Research Article

Farzam Matinfar*

Linking Web Resources in Web of Data to Encyclopedic Knowledge Base

<https://doi.org/10.1515/comp-2020-0102>

Received Oct 11, 2019; accepted Mar 24, 2020

Abstract: This paper introduces Wikipedia as an extensive knowledge base which provides additional information about a great number of web resources in the semantic web, and shows how RDF web resources in the web of data can be linked to this encyclopedia. Given an input web resource, the designed system identifies the topic of the web resource and links it to the corresponding Wikipedia article. To perform this task, we use the core labeling properties in web of data to specify the candidate Wikipedia articles for a web resource. Finally, a knowledge based approach is used to identify the most appropriate article in Wikipedia database. Evaluation of the system shows the high performance of the designed system.

Keywords: Semantic Web, Linked Data, Disambiguation, Interlinking, Information Retrieval

1 Introduction

In recent years, the web has evolved from a global information space of linked documents to a space where both documents and data are linked [1]. Linked data refers to the published data on the web, where the data is machine-readable, has explicit meaning, is linked to other external datasets, and can be linked to from external datasets [1]. Recently, various datasets have been published in the web of data following linked data principles [2] including DB-Pedia, DBLP, Freebase, GeoSpecies, etc. Moreover, various domains such as biology [3], and art [4] have applied these principles to publish their linked data.

The web of data introduces an environment where various researches and applications including information retrieval systems benefit from its specifications [5, 6] and various tools have been introduced for linked Data consumption [7]. One of the main specifications of the linked data is *RDF¹ links* which can determine the relationships between web resources. The RDF links in linked Data generate triples where the subject and object resources are URI references in the namespace of initial dataset and target dataset respectively [8] and the predicate part shows the relationship between them. These predicates are semantic terms which are selected from ontologies. Ontologies introduce different terms to represent various meanings; for example, owl:sameAs, and rdf:type show the equality relationships and type relationship between two entities respectively.

One of the widely used predicates in semantic web is *rdfs:seeAlso*. It has been defined in RDF schema and links two related web resources. The World Wide Web Consortium, a community which develops standards in web, states that *rdfs:seeAlso* is an instance of *rdf:property* that is used to indicate a resources that might provide additional information about the subject resource². The link *seealso* is extensively used in traditional web to link related web documents. Generally, users decide about the target documents which provide additional information. In [9] different possible meanings of this predicate in the semantic web are discovered by analyzing the web data. This predicate can be used to link the same entities or to link web resources to the documents which provide additional information about the subject resource. However, linking web resources manually is time consuming and hence interlinking datasets and linking the web resources to pages which provide additional information has been turned to an important topic in the web of data.

Wikipedia is one of the largest online repositories of encyclopedic knowledge that can be seen as a target database. It is continuously growing and the number of articles in English Wikipedia has increased from less than 1

Article note: This work was supported by the Allameh Tabataba'i University, Department of Statistics, Mathematics, and Computer Science.

***Corresponding Author: Farzam Matinfar:** Department of Statistics, Mathematics, and Computer Science, Allameh Tabataba'i University, Tehran, Iran; Email: f.matinfar@atu.ac.ir

¹ Resource Description Framework

² <https://www.w3.org/2001/sw/RDFCore/Schema/200203/>

million in 2006 to more than 5.7 million articles in 2018³. Moreover, Wikipedia is available for more than 302 languages and the number of articles ranges from a few pages per language to millions of articles per language⁴. Wikipedia is a free knowledge repository which is supported by the Wikipedia foundation and the fact it is collaboratively edited is one of its important characteristics. It has been used in various researches such as named entity recognition and disambiguation [10–15], document topic identifications and relatedness [16–18], information retrieval [19], WorldNet to Wikipedia mapping [20, 21], etc. Wikipedia uses hyperlinks between pages to connect related articles. These links have been used in various knowledge bases such as YAGO and DBpedia and play an important role in our designed system. In [22] a rule mining technique is presented to propose semantic relations for these hyperlinks.

In this paper, we benefit from Wikipedia dataset which contains information about millions of entities. These entities belong to different domains and there is a lot of information about each of them in each wiki page. The goal of our designed system, which we call RDFWikify, is linking web resources to the corresponding Wikipedia pages which provide additional information. Our system mainly consists of four parts: 1) extracting candidate pages, 2) constructing contexts of web resources, 3) constructing contexts of candidate pages and 4) mapping contexts step which is able to link RDF pages in the web of data to the corresponding pages in the Wikipedia knowledge base.

We apply a knowledge-based method to find the most appropriate sense of a RDF page in the Wikipedia database. There are several knowledge-based approaches to solve the word sense disambiguation problem [23, 24]. Knowledge-based approaches are based on the information in the sense definitions provided in the knowledge base. Lest algorithm (a knowledge based approach) was introduced in [24] where it tries to identify the most appropriate sense of a word in a context based on the overlap of contexts of the ambiguous word and the candidate senses in the target knowledge base.

The rest of the paper is organized as follows: in section II, related works are presented. In section III, Wikipedia knowledge base and its specifications are introduced. In section IV, the architecture of the designed systems is illustrated. In section V, the evaluation of the designed system is provided and in section VI, conclusion and future work are presented.

