF triples have been added into LOD cloud. This gives us an inspiration to measure named entities semantic relatedness based on LOD. As LOD consists of lots of data sources from different domains, by leveraging LOD, the named entity coverage problem can be overcome. And it gives us a possible solution to synthesize multi-sources. While the statistical based approaches regard named entities which have the same name in all documents as the same entity, LOD represents them as different entities. So even the low frequent entity can have its own description, which can be distinguished from other entities of the same name.

Figure 1 shows the architecture of our approach LODDO, which measures named entities semantic relatedness based on LOD. There are two components in the architecture: offline and online components. The offline component is aimed to build an index from the various LOD sources which can be used to find the entities corresponding to a specific entity name fleetly. For the online component, the Description Retrieval can retrieve all the description information of a given entity name from data sources by leveraging LOD Index. The Description Overlap Measuring uses the description information of two named entities to calculate the semantic relatedness between them.

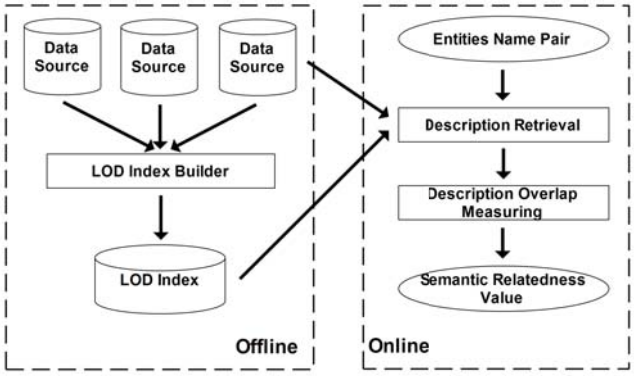

Fig. 1. The Architecture of LODDO

3.1 LOD Index Builder

LOD uses RDF, which is a generic, graph based data framework that represents information based on triples of the form (*subject*, *predicate*,*object*), to organize the data. An entity can be either a subject or an object of any RDF triple. LOD identifies an entity via a HTTP scheme based Uniform Resource Identifier (URI). The URI does not only serve as a unique global identifier but it also provides access to a structured data representation of the identified entity.

It is not trivial to find the entities which have a specific name directly. For example, two data sources, even the same data source, may represent one target entity by different uris. So it becomes very important to find all the uris which mean the same entity. Moreover, some name variants may correspond to one entity, which is ineffectively solved just by leveraging the string similarity. To solve these problems, we leverage the name properties, uri format and certain relationships in LOD to enumerate all possible name variants and uris to an entity, which can be represented as an entity triple (*entity_id*, *uri_set*, *name_set*). Here, *entity_id* is an automatic generated University Unique Identifier (UUID) of an entity.

3.1.1 Name Extraction for URI Thanks to the broad coverage of LOD, most name variants of an entity can be discovered by mining the diverse data sources. In this subsection, we focus on the name extraction for each uri. Unfortunately, different data sources may have different representation for names of an entity. A predicate may be used in different ways in different sources. For example, in RDF schema, the predicate rdfs[2]:label is defined to provide a human-readable version of a resource's name. However, DBpedia uses it in a different way. Here is an example of rdfs:label in DBpedia:

(*dbpedia*[3]*:The_World_Health_Organization*, *rdfs:label*, "*The*").

Obviously it is not right to regard "*The*" as the name. Therefore, we need to analyze the LOD data sources respectively and identify the ways that may describe the name information. In such a way, we can get all the name variants of a uri and by automatically generating unique entity id corresponding to the uri, we get the initial entity triple space Γ.

$$\Gamma = \{(entity\_id, uri, name\_set) \mid uri \in LOD\} \tag{1}$$

Here we present the name schema of several data sources: DBpedia, Musicbrainz [21] (DBtune), and Freebase [4].