2 Related Work

There have been various researches about enriching texts and web pages [25, 26]. In [25], links are added to web pages to improve the users' navigation and in [26], book sections which are not well written, are identified. Generally, annotating web pages and linking web resources leads to more information provision. Linking web resources which describe the same entity is a research area in semantic web environment and various tools have been designed to consume linked data [27]. In recent years, several approaches and tools [28–30] have been designed to do the interlinking process in semantic web. In [29] utilizes the mathematical characteristic of metric spaces to filter out a large number of instance pairs from mapping process. Approximate distances between pairs are calculated based on triangle inequality. In [30] an approach is suggested which reduces the role of user in interlinking process. In this research, property filtering is used to remove problematic attributes and instance profiling is applied to represent each resource by a relevant sub-graph to enhance the mapping process. Using string matching and similarity measures is one of the widely used approached in this area [31–33]. In [31], user identifies the correspondence between attributes and then the values of each attribute pairs are compared using a string similarity measure. In similarity measurement, various metrics such as jaro similarity, jaroWinklerSimilarity, etc. can be used. In [32] a similar string similarity measure is used in interlinking process. However, there several steps are performed before mapping including preprocessing, checking, materialization, translation and ontology evolution. In these steps, operations such as checking the consistency of ontology, translation of given properties from one language to another, and modifying properties values are performed. In [33] the similarity between strings in the object parts of triples and adaptive learning are used to make the fusion ontology. In several researches [32, 33] fusion ontology can also be created. In fusion process datasets are fused according to the alignment and a new ontology is created.

There are several features which we can use them to categorize the existing approaches. These features include domain of data, ontology alignment ability, the extent of user contribution, etc. Some of researches are mainly focused on designing approaches for interlinking datasets in specific domains such as linking movie datasets [34], linking music datasets [35], linking spatial data, interlinking video programs [36], interlinking multimedia [37] and linked data in education [38]. In [39], a domain inde-

³ https://en.wikipedia.org/wiki/Main_Page

⁴ https://meta.wikimedia.org/wiki/List_of_Wikipedias

pendent approach for interlinking instances is proposed which uses matching histories in its process.

Most of the designed approaches use the alignment between attributes (ontology terms) in interlinking process. Therefore, some of these approaches do not support multi ontology environments [33]. In other words, datasets have to be described with the same ontologies. Another category of the interlinking methods require inputs from the users. For example, the alignment of ontologies is needed as input to do the interlinking process [31] or user contribution and comment is another way to interlink datasets [40] which is not efficient for large datasets.

In several domains, there are primary keys, similar to ones that exist in database systems, which can be used to recognize the described entities. For example, ISSN numbers are used in the publishing domain. However, the problem is the lack of such keys in most of the domains.

One of the widely used approaches in interlining datasets is analyzing and comparing the content of the web resources where ontology matching can play an important role. The greater the more similarity between web resource descriptions means the greater the more probability of their equivalence. It is possible to use an ontology learning model [41] and ontology mapping and ontology fusion approaches [42, 43] to perform the interlinking process. However, the main obstacle with these approaches is schema heterogeneity. Dataset owners usually use various and different ontologies and properties to describe their dataset, makes it more difficult to compare and to identify the web resources describing the same entity.

We use core labeling properties [44] and context of the web resources to perform the mapping process in our designed system. In the Web of Data, web resources describe entities and usually in these descriptions there are properties which are used to present the name of entities. The set of properties *i.e. core labeling properties* includes `rdf-schema#label`, `foaf:name`, `po.owl#atomname`, etc. For example consider the following information in *dbpedia* dataset:

URI: <http://dbpedia.org/resource/Plant>

Part of description: <http://dbpedia.org/resource/Plant>
foaf:name Plants

In this example, property `foaf:name` is a core labeling property.

To analyze the Wikipedia knowledge base, various semantic-related text mining approaches [45–47] can be applied. However, we benefit from several characteristics of Wikipedia (described in the next section) to retrieve the

relevant articles and texts and to identify the topics and their meaning in articles.

While the goal of RDFWikify is to find the most appropriate article (each article may act as a sense for a title term) from Wikipedia, which is big sense repository, several researches in Word Sense disambiguation topics [48–50] use Wikipedia as an external repository to increase the performance of finding the best sense of a word, and computing word relatedness [51].

Entity linking and linking terms in a text to Wikipedia knowledge base to find additional information about them was previously investigated in [12, 16, 52, 53]. In [52] extracting keywords in a text and linking them to the Wikipedia repository are the main objectives. The overlaps between the contexts of Wikipedia pages and ambiguous words are computed to find the corresponding pages and we apply a similar approach in our system to find the pages which provide additional information about the input web resources.

3 Wikipedia knowledge base

Wikipedia is a free multilingual online encyclopedia. Its articles are written collaboratively by volunteers. It has about 100 thousand active contributors and it has become one of the largest and most popular reference works in the internet. It consists of millions of pages each of which describes an entity or a concept or an event. In Wikipedia structure, some of the pages are called “redirect”. A redirect page is a page that does not have any content itself and leads the readers or applications to another page or in some cases to a section of a page. For example, if you search “Persia” or type <http://en.wikipedia.org/wiki/Persia> in your browser, then you will be directed to a page that describes “Iran” and the expression “redirected from Persia” is stated in the top of the page. There are several reasons that Wikipedia employ redirects⁵. Redirecting mechanism makes users able to define alternative names for a page.

Another structure that is highly relevant to our work is the disambiguation page. Sometimes, one title can refer to several entities. In such situations, the users are taken to a disambiguation page which shows the different possible meanings of a title. For example, when you search “Persian” or type “<http://en.wikipedia.org/wiki/Persian>”

⁵ For more information visit <http://en.wikipedia.org/wiki/Wikipedia:Redirect>

in your browser, a disambiguation page is shown which contains the different possible meanings for term “Persian” and each meaning has a link to the corresponding Wikipedia article.

Sometimes, when a user searches a title, the returned page is not a disambiguation page. However, the user is able to go to the disambiguation page related to the searched title by the link provided in the top of the page. This link contains the name of searched title and the phrase “(disambiguation)”. For example, when searching title “Persia”, a non-disambiguation page is returned that contains a link named “Persia (disambiguation)” in the top of the page. Clicking that link navigates the user to the page which has the different meanings of “Persia”.

The first part of each article in Wikipedia, which can contain several paragraphs, provides the most important information related to the article’s title. This part usually has enough information for readers to recognize the entity that is described in the article without knowing the title of the page. After the first part of an article, usually there are several sections which provide additional information about the entity in different point of views.