- For DBpedia, we find that there's no exact *predicate* which can show the name of a DBpedia uri. As a solution, we extract the name by deleting "_" and "()" components from the tail of the uri. For example:
  dbpedia:James_Sikes has the name "*James Sikes*".
  dbpedia:Think_Again_(band) has the name "*Think Again*".

- Musicbrainz (DBtune) represents a uri's name by *predicate*: foaf[4]:name, mo[5]:title and skos[6]:altLabel.
- Freebase uses fb[7]:type.object.name as the *predicate* of a uri's name.

[2] http://www.w3.org/2000/01/rdf-schema#
[3] The dbpedia: stands for the prefix for URI from DBpedia
[4] http://xmlns.com/foaf/0.1/
[5] http://purl.org/ontology/mo/
[6] http://www.w3.org/2004/02/skos/core#
[7] http://rdf.freebase.com/ns/

3.1.2 Integrate Entity Triples We have mentioned that different data sources, even the same data source, may represent one target entity by different uris in LOD. However, there exist some relationships connecting uris which are actually telling the same entity. We have identified three such relationships and make use of them to integrate the entity triples.

DBpedia:disambiguates Relationship. Disambiguation in DBpedia is the process of resolving the conflicts that arise when a name is ambiguous—when it refers to more than one topic covered by DBpedia. A disambiguation uri is linked with other different uris which have the same name. For example, there are two disambiguation triples in DBpedia:

(*dbpedia:Bell*, *dbpedia:disambiguates*, *dbpedia:Bell_Island*)
(*dbpedia:Bell*, *dbpedia:disambiguates*, *dbpedia:Bell_Labs*)

which means *dbpedia:Bell_Island* has "*Bell*" and "*Bell Island*" as its name variants. And *dbpedia:Bell_Labs* has "*Bell*" and "*Bell Labs*" as its name variants.

Algorithm 1 Entity Triples Integration

Input: Initial entity triple (*entity_id*, *uri_set*, *name_set*) space Γ got from subsection "Name Extraction for URI"; LOD triple (*subject*, *predicate*, *object*) space Σ.
Output: Entity triple space Γ.

for all x in Σ **do**
  if x is a dbpedia:disambiguates or dbpedia:redirect triple **then**
    $et1 \leftarrow$ entity triple whose *uri_set* contains $x.subject$
    $et2 \leftarrow$ entity triple whose *uri_set* contains $x.object$
    $et2.name\_set = et1.name\_set \cup et2.name\_set$
  end if
end for
for all x in Σ **do**
  if x is a dbpedia:disambiguates or dbpedia:redirect triple **then**
    $et \leftarrow$ entity triple whose *uri_set* contains $x.subject$
    $\Gamma = \Gamma - et$
  else if x is a owl:sameAs triple **then**
    $et1 \leftarrow$ entity triple whose *uri_set* contains $x.subject$
    $et2 \leftarrow$ entity triple whose *uri_set* contains $x.object$
    $entity\_id \leftarrow UUID\_Generation()$
    $\Gamma = \Gamma \cup \{(entity\_id, et1.uri\_set \cup et2.uri\_set, et1.name\_set \cup et2.name\_set)\}$
    $\Gamma = \Gamma - et1$
    $\Gamma = \Gamma - et2$
  end if
end for

DBpedia:redirect Relationship. DBpedia may use a redirect relationship to link one uri, which has no description, to another uri which has a description. The reasons for creating and maintaining such a schema include: alternative names, alternative spellings or punctuation, abbreviations, etc [2]. If uri_1 redirects to uri_2, the uri_2 should also have the name of uri_1 as its name variant. For example, we have such a triple in DBpedia:

(*dbpedia:UK*, *dbpedia:redirect*, *dbpedia:United_Kingdom*)
which means *dbpedia:United_Kingdom* has "UK" and "United Kingdom" as its name variants.

Owl:sameAs Relationship. By common agreement, Linked Data publishers use the link type owl[8]:sameAs to state that two URI aliases refer to the same resource. Therefore, if uri_1 owl:sameAs uri_2, their entity triples should be integrated.

If two uris have an $owl : sameAs$ relationship, their uris and names will be integrated to the same *entity_id*. For $dbpedia : disambiguates$ and $dbpedia : redirects$ relationships, we just integrate their names excluding uris. The detail algorithm of Entity Triples Integration is shown in Algorithm 1. And the time complexity is $O(|\Sigma|)$.

3.1.3 Index Storage After getting all the entity triples, we need a mechanism to store and index them in order to guarantee the efficient retrieval for online semantic relatedness measuring. Considering the existence of one word's different formats, such as *apple* and *Apples* which may indicate to the same entity, we need to normalize the names at first. The rules are as follows: (1) convert the names to lowercase; (2) perform word stemming on the names; (3) remove any articles from names.

Then, the inverted list is utilized to store such information. The storage mechanism is shown in Figure 2 and corresponding notation description is in Table 1.

After Entity Triple Integration, all the name variants and uris of an entity are extracted which means that the challenge, different representations of a single entity, has been solved. By using the LOD Index, we can find all the entities of a given name, which means that the word ambiguity challenge also has been solved.

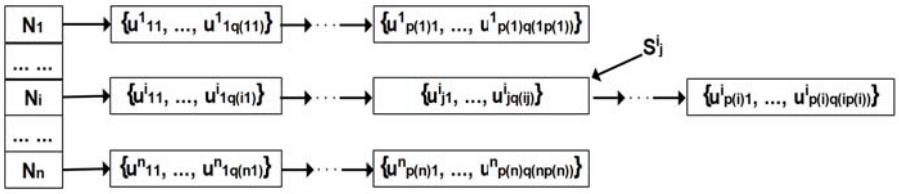

Fig. 2. LOD Index mechanism

3.2 Semantic Relatedness Measuring

Given an entity name, normalization of the name should be processed at first. Then we can retrieve all the entities with such a normalized name variant by

[8] http://www.w3.org/2002/07/owl#

Table 1. Notations for (Figure 2)

Notation	Description
N_i	name string
S_j^i	jth entity with a name variant of N_i
u_{jk}^i	kth uri which indicates to the S_j^i entity
n	the whole number of name strings
$p(i)$	number of entities with N_i as its name variant
$q(ij)$	number of uris corresponding to entity S_j^i

leveraging the LOD Index. As there is a large variety of description about an entity in LOD, the heuristics arises that the more common description two entities have, the stronger semantic relatedness they have. In the following section, we will describe Description Retrieval and Description Overlap Measuring in detail.

3.2.1 Description Retrieval Since an entity is represented as a set of uris, the description of the entity can be constructed by accumulating the description of the uris in the *uri_set*. The description of a uri is defined as a vector of subjects and objects which forms RDF triples with the uri. In a LOD triple, if uri_i is the subject, then the object should be inserted into the description of uri_i. Otherwise, if uri_i is the object, the subject should be inserted into uri_i's description. In LOD, an entity uri may have types in all probability. However, there exist some type assertions which are too loose. For example, almost every entity uri in DBpedia has a type of *owl:Thing*. So for avoiding such noise in LOD, we ignore the type assertion when generating the description.

3.2.2 Description Overlap Measuring Having the heuristics that two related named entities may have many common related things, we leverage the LOD Description Overlap, named as LODDO, to calculate the semantic relatedness between two named entities.