4 RDFWikify System

The main technical goal of this work is mapping RDF pages to the corresponding Wikipedia articles. The architecture of the system is shown in Figure 1. The steps to link a RDF page to a Wikipedia article are as follows:

- Recognizing the name of entity which is described in the page
- Identifying and extracting candidate articles from Wikipedia
- Building the context of the input page
- Building the context of candidate articles
- Mapping contexts and link generation

4.1 Identifying titles of RDF pages

Labeling properties are ontology terms which are used to present the title or the name of the entity. Given a RDF page p , we benefit from core labeling properties used in the semantic web to identify the title of the input page.

In the web of data environment, there are various ontologies. So, datasets owners can use some of them or they can even create their own ontologies to describe their entities. So, in the heterogeneous web environment, there are many labeling properties such as foaf:name, and dc:title.

Figure 1: Architecture of the RDFWikify system

Table 1: Labeling terms

Ontology Terms

http://www.aktors.org/ontology/portal#name
http://xmlns.com/foaf/0.1/givenName
http://www.w3.org/2000/10/swap/pim/contact#fullName
http://xmlns.com/foaf/0.1/surName
http://swrc.ontoware.org/ontology#title
http://swrc.ontoware.org/ontology#booktitle
http://www.aktors.org/ontology/portal#has-pretty-name
http://purl.uniprot.org/core/orfName
http://purl.uniprot.org/core/name
http://www.daml.org/2003/02/fips55/fips-55-ont#name

In [44], a data mining approach is presented to discover the most important labeling properties in the web of data. Some of these terms are shown in Table 1.

For each web resource, a query is built to retrieve the object part of the triple(s) which its predicate is equal to the one the labeling properties. For example, consider the web resource address <http://dbpedia.org/resource/Berlin>. There are several triples in its description including:

<http://dbpedia.org/resource/Berlin> *dbp:name* *berlin*
(subject) (predicate) (object)

Predicate *dbp:name* is one of labeling properties. Therefore, the object part i.e. “berlin” represents the entity name.

4.2 Extraction of Candidate Wikipedia Articles

Millions of articles exist in Wikipedia repository each describing an entity or concept or event. Given a RDF page p , considering all articles in Wikipedia as candidate target pages is not an efficient solution. In this step, we identified

candidate Wikipedia articles for the given input RDF page. We queried Wikipedia repository using entity name (identified in previous step) to retrieve the candidate articles. We used the dump files of Wikipedia⁶ to identify the candidate articles and disambiguation pages. We stored Wikipedia articles and their data in MySQL database and then used the freely available JWPL Wikipedia API [54] to access and query them. The outputs of a query are articles (including the addresses of the pages and the whole content) which their titles correspond to the input title which was identified using labelling properties.

Since the title of the input page may have various meanings, we extract several articles (as candidates) from Wikipedia, each describing one of the meanings. To do that, we extract different senses of the input title from the corresponding disambiguation page. Each of the senses in disambiguation page represents an article in Wikipedia encyclopaedia and these articles are the candidate target pages. For example, consider that “Persia” is the title of the input web resource. We query Wikipedia with this title to retrieve the corresponding target pages. Due to the fact that the retrieved page is not a disambiguation page, we retrieve the content of the *Persia (disambiguation)* page which holds the senses of the Persia. Finally, candidate articles are extracted regarding the identified senses. In some cases there is only one sense for the title. It means that there is not a disambiguation page for that title and the retrieved page represents the only candidate page.

4.3 Making candidate articles’ contexts

We use the information provided in the content of articles to build the contexts. Each Wikipedia article usually consists of several paragraphs and each paragraph includes sentences which are related to the topic of the page. Hence, the words provided in each page can be used as the context of the page. But, each page usually contains thousands of words and most of the words may not be useful to make a rich context.

One of the important characteristics of Wikipedia that we used is the provided links in each article. The style manual of Wikipedia states that the links in each article should be relevant and helpful in the context⁷. Hence, we use these links to construct the context of the pages. The most related part which describes the topic of an article

is the first paragraph. We used the links in these parts to build the contexts of articles.

4.4 Making web resources’ contexts

In this step, we constructed the disambiguation contexts of the input web resources. The existing data in a given web resource, are related to the main topic of the page. These data are presented by RDF data model. The following algorithm describes the process of disambiguation context construction.

Algorithm:

Given web resource w , we extracted related data to construct the disambiguation context $C(w)$. These data are connected to w directly or indirectly. Direct data are those which are available in the description of w and the indirect data are acquired by analyzing web resources which are linked to w . We assign different weights to the extracted data based on their distance with w . The pseudo code of the process is shown in algorithm 1. Several parameters such as *local distance*, *distance*, *weight*, *MaxSize*, and *Max-URI* are used in this algorithm. The descriptions of the parameters are as follows:

Local Distance: analyzing web resource w , several triples are extracted as follows:

$$p_i \quad \text{predicate} \quad q_i$$

We define the local distance between w and q_i as follows:

$$\text{dis}(w, q_i) = \begin{cases} 1, & w = p_i \\ 2, & w \neq p_i \end{cases} \quad (1)$$

Where $q_i \neq w$

In the description of w , if the subject resource is equal to w , then the object resource is connected directly to w . In situations where neither p_i nor the q_i are equal to w , the local distance is 2.

It is possible that the distance between w and a web resource q_i be identified through different triples. In such situations the minimum local distance is considered as the final local distance.