In the real world, there exists such a situation: entity p has many related entities including entity q which leads to a weak semantic relatedness between p and q, however q only has few related entities including p which leads to a stronger semantic relatedness. So, it becomes an issue about how to determine the final semantic relatedness between p and q. Having such a puzzle, we use the following two strategies to determine the final semantic relatedness.
(1) LODJaccard: Consider equally to both named entities when measuring the semantic relatedness. It is defined as follows:

$$\begin{aligned} CommonDescription(p,q) &= |Description(p) \cap Description(q)| \\ Denominator(p,q) &= |Description(p)| + |Description(q)| \\ &\quad - |Description(p) \cap Description(q)| \\ LODJaccard(p,q) &= \frac{CommonDescription(p,q)}{Denominator(p,q)} \end{aligned} \tag{2}$$

(2) LODOverlap: Have a bias towards the less description named entity when measuring the semantic relatedness. It is defined as follows:

$$\begin{aligned} CommonDescription(p,q) &= |Description(p) \cap Description(q)| \\ Denominator(p,q) &= min(|Description(p)|, |Description(q)|) \\ LODOverlap(p,q) &= \frac{CommonDescription(p,q)}{Denominator(p,q)} \end{aligned} \tag{3}$$

where *Description(p)* means the description of entity p.

Table 2. Four strategies to measure LOD Description Overlap

label	strategy name	description
1	LODJaccard_L	Choose LODJaccard to determine semantic relatedness. And choose largest LODJaccard to deal with multi-pairs problem.
2	LODOverlap_L	Choose LODOverlap to determine semantic relatedness. And choose largest LODOverlap to deal with multi-pairs problem.
3	LODJaccard_LC	Choose LODJaccard to determine semantic relatedness. And choose largest CommonDescription to deal with multi-pairs problem.
4	LODOverlap_LC	Choose LODOverlap to determine semantic relatedness. And choose largest CommonDescription to deal with multi-pairs problem.

As there may be several entities which have the same name variant, given two entity names, multi-pairs may be generated. So we should determine which two entities should be chosen to calculate the semantic relatedness. Because of the lack of context around the given entity names, we should choose the entities pair which is mostly in agreement with usual human sense. There are two strategies: (1) Largest LODJaccard or LODOverlap: Choose the pair which has the strongest semantic relatedness. This strategy has been adopted by many semantic relatedness measuring approaches, such as [10, 18]. (2) Largest CommonDescription: Choose the pair which has the most abundant related things in common. If several pairs have the same largest CommonDescription, the smallest Denominator will be chosen.

Assume m means the entity number of p, n means the entity number of q, ap means the average size of p's description, aq means the average size of q's description. Then the time complexity of Description Overlap Measuring is $O(m \times n \times (ap+aq))$. $(ap+aq)$ means the time complexity of $CommonDescription(p,q)$.

All in all, four strategies can be used to deal with the semantic relatedness between two named entities. They are described detailedly in Table 2. An experimental study is provided in Section 4 to compare the four strategies.

4 Experiments

In this section, we conducted some experiments to demonstrate the effectiveness of our proposed approach in named entities semantic relatedness measuring. The experiments results showed that our approach greatly outperformed the previous semantic relatedness measuring approaches. Extensive experiments were also carried out to prove the robustness of our approach.

4.1 Experimental Setup

4.1.1 LOD Data Sources In our work, we randomly select two cross-domain data sources: DBpedia, Freebase, and a specific-domain data source: Musicbrainz (DBtune). In our future work, we will consider other domains and do more comprehensive experiments. we have generated a LOD Index which includes DBpedia, Musicbrianz (DBtune) and Freebase. The scale statistics are shown in Table 3.

Table 3. LOD scale statistics

Data Source	DBpedia	Musicbrainz (DBtune)	Freebase
Entity Number (million)	3.9	23.2	29

From Table 3 we find that the entity number of Musicbrainz (DBtune) and Freebase exceeds DBpedia greatly. As DBpedia is extracted from Wikipedia, and Wikipedia has a larger coverage than WordNet, we can conclude that LOD does enlarge entity coverage tremendously than Wikipedia and WordNet. So by leveraging LOD, the entity coverage problem, which appears in traditional knowledge based approaches, can be solved.

4.1.2 Evaluation Measure There are two different correlation measures which have been used for evaluating semantic relatedness measuring. The Pearson product-moment correlation coefficient γ is to correlate the scores computed by a semantic relatedness measuring approach with the numeric judgements of semantic relatedness provided by humans. The Spearman rank order correlation coefficient ρ is to correlate named entities pair rankings. Zesch [23] compared these two measures and recommended to use Spearman rank correlation to evaluate semantic relatedness measuring. So in our experiments we just leverage Spearman rank correlation ρ as the evaluation measure.

4.1.3 Dataset Unfortunately, there is no benchmark data set for named entities semantic relatedness measuring. In our experiment, we make our own data set and offer it as a standard for testing named entities semantic relatedness. In our work, we have generated a LOD index which includes DBpedia, Musicbrianz

(DBtune) and Freebase. Musicbrainz (DBtune) mainly focuses on the music domain while DBpedia and Freebase are cross-domain data sources. we randomly select 60 music related entities pairs from last.fm[9] and 60 other domains entities pairs from Wikipedia, giving a total of 120 pairs of named entities.

In the evaluation work, the semantic relatedness of each pair is rated by six subjects with the following instructions:

Indicate how strongly these named entities are related using integers from 0 to 4. The description and an example corresponding to each number are given as follows, and if you think some pairs fall in between two of these categories you must push it up or down (no halves or decimals).

0: not at all related; "Linux" and "Beijing"
1: vaguely related; "China" and "Tokyo"
2: indirectly related; "Backstreet Boys" and "Britney Spears"
3: strongly related; "Backstreet Boys" and "As Long as You Love Me"
4: inseparably related; "Gate of Heavenly Peace" and "Tiananmen"

The named entities pairs were sorted in descending order by average score, and 100 pairs were selected in order to balance the rate distribution from 0 to 4. The average Spearman rank correlation ρ among these six subjects is 0.9617, which means the rate result is objective. Moreover, 0.9617 can also be used as the upper bound of the performance.

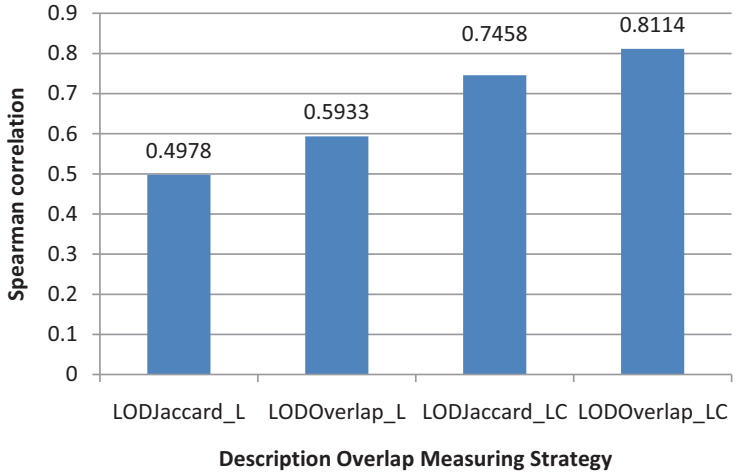

Fig. 3. Four Description Overlap Measuring strategies' performance; Spearman rank correlation ρ with humans

[9] http://www.last.fm/

4.2 Description Overlap Strategy Comparison

In this section, we compare the performance of the four strategies in Description Overlap Measuring. The results are shown in Figure 3.

From Figure 3, we can see that LODOverlap_L outperforms LODJaccard_L, LODOverlap_LC outperforms LODJaccard_LC. This tells us that when dealing with the semantic relatedness between named entities, it is more reasonable to focus on the less description named entity. From the results, we also find that LODOverlap_LC is better than LODOverlap_L, LODJaccard_LC is better than LODJaccard_L. It is mainly caused by the noise in LOD when handling multi-pairs problem. In LOD, there exist some obsolete and incomplete uris. They have little and even wrong description which will lead to high overlap between two unrelated named entities and thus reduce the performance. Leveraging the largest common description pair has two advantages: (1) The largest common description pair is probably well described in LOD, which can reduce the influence of noise in LOD; (2) It is more likely to have an objective semantic relatedness which conforms to the human sense.