Distance: to build the context of w i.e. $C(w)$, several RDF web resources (which are extracted from w ’s description) are analyzed. As it is shown in algorithm 1 (line 6), this process is repeated until the end criteria. In this process, there

⁶ The dump files are available in “<http://dumps.wikimedia.org/>”

⁷ For more information visit http://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style#Wikilinks

is a collection of web resources. Some of these resources are connected to w indirectly. We define the distance between w and the other web resources y_i as follows:

$$\begin{aligned} dis(w, y_i) &= dis(w, m_1) + dis(m_1, m_2) \\ &+ dis(m_2, m_3) + \dots + dis(m_n, y_i) \\ m_i &\in \text{intermediated web resources} \end{aligned} \quad (2)$$

Algorithm 1

Input: uri, w , α , β , θ and λ

Output: context

Output: bag*

```

1: initialize a queue, q=empty, initialize context=empty;
2: initialize bag=empty*;
3: nwords=0;
4: nuris=0;
5: add (w, 0) to q; //q contains pairs (web resources, distance to w)
   // that will be investigated unless termination condition
6: While (q is not empty) && !(termination condition)
7:   object=retrieve the first object in q and remove it from q
8:   uri=object.firstfield // uri address
9:   distance=object.secondfield //distance of the uri to w
10:  nuris++
11:  initialize tp=triples in uri description
12:  for every triple in tp do
      //building the context of w (find
      // values and strings in uri, add them to
      // context, retrieve new uris, add them to q
13:    if the object-part of triple is a value
14:      add value to the bag*
15:      if (subject-part==uri) //triple directly describes the
                               web resource
16:        weight =  $\alpha / (\text{distance} + 1)$ 
17:      Else
18:        weight =  $\beta / (\text{distance} + 1)$ 
19:      end if
20:      if (object-part does not exist in context)
21:        add (object-part, weight) to context
22:        nwords++;
23:      end if
24:    end if
25:    if the object-part is a uri
26:      if (subject-part=w)
27:        add(object-part, distance+1) to q
28:      Else
29:        add(object-part,distance+2) to q
30:      end if
31:    end if
32:  end do
33: end While
34: return context
end criteria= ( $\theta < \text{nwords}$ ) or ( $\lambda < \text{nuris}$ )

```

* The codes with star will be used in algorithm 2.

The distance is calculated by adding the distances between web resources located virtually between w and y_i .

Weight: while building the disambiguation context of a web resource w , various values are added to the context. However, these values have different importance regarding their relationships to the input web resources w . Consider that data l is extracted from web resource w_l . We define parameter *weight* which shows the relative importance of the data l . This parameter is calculated based on the distance between w and w_l i.e. $dis(w, w_l)$ as follows:

$$weight(l) = \begin{cases} \frac{\alpha}{dis(w, w_l)+1} & \text{if subject resource of } l = w_l \\ \frac{\beta}{dis(w, w_l)+1} & \text{if subject resources of } l \neq w_l \end{cases}$$

$$0 \leq \beta, \alpha \leq 1 \quad (3)$$

Where α and β are constant coefficients and generally, α must have a greater value than β . As it is shown in formula (3), the weight of a data has a reverse relation with the distance of the web resource where the data is located in.

Max_Size (θ) and MAX_URIs (λ): in context making algorithm i.e. algorithm 1, there are two termination criteria. The first one (θ) shows the maximum allowed size of the context and the second one (λ) shows the maximum number of web resources which can be analyzed. Both of these parameters are related to the performance issue. While analyzing web resources related to the input web resource w , we should take care about the distances. The data extracted from web resources with longer distances have less connection and relationship with w and may provide unrelated information. Moreover, analyzing a high number of web resources is time consuming. Therefore, we limit the number of web resources (λ) for higher performance. Similarly, bigger size of the bag of words (θ) allows unrelated information to be entered to the contexts, making the mapping process (which will be discussed in a later section) inefficient. Therefore, these parameters are defined to control the performance issues.

4.5 Mapping Algorithm and Link Generation

In previous steps, the contexts of the input w and candidate Wikipedia articles are made. The goal of this step is identifying the most appropriate article explaining the same entity described in w . However, it does not mean that a corresponding article necessarily exists in the Wikipedia

knowledge base. Therefore, we have:

$$\text{map}(w) = \begin{cases} \text{article}_w & \text{entityname}(w) = \text{title}(\text{article}_w) \\ \emptyset & \text{no correspondence exists} \end{cases} \quad (4)$$

Where $\text{entityname}(w)$ is the title of the entity described in w .

In section B, an approach for identifying the candidate articles is presented. We store them in a collection as follows:

$$\text{candidate} - \text{articles} = \{a_1, a_2, a_3, \dots, a_n\} \quad (5)$$

We use conditional probability $p(a_i|w)$ to identify the target article. Among the articles in *article-collection* set, the one which maximizes the conditional probability is regarded as the target article. This probability is calculated as follows:

$$\begin{aligned} \text{map}(w) &= \text{argmax}_{a \in \text{candidate articles}(w)} p(a|w) = \\ &\text{argmax}_{a \in \text{candidate articles}(w)} \frac{p(a, w)}{p(w)} \end{aligned} \quad (6)$$

The denominator part is equal and constant for all probabilities. Therefore, we have:

$$\text{map}(w) = \text{argmax}_{a \in \text{candidate articles}(w)} p(a, w) \quad (7)$$

Where:

$$p(a, w) = \frac{u(a, w)}{\sum_{a_i \in \text{candidate_articles}(w)} u(a_i, w)} \quad (8)$$

We define a utility function to estimate the probability $p(a, w)$. This utility represents the size of intersection between two contexts as follows:

$$u(a, w) = |C(a) \cap C(w)| \quad (9)$$

Higher contexts overlap between w and a_k increases the chance of a_k to be the target article.

4.6 Web Resource context enrichment

Web resources are described using arbitrary ontology terms and their corresponding values. Object part of the triples can be a string, digit, text, etc. Our approach to make the disambiguation context is adding these data with their weight to a bag of words and finally the contents of the bags are compared with the contexts of candidate articles. However, there are situations which make these values (data) useless. They may contain text (especially long text) which usually consists of several sentences. This condition happens when some specific properties such as

Algorithm 2

Input: bag, μ

Output: Enriched_context

```

1: Enriched_context = context
2: bag is tokenized resulting in bag_token
3: For each literal in bag_token
4:   If literal is a Stopword remove it
5: size = size of (bag_token)
6: For each literal in bag_token
7:   calculate frequency (literal) and frequency_percentage(literal)
8: AM=average (frequency_percentage) //arithmetic mean
9: For each literal in bag_token
10:  If frequency_percentage(literal) >  $\mu * AM$ 
11:    Add literal to Enriched_context
12: Return Enriched_context

```

rdfs:comment, *dbpedia-owl:abstract* are used in predicate part. These properties naturally are more probable to have text values in their object part. Values which contain several words or sentences usually cannot have an effect on the contexts comparison result. There is a very low probability that the same value (for example, a text) exists in the candidates' contexts. For example, consider the URI address "http://dbpedia.org/resource/Plant" which contains the following triple:

Subject: http://dbpedia.org/resource/Plant

Predicate: dbo:abstract

Object: Green plants have cell walls with cellulose and The scientific study of plants is known as botany, a branch of biology.