In the following experiments, we choose LODOverlap_LC as the strategy of our approach LODDO. Table 4 shows some result examples of LODDO.

Table 4. Result examples of LODDO

Named Entities pair	LOD Description Overlap
"*Gate of Heavenly Peace*" and "*Tiananmen*"	1
"*Backstreet Boys*" and "*As Long as You Love Me*"	0.3758
"*Backstreet Boys*" and "*Britney Spears*"	0.1538
"*China*" and "*Tokyo*"	0.0556
"*Linux*" and "*Beijing*"	0.0047

4.3 Semantic Relatedness Measuring Performance

Six previous semantic relatedness approaches are used to compare with our proposed approach.

- Rad [17] regards WordNet as a graph: concepts as vertexes and all types of relationships as edges. Given two concepts, the semantic relatedness is represented by the shortest path length between them, the larger path length, the weaker semantic relatedness between them.
- GlossOverlap [20] calculates the text overlap of two concepts' glosses, which are the first paragraph of their Wikipedia articles, to measure the semantic relatedness. GlossOverlap is defined as follows:

$$GlossOverlap(p,q) = \tanh\left(\frac{overlap(Gloss(p), Gloss(q))}{length(Gloss(p)) + length(Gloss(q))}\right) \quad (4)$$

– Intrinsic Information Content (IIC) [16] applies an intrinsic information content measure relying on the hierarchical structure of the Wikipedia category tree. It's defined as follows:

$$IIC(p,q) = 1 - \frac{log(hypo(lcs(p,q)) + 1)}{log(C)} \tag{5}$$

where *lcs(p, q)* means the least common subsumer of p and q in Wikipedia category tree. *hypo(lcs(p, q))* is the number of hyponyms of node *lcs(p, q)* and C equals the total number of conceptual nodes in the hierarchy.

– ESA [7] firstly constructs weighted vector of Wikipedia concepts to each input text. Then to compute semantic relatedness of this pair of text, it compares their vectors using the cosine metric.

– WebJaccard and WebOverlap [5] are two popular co-occurrence measures to compute semantic similarity using page counts. They are defined as follows:

$$WebJaccard(p,q) = \frac{H(p \cap q)}{H(p) + H(q) - H(p \cap q)} \tag{6}$$

$$WebOverlap(p,q) = \frac{H(p \cap q)}{min(H(p), H(q))} \tag{7}$$

Here *H(p)* denotes the page counts for the query p in a search engine. In our experiment, we choose Google[10] to get page counts.

Figure 4 shows the results of these approaches on the test dataset.

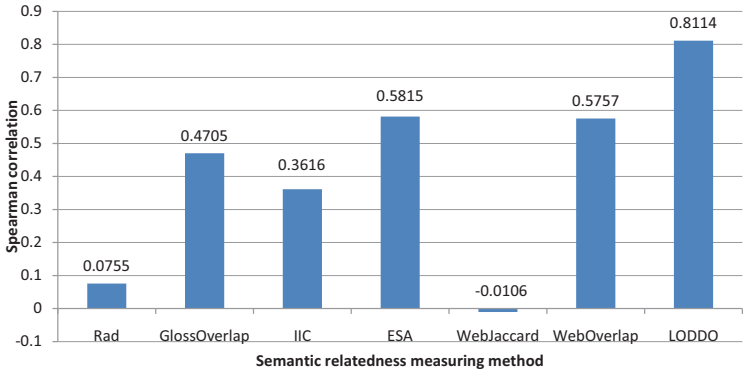

Fig. 4. Different approaches' performance; Spearman rank correlation ρ with humans

From Figure 4, we can find that our proposed approach significantly improves the performance of named entities semantic relatedness measuring. Even compared with ESA, the second best performance, we get an improvement of 39.6%.