While the object part of this triple has valuable information, it is long enough that there is a very high probability it does not exist exactly in the same form in other texts (here Wikipedia articles). Therefore, to make such texts usable, we need to extract the main keywords from them. The algorithm of key words extraction is depicted in algorithm 2. The steps are as follows:

- In line 14 of Algorithm 1, each value is added to the variable *bag*. Finally, *bag* has all the literals and texts in the string format.
- In line 1, the enriched context is initialized. Its initial value is the context which is generated in Algorithm 1.
- In lines 2-4, *bag* is tokenized and all tokens are put in a collection of literals and then the *Stopwords* are deleted from the collection
- Through lines 5-7, the percentage use of each literal and then the average of percentages are calculated.

These percentages show the relative frequency of each term in the description of a web resource.

- In lines 8-11, the arithmetic mean of the literals' percentage is calculated (AM) and then the literals with percentages higher than $\mu * AM$ are added to disambiguation context of a web resource. Variable μ is a parameter which is used to decide what literals to be inserted to the context regarding to their significance. This parameter takes values above 1 and choosing higher values means the fewer literals and of course the more important ones are inserted into the context.

The returned contexts by algorithm 2 are used as the contexts of the web resources in mapping process.

5 Experiments

We have conducted several experiments to evaluate the performance of the RDFWikify system. Experiments are implemented in Java environment (NetBeans IDE) and various APIs are used in these experiments⁸.

5.1 An Example

First of all, we describe an example. Consider the following URI address:

https://semanticweb.cs.vu.nl/europeana/browse/list_resource?r=http%3A//purl.org/vocabularies/princeton/wn30/synset-earth-noun-5

We query this URI using core labeling properties and the following triple is retrieved:

URI: https://semanticweb.cs.vu.nl/europeana/browse/list_resource?r=http%3A//purl.org/vocabularies/princeton/wn30/synset-earth-noun-5

Predicate: `rdfs:label`

Value: "earth"@en-us

Therefore, this URI is about "earth". We query the Wikipedia knowledge base and extract various articles related to earth using disambiguation page⁹. Parts of contexts of three candidate articles (out of more than ten ar-

ticles) are as follows:

Context of the input URI

$= \{elements, universe, thought, \dots\}$

Context of the candidate article – 1 = $\{Planet, sun, \dots\}$

Context of the candidate article – 2

$= \{electrical, engineering, fuses, \dots\}$

Context of the candidate article – 3

$= \{elements, plato, \dots\}$

Finally, the input URI is linked to the third sense of earth. In other words, the input URI is not about "planet earth" or other meanings. It is about one of the four elements composing the universe.

5.2 Evaluation of the Mapping Algorithm

In the first experiment, we evaluated the performance of the mapping algorithm regarding the different values of the used parameters using a gold standard dataset. We selected 550 RDF pages from BTC-2014¹⁰ dataset randomly and three expert users identified the corresponding articles in Wikipedia. In cases of conflict between them, we voted to identify the most appropriate article. The average number of candidate articles was 5.54 per input page.

We conducted several executions of our algorithm with different parameter values. We assigned input parameters $\mu=2$, $\alpha=1$ and $\beta=0.3$ and executed our mapping algorithm with different values for the other input parameters θ and λ .

Figure 2: Precision of the mapping algorithm with different values for parameters θ , λ

⁸ Jena, JWPL, Lucene

⁹ [https://en.wikipedia.org/wiki/Earth_\(disambiguation\)](https://en.wikipedia.org/wiki/Earth_(disambiguation)).

¹⁰ This dataset ideally includes web resources exists in the web of data. The dataset was crawled during February to June 2014. Several seed sets collected from multiple sources are used. More information in: <http://km.aifb.kit.edu/projects/btc-2014/>

Figure 3: Precision of the mapping algorithm with different values for parameters θ, λ

The results are shown in Figure 2 and Figure 3. The best results were acquired with the values $(\theta, \lambda)=(150,30)$ and $(200,40)$. However, using the pair $(150,30)$ is more efficient than the other one due to its lower running time.

5.3 Evaluation of the Mapping Algorithm: comparing with other approaches

To the best of our knowledge, the RDFWikify system is the first system that is able to map RDF web resources to the corresponding Wikipedia articles automatically. However, there are several methods which can be applied to find the target Wikipedia articles:

a) Random selection

In this approach, after identifying the candidate senses, one of them is selected randomly.

b) First sense selection

Various knowledge bases such as Wikipedia and WordNet act as sense repository and provide the different senses of a term. WordNet ranks the senses of a term based on their usage and popularity. The first sense is more frequent¹¹ and therefore more probable to present the real meaning of the input term. Based on this idea, first sense selection approach selects the first sense as the target sense.

c) Naming schemas

Using naming schemas is one of the approaches to identify the name of the entities described in web resources. This approach can be applied in domains which have accepted naming schemas. For example, BookMaushup database uses ISBN in the URL address. These unique identifiers are used for mapping web resources.

d) Title matching

One of the approaches for linking web resources uses the URI or URL address of the web resources to find the corresponding web resources. In this approach, the last parts of the URL addresses of Wikipedia articles are used to identify the title of the web resource. For example, the URL address “http://en.wikipedia.org/wiki/Plant” describes the term “plant”. One of the drawbacks of this approach is that it is unable to discover and identify various meanings of term “plant”. This approach links web resource “http://wordnet.rkbexplorer.com/id/synset-plant-noun-1” (which describes term “plant”) to the article “http://en.wikipedia.org/wiki/Plant”. But the URL of the corresponding article in Wikipedia is “http://en.wikipedia.org/wiki/Physical_plant” which is one of the senses of the term “plant”. Therefore, using the last part of the URL address may not provide the correct results.

We have compared the performance of our proposed system with the other approaches. Table 2 shows the precision and recall of the RDFWikify system and the other approaches. Precision shows the number of correctly mapped pages divided by the total number of mapped pages by the system, and recall shows the number of correctly mapped pages divided by the total number of user created mappings.

Table 2: Performance of the mapping algorithm with the values $\mu=2$, $\alpha=1$ and $\beta=0.3$ $\theta=150$, and $\lambda=30$. F-measure= $2 \cdot P \cdot R / (P+R)$

	Precision	Recall	F-measure
Random selection	46.31%	41.75%	43.91
First sense selection	54.86%	49.46%	52.02
Naming schema	100%	0.36%	0.7174
Title matching	54.86%	49.46%	52.02
RDFWikify system	92.39%	83.29%	87.61

The results show that using *naming schemas* approach results in the best precision, but it has a poor recall, because there is no naming schema for most of the web re-

¹¹ <https://wordnet.princeton.edu/documentation/wndb5wn>

sources. The results of *first sense selection* and *title matching* approaches show that they have the same performances and are able to find nearly to 50% of target articles. RDFWikify system is able to find more than 83% of the corresponding Wikipedia articles with precision more than 92% which dominates the performance of the other approaches.

5.4 Evaluation of Mapping Algorithm: other benchmark datasets

One of the main characteristics of interlinking tools is the level of automaticity. Several tools [29, 31, 35] need the user configuration effort. For example, in Silk, user specifies the properties which should be considered in interlinking process. In RDFWikify, there are just several parameters which can be tuned (such as similarity threshold).

In this experiment, we compare the performance of RDFWikify with several other interlinking approaches on two benchmarks. These benchmarks are selected from OAEI-2017 instance matching task. The results of the evaluation are presented in Table 3.

Table 3: Results on different benchmarks, compared to other linking approaches

Benchmark		Precision	Recall	F-measure
HT	Legato	0.930	0.930	0.930
	AML	0.851	0.479	0.613
	I-Match	0.680	0.071	0.129
	LogMap	0.406	0.882	0.556
	Silk	0.34	0.12	0.18
	RDFWikify	0.66	0.53	0.58
FPT	Legato	1.000	0.980	0.990
	AML	0.914	0.427	0.582
	I-Match	1.000	0.053	0.101
	LogMap	0.119	0.880	0.210
	Silk	0.45	0.2	0.27
	RDFWikify	0.51	0.76	0.61

The results show that in the HT dataset the precision and recall of the designed system is 66% and 58% respectively. The Legato system has the best performance and RDFWikify system has better performance than three other systems (I-Match, LogMap, Silk). In FPT dataset, RDFWikify system has shown better performance compared with the HT dataset. While Legato has the best precision and recall, the RDFWikify system has the second best F-measure. In the FPT dataset, most of the approaches have better pre-

cision than recall. Three of the approaches were able to find less than 27% of the target resources. However, the RDFWikify system was able to identify more than 60% of target resources.

Of the main contributions of RDFWikify is that it is able to filter out many unrelated articles and only a few articles are selected as candidates using disambiguation pages. Other target datasets do not have such features (disambiguation pages). Moreover, RDFWikify focuses on identifying senses of entities while the goal of OAEI datasets is instance matching. Eliminating the step which filters out unrelated articles can affect the performance of the designed system. However, we can observe that RDFWikify has comparable performance to the other systems. As mentioned before, we have used core labeling properties to extract candidate entities and links are generated based on context matching.

6 Conclusion and Future Work

In this paper, we introduced Wikipedia as an extensive database of sense inventory that provides additional information about web resources belonging to various and different domains. The RDFWikify system is able to identify the title of web resources, make contexts of web resources, analyze the information in the Wikipedia repository, and discover the corresponding article. The results of the conducted experiments show that the links defined in each article can be highly helpful to construct the contexts of the articles and therefore these links represent the key words in the texts of articles. The designed system benefits from these keywords, labeling properties, and available data in linked data to enrich web resources. This system has showed noticeable performance regarding to the heterogeneous nature of the web of data.

In future, we will extend and update the labeling properties set and will extend the RDFWikify system to become able to interlink various datasets in the web of data using generated links between each dataset and Wikipedia knowledge base. Moreover, in another trend, we will aggregate this system with word sense disambiguation systems to make a connection between texts (social texts, educational texts, ...) and web data to improve the performance of the information retrieval systems.

Acknowledgement: This work was supported by the Alameh Tabataba'i University. Moreover, I appreciate the anonymous persons who helped to conduct the experiments.