[10] http://www.google.com

As WordNet has limited entity coverage and 75 pairs in the test dataset cannot be measured because of the miss-hit in WordNet, Rad achieves a low Spearman rank correlation. ESA, GlossOverlap and IIC obtain a better performance than Rad, because Wikipedia has a larger coverage and richer description than WordNet. In Wikipedia, only 6 pairs in the test dataset is miss-hit. However, ESA considers the words in a name independently, thus may misunderstand the meaning of the name. GlossOverlap regards the uncritical words equally to the critical words in the gloss, thus the effectiveness may be reduced by the uncritical words. Since IIC only takes into account the category hierarchy relation without considering other meaningful relations, the performance is limited. WebJaccard and WebOverlap use the Google search statistical information to measure the semantic relatedness between named entities. As they regard a name in all documents as the same meaning, the effectiveness can be reduced. Since WebJaccard considers the two named entities equally, the larger hit entity brings more noise which influences the accuracy greatly. Furthermore, WebOverlap provides a better performance than WebJaccard, which proves the heuristics that the semantic relatedness should bias the less description entity.

4.4 LOD Data Source Selection

In this section, the influence of selecting different LOD data source is figured out. What will the performance change if we merge the data sources rather than use them singly. Table 5 gives the results of using different data sources.

Table 5. Performance of selecting different data sources

Data Source	average description number	missed pairs number	Spearman rank correlation ρ with humans
Musicbrainz (DBtune)	35.79	26	0.0128
Freebase	10468.4	16	0.4217
DBpedia	11658.5	6	0.7668
Musicbrainz (DBtune) & Freebase & Dbpedia	26076.1	0	0.8114

It is noted that the Spearman rank correlation is calculated without the consideration of the pairs which can't be found in corresponding data sources. There are two reasons why Musicbrainz (DBtune) gets such a low performance: (1) The description of an entity is insufficient (only 35.79 descriptions on average), compared with other data sources (more than 10k descriptions on average); (2) The entity corresponding to a name in Musicbrainz (DBtune) sometimes is not the sense in our daily experience, for example "Ferrari" is a song in Musicbrainz (DBtune) rather than automotive in common sense. From the column "*missed pairs number*" we can know that the use of single data source also leads to entity coverage problem, however, by merging the data sources together, the coverage

problem can be relieved. Although the average description number of Freebase and DBpedia are similar, their performances are different. So we can conclude that different data sources may have different constructions and qualities, which contributes to the different semantic relatedness measuring performances. In addition, having more description is likely to lead to better performance. It verifies that with more data sources, the performance can be improved steadily, which proves the robustness of our approach.

5 Conclusion

In this paper, we target on the task of named entities semantic relatedness measuring. As the existing knowledge based approaches have the entity coverage issue and the statistical based approaches have unreliable result to low frequent entities, we propose a more comprehensive approach by leveraging LOD to solve these problems. By exploiting the semantic associations in LOD, we propose a novel algorithm, called LODDO, to measure the semantic relatedness between named entities. Specifically, we first propose a mechanism to index the various LOD sources which can be used to find the entities corresponding to a specific entity name fleetly. Then, we bring forward LOD Description Overlap to measure the named entities semantic relatedness. The experimental results show that our approach greatly outperforms the previous semantic relatedness measuring approaches. And it is robust to leverage more data sources in LOD and provide better performance.

In the future, we plan to investigate more data sources from LOD in order to extend the coverage and promote the performance. We will investigate the quality of LOD and see how it will influence the performance of semantic relatedness measuring between named entities. We will also try to find a uniform approach to measure the semantic relatedness between abstract concepts and named entities.