References

- [1] Bizer, C., Heath, T., and Berners-Lee, T.: 'The Story So Far', *International Journal on semantic Web and Information Systems*, 2009, 5, (3), pp. 1-22
- [2] <http://www.w3.org/DesignIssues/LinkedData.html>
- [3] Venkatesan, A., Ngompe, G.T., Hassouni, N.E., Chentli, I., Guignon, V., Jonquet, C., Ruiz, M., and Larmande, P.: 'Agronomic Linked Data (AgroLD): a Knowledge-based System to Enable Integrative Biology in Agronomy', *PLOS ONE*, 2018, 13, (11)
- [4] Knoblock, C.A., Szekely, P., Fink, E., Degler, D., Newbury, D., Sanderson, R., Blanch, K., Snyder, S., Chheda, N., Jain, N., Krishna, R.R., Sreekanth, N.B., and Yao, Y.: 'Lessons Learned in Building Linked Data for the American Art Collaborative', in Editor (Ed.)^(Eds.): 'Book Lessons Learned in Building Linked Data for the American Art Collaborative' (2017, edn.), pp. 263-279
- [5] Bechhofer, S., Buchan, I., Roure, D.D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., Gamble, M., Michaelides, D., Owen, S., Newman, D., Sufi, S., and Goble, C.: 'Why linked data is not enough for scientists, Future Generation Computer Systems', *Future Generation Computer Systems*, 2013, 29, (2), pp. 599-611
- [6] Tummarello, G., Delbru, R., and Oren, E.: 'Sindice.com: weaving the open linked data', in Editor (Ed.)^(Eds.): 'Book Sindice.com: weaving the open linked data' (2007, edn.), pp. 552-565
- [7] Klimek, J., Skoda, P., and Necasky, M.: 'Survey of Tools for Linked Data Consumption', *Semantic Web journal*, 2019, 10, (4), pp. 665-720
- [8] <http://wifo5-03.informatik.uni-mannheim.de/bizer/pub/LinkedDataTutorial/>
- [9] Matinfar, F., Nematbakhsh, M.A., and Lausen, G.: 'Discovery of RDFS: SeeAlso Patterns in Semantic Web', *International Journal of Pattern Recognition and Artificial Intelligence*, 2014, 28, (2)
- [10] Bunesco, R., and Pasca, M.: 'Using Encyclopedic Knowledge for Named entity Disambiguation', in Editor (Ed.)^(Eds.): 'Book Using Encyclopedic Knowledge for Named entity Disambiguation' (2006, edn.), pp. 9-16
- [11] Nguyen, H., and Cao, T.H.: 'Named entity disambiguation on an ontology enriched by Wikipedia', in Editor (Ed.)^(Eds.): 'Book Named entity disambiguation on an ontology enriched by Wikipedia' (2008, edn.), pp. 247 - 254
- [12] Klang, M., and Nugues, P.: 'Linking, Searching, and Visualizing Entities inWikipedia', in Editor (Ed.)^(Eds.): 'Book Linking, Searching, and Visualizing Entities inWikipedia' (2018, edn.), pp. 3426 - 3432
- [13] Mehler, A., Baumartz, D., Henlein, A., and Hemati, W.: 'fastSense: An EfficientWord Sense Disambiguation Classifier', in Editor (Ed.)^(Eds.): 'Book fastSense: An EfficientWord Sense Disambiguation Classifier' (2018, edn.), pp.
- [14] Kazama, J.i., and Torisawa, K.: 'Exploiting Wikipedia as External Knowledge for Named Entity Recognition'. *Proc. Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, {Prague, Czech Republic2007 pp. Pages
- [15] Moro, A., Raganato, A., and Navigli, R.: 'Entity Linking meetsWord Sense Disambiguation: a Unified Approach', *Transactions of the Association for Computational Linguistics*, 2014, 2, (1), pp. 231-244
- [16] Aggarwal, N., and Buitelaar, P.: 'Wikipedia-Based Distributional Semantics for Entity Relatedness', in Editor (Ed.)^(Eds.): 'Book Wikipedia-Based Distributional Semantics for Entity Relatedness' (2014, edn.), pp.
- [17] Syed, Z., Finin, T., and Joshi, A.: 'Wikipedia as an Ontology for Describing Documents', in Editor (Ed.)^(Eds.): 'Book Wikipedia as an Ontology for Describing Documents' (2008, edn.), pp.
- [18] Schonhofen, P.: 'Identifying document topics using the Wikipedia category network', *Journal of Web Intelligence and Agent Systems*, 2009, 7, (2), pp. 195-207
- [19] Medelyan, O., Milne, D.N., Legg, C., and Witten, I.H.: 'Mining meaning from Wikipedia', *International Journal of Human-Computer Studies*, 2009, 67, (9), pp. 716-758
- [20] McCrae, J.: 'MappingWordNet Instances to Wikipedia', in Editor (Ed.)^(Eds.): 'Book MappingWordNet Instances to Wikipedia' (2018, edn.), pp.
- [21] Abdulgabbarsaif, Omar, N., Zainodin, U.Z., and Aziz, M.J.A.: 'Building Sense Tagged Corpus Using Wikipedia for Supervised Word Sense Disambiguation', *Procedia Computer Science*, 2018, 123, pp. 403-412
- [22] Galárraga, L., Symeonidou, D., and Moissinac, J.-C.: 'Rule Mining for Semantifying Wikilinks'. *Proc. Linked Data on the Web (LDOW2015)*, Florence, Italy2015 pp. Pages
- [23] Roberto Navigli, P.V.: 'Structural semantic interconnections a knowledge-based approach to word sense disambiguation', *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2005, 27, (7), pp. 1075 - 1086
- [24] Lesk, M.: 'Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone', in Editor (Ed.)^(Eds.): 'Book Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone' (1986, edn.), pp. 24-26
- [25] Araujo, S., Houben, G.-J., and Schwabe, D.: 'Linkator: Enriching Web Pages by Automatically Adding Dereferenceable Semantic Annotations'. *Proc. International Conference on Web Engineering2010 pp. Pages*
- [26] Agrawal, R., Gollapudi, S., Kannan, A., and Kenthapadi, K.: 'Identifying enrichment candidates in textbooks', in Editor (Ed.)^(Eds.): 'Book Identifying enrichment candidates in textbooks' (2011, edn.), pp. 483-492
- [27] Klímek, J., Škoda, P., and Nečaský, M.: 'Survey of tools for Linked Data consumption', *Semantic Web*, 2018, pp. 1-57
- [28] Nentwig, M., Hartung, M., Ngomo, A.-C.N., and Rahm, E.: 'A survey of current Link Discovery frameworks', *Semantic Web*, 2017, 8, (3), pp. 419-436
- [29] Ngomo, A.-C.N., and Auer, S.: 'LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data', in Editor (Ed.)^(Eds.): 'Book LIMES - A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data' (2011, edn.), pp.
- [30] Achichi, M., Bellahsene, Z., Ellefi, M.B., and Todorov, K.: 'Linking and disambiguating entities across heterogeneous RDF graphs', *Journal of Web Semantics*, 2019, 55, pp. 108-121
- [31] Volz, J., Bizer, C., Berlin, F.U., Gaedke, M., and Kobilarov, G.: 'Silk – A Link Discovery Framework for the Web of Data', in Editor (Ed.)^(Eds.): 'Book Silk – A Link Discovery Framework for the Web of Data' (2009, edn.), pp.
- [32] Scharffe, F., Liu, Y., and Zhou, C.: 'RDF-AI: an Architecture for RDF Datasets Matching, Fusion and Interlink', in Editor (Ed.)^(Eds.): 'Book RDF-AI: an Architecture for RDF Datasets Matching, Fusion and Interlink' (2009, edn.), pp.

- [33] Nikolov, A., Uren, V., Motta, E., and Roeck, A.d.: 'Integration of semantically annotated data by the KnoFuss architecture', in Editor (Ed.)^(Eds.): 'Book Integration of semantically annotated data by the KnoFuss architecture' (2008, edn.), pp.
- [34] Hassanzadeh, O., and Consens, M.P.: 'Linked Movie Database', in Editor (Ed.)^(Eds.): 'Book Linked Movie Database' (2009, edn.), pp.
- [35] Raimond, Y., Sutton, C., and Sandler, M.: 'Automatic Interlinking of Music Datasets on the Semantic Web', in Editor (Ed.)^(Eds.): 'Book Automatic Interlinking of Music Datasets on the Semantic Web' (2008, edn.), pp.
- [36] Olfa Ben Said, A.W., Adel M. Alimi: 'Interlinking video programs with Linked Open Data'. Proc. 15th International Conference on Intelligent Systems Design and Applications (ISDA)2015 pp. Pages
- [37] Hausenblas, M., Troncy, R., Raimond, Y., and Bürger, T.: 'Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments', in Editor (Ed.)^(Eds.): 'Book Interlinking Multimedia: How to Apply Linked Data Principles to Multimedia Fragments' (2009, edn.), pp.
- [38] Rajabi, E., and Greller, W.: 'Exposing Social Data as Linked Data in Education', International Journal on Semantic Web and Information Systems 2019, 15, (2), pp. 92-106
- [39] Dezhao Song, Y.L., Jeff Heflin: 'Linking Heterogeneous Data in the Semantic Web Using Scalable and Domain-Independent Candidate Selection', IEEE Transactions on Knowledge and Data Engineering, 2016, (99)
- [40] Hausenblas, M., Halb, W., and Raimond, Y.: 'Scripting User Contributed Interlinking', in Editor (Ed.)^(Eds.): 'Book Scripting User Contributed Interlinking' (2008, edn.), pp.
- [41] James N. K. Liu, Y.-L.H., Edward H. Y. Lim, Xi-Zhao Wang: 'A New Method for Knowledge and Information Management Domain Ontology Graph Model', IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2013, 43, (1), pp. 115 - 127
- [42] Sun, H., Fan, W., Shen, W., and Xiao, T.: 'Ontology Fusion in High-Level-Architecture-Based Collaborative Engineering Environments', IEEE Transactions on Systems, Man, and Cybernetics, 2013, 43, (1), pp. 2 - 13
- [43] Zhao, L., and Ichise, R.: 'Ontology Integration for Linked Data', Journal on Data Semantics, 2014, 3, (4), pp. 237–254
- [44] Matinfar, F., Nematbakhsh, M., and Lausen, G.: 'Web Resource Sense Disambiguation in Web of Data', Journal of Universal Computer Science, 2013, 19, (13), pp. 1871-1891
- [45] Bloehdorn, S., Hotho, A., and Staab, S.: 'An Ontology-based framework for text mining', GLDV Journal for computational linguistics and language technology, 2004, 20, (1), pp. 87-112
- [46] Rajpathak, D.G., and Singh, S.: 'An Ontology-Based Text Mining Method to Develop D-Matrix From Unstructured Text', IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2014, 44, (7), pp. 966 - 977
- [47] Lund, K., and Burgess, C.: 'Producing high-dimensional semantic spaces from lexical co-occurrence', Behavior Research Methods, Instruments, & Computers, 1996, 28, (2), pp. 203–208
- [48] Ponzetto, S.P., and Navigli, R.: 'Knowledge-rich Word Sense Disambiguation rivaling supervised systems', in Editor (Ed.)^(Eds.): 'Book Knowledge-rich Word Sense Disambiguation rivaling supervised systems' (2010, edn.), pp. 1522-1531
- [49] Fogaroli, A.: 'Word Sense Disambiguation Based on Wikipedia Link Structure'. Proc. IEEE International Conference on Semantic Computing2009 pp. Pages
- [50] Mihalcea, R.: 'Using Wikipedia for Automatic Word Sense Disambiguation', in Editor (Ed.)^(Eds.): 'Book Using Wikipedia for Automatic Word Sense Disambiguation' (2007, edn.), pp.
- [51] Strube, M., and Ponzetto, S.P.: 'WikiRelate! computing semantic relatedness using Wikipedia', in Editor (Ed.)^(Eds.): 'Book WikiRelate! computing semantic relatedness using Wikipedia' (2006, edn.), pp. 1419-1424
- [52] Csomai, A., and Mihalcea, R.: 'Linking documents to encyclopedic knowledge', IEEE Intelligent Systems, 2008, 23, (5), pp. 34 - 41
- [53] David Tomýs, Y.G., Francisco Agullý: 'Entity linking in media content and user comments: Connecting data to wikipedia and other knowledge bases'. Proc. eChallenges e-2015 Conference2015 pp. Pages
- [54] Zesch, T., Müller, C., and Gurevych, I.: 'Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary', in Editor (Ed.)^(Eds.): 'Book Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary' (2008, edn.), pp. 1646–1652