References

1. Linked open data. `http://linkeddata.org`
2. Wikipedia:redirect. `http://en.wikipedia.org/wiki/Wikipedia:Redirect`
3. Banerjee, S., Pedersen, T.: An adapted lesk algorithm for word sense disambiguation using wordnet. In: The Third International Conference on Computational Linguistics and Intelligent Text Processing. pp. 136–145. London: Springer Verlag (2002)
4. Bollacker, K., Evans, C., Paritosh, P., Sturge, T., Taylor, J.: Freebase: a collaboratively created graph database for structuring human knowledge. In: The 2008 ACM SIGMOD International Conference on Management of Data. pp. 1247–1250. New York, USA (2008)
5. Bollegala, D., Yutaka, M., Ishizuka, M.: Measuring semantic similarity between words using web search engines. In: The 16th International Conference on World Wide Web. pp. 757–766. New York, NY, USA (2007)

6. Cilibrasi, R., Vitanyi, P.M.B.: The google similarity distance. IEEE Trans. Knowledge and Data Engineering (3), 370–383 (2007)
7. Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. In: The 20th International Joint Conference on Artificial Intelligence (IJCAI). pp. 1606–1611. Hyderabad, India (2007)
8. Gracia, J., Mena, E.: Web-based measure of semantic relatedness. In: The 9th International Conference on Web Information Systems Engineering. pp. 136–150. Springer-Verlag Berlin, Heidelberg (2008)
9. Hirst, G., St-Onge, D.: Lexical chains as representation of context for the detection and correction malapropisms. In: WordNet: An Electronic Lexical Database (Language, Speech, and Communication), pp. 305–332. MIT Press (May 1998)
10. Jarmasz, M., Szpakowicz, S.: Roget's thesaurus and semantic similarity. In: Recent Advances in Natural Language Processing. pp. 212–219 (2003)
11. Lesk, M.: Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In: The 5th Annual International Conference on Systems Documentation. pp. 24–26. Toronto, Canada (1986)
12. Miller, G.A.: Wordnet: A lexical database for english. Communications of the ACM (11), 39–41 (1995)
13. Milne, D., Witten, I.H.: An effective, low-cost measure of semantic relatedness obtained from wikipedia links. In: The AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008). Chicago, IL (2008)
14. Patwardhan, S., Banerjee, S., Pedersen, T.: Using measures of semantic relatedness for word sense disambiguation. In: The Fourth International Conference on Intelligent Text Processing and Computational Linguistics. pp. 241–257. Berlin, Heidelberg (2003)
15. Patwardhan, S., Pedersen, T.: Using wordnet based context vectors to estimate the semantic relatedness of concepts. In: The EACL 2006 Workshop Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together. pp. 1–8. Trento, Italy (2006)
16. Ponzetto, S.P., Strube, M.: Knowledge derived from wikipedia for computing semantic relatedness. Journal of Artificial Intelligence Research pp. 181–212 (2007)
17. Rada, R., Mili, H., Bicknell, E., Blettner, M.: Development and application of a metric on semantic nets. IEEE Transactions on Systems, Man, and Cybernetics (1), 17–30 (1989)
18. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: The 14th International Joint Conference on Artificial Intelligence. pp. 448–453. San Francisco, CA, USA (1995)
19. Spanakis, G., Siolas, G., Stafylopatis, A.: A hybrid web-based measure for computing semantic relatedness between words. In: The 2009 21st IEEE International Conference on Tools with Artificial Intelligence. pp. 441–448. Washington, DC (2009)
20. Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. In: The 21st National Conference on Artificial Intelligence. pp. 1419–1424. Boston, MA (2006)
21. Swartz, A.: Musicbrainz: A semantic web service. IEEE Intelligent Systems (1), 76–77 (2002)
22. Wubben, S., van den Bosch, A.: semantic relatedness metric based on free link structure. In: The Eighth International Conference on Computational Semantics. pp. 355–358. Tilburg, The Netherlands (2009)
23. Zesch, T., Gurevych, I.: Wisdom of crowds versus wisdom of linguists measuring the semantic relatedness of words. Natural Language Engineering (1), 25–59 (2010)