Disambiguating named entities by semantic web

Ideh Azari
Department of Computer, Islamic Azad University
Mobarakeh
Mobarakeh branch, Islamic Azad University
Mobarakeh
Isfahan, Iran
Email: ide.azari@mau.ac.ir

Fateme Koohpeyma
Department of Computer, Islamic Azad University
Mobarakeh
Mobarakeh branch, Islamic Azad University
Mobarakeh
Isfahan, Iran
koohpeyma@mau.ac.ir

***Abstract*—There are many words in natural language sentences which describe an object or event in real world. An important step to understand a sentence is finding the exact meaning of all words. In this paper we propose an approach to identify meaning of each named entity of a text by using web of data (LOD) as a large scale knowledge base. The proposed approach is referencing named entities to suitable URIs of web of data. It compares the context of the text with RDF graphs retrieved from the web and then select URIs which have maximum agreement with named entities of the text. Experimental Results show that our approach outperforms previous works in terms of accuracy and Precision on Wikipedia articles.**

Keywords- linked open data, named entity disambiguation, natural language processing, semantic web

I. INTRODUCTION

Understanding natural language sentences (e.g., English, French or Persian) is very complication. Ambiguity of words is one of the main reasons for this complication. Even though human brain can simply disambiguate such words, identifying the correct meaning of words by a machine can be a considered a great significant task.

In this work we have focused on named entities disambiguation. Named entity can be one or more words that describe an object (person, location, organization …) in real world. For example, the following sentences about "Wyler" can be found in Wikipedia articles which describe four different persons.

***Wyler** is a neurosurgeon and author.*

***Wyler** is a Brazilian translator.*

***Wyler** was a British-born American actor and writer.*

***Wyler** was a leading American motion picture director, producer, and screenwriter.*

Most of the approaches for Named Entity Disambiguation (NED) have used context information extracted from the words around a named entity in a text. In such approaches, knowledge base has effective role on accuracy and precision of NED. Knowledge base offers usable data to find suitable referent for named entities. NED process creates precise results if they obtain a comprehensive knowledge base.

In recent years, NED methods use WWW webs as a knowledge base [1]. These approaches have many limitations due to different web formats, no typed links, and transforming web documents to machine readable format.

As an example [2] has disambiguated by computing similarity between context information extracted from Wikipedia and the text. [3] use a technique to enrich a given Wikipedia articles term with words from article in same category. [4] transforms a text to a list of surfaces and for all surfaces extracts means and features from Wikipedia articles and use a random graph walk model to derive a relatedness score. All of these techniques have not been a practical solution and have tried to overcome the limitations of traditional web.

A new technique called Linked Of Data (LOD), have been proposed by Tim Berners-Lee to set a best practice for publishing and connecting structured data on the Web [5]. LOD is a high scale machine readable knowledgebase than can be used as an open domain encyclopedia.

[6] use LOD for named entity classification and make a knowledge base for scoring the entities from data extracted of web of data. NERD [7] uses it to named entity recognition.

In this paper, we propose an approach to named entity disambiguation by using LOD as a high scale and integral encyclopedia. The reminder of the paper is organizing as follow: section 2 describes named entity recognition. Section 3 describes how URI retrieved from LOD. Section 4 and 5 produced an overview for NED in sentence and whole text and section 6 and 7 conclude experimental results and conclusions.

II. NAMED ENTITY RECOGNITION

We used Alchemy API for named entity recognition before performing NED process on the text.

Named entities are sorted based on their location in the text and then we must identify number of occurrences of

each one in the text. This method has to sort named entities because it needs to disambiguate named entities, sentence by sentence and if a named entity occurs in two sentences, it is disambiguated by context of each sentences and effect on another named entities in that sentence. Therefore if a named entity in two locations has two different references, this method can identify it. For example, there are four named entity in this sentence: "**Ingmar Bergman** was born in **Uppsala**, **Sweden**, the son of **Erik Bergman**, a Lutheran minister and later chaplain to the King of **Sweden**." Bergman occurs in two locations and has two different references.

III. Information Extraction From Linked Data

We retrieve relative URIs to named entities by Sindice, a search engine of web of data. Indeed we query on the web of data by Sindice SPARQL endpoint. Each query is generated based on each named entity and its type in the text.

Retrieved URIs by these queries are RDF files with triples that usually are URI as well and should be compared with a text that just contains some words (literals). Therefore, all of results are saved in a new format with a URI. This URI is a reference to the named entity and other URIs change to one of its objects base on its properties. Following code shows a triple that describe William Wyler. Subject, object and predicate in this triple are a URI that cannot be compared with context of the text.

```
<rdf:Description
rdf:about=http://dbpedia.org/resuorce/john_f
ord>
<Dbpprop:after rdf:resource="http://
dbpedia.org/resuorce/William_Wyler"/>
</rdf:Description>
```

In this triple, object is a reference to William Wyler (named entity in the text). Subject is a reference to john ford and its RDF:type is person. Then change this triple to:

```
<dbpprop:after>John Ford</dbpprop:after>
```

Value of property of foaf:name is instead of ford's URI.

In the end of this step, there is an integral knowledge base contain RDF files with a unique format and usable for next step to disambiguate named entities.

IV. Disambiguation Of Named Entities In A Sentence

The first step is to survey a sentence and make a primary result based on context and named entities of a sentence. Assume $\overline{NE}$ is a vector of named entities in the sentence and is sorted by location of each named entity.

$$\overline{NE} = \{ne_1, \dots, ne_m\} \quad (1)$$

$$\overline{R_m} = \{r_1, \dots, r_l\} \quad (2)$$

$\overline{R_m}$ is a vector of results and is generated by querying on the LOD.

$r_1{}^m$ is lth obtained result for NE_m

V. Similarity Vector

Vector is computed by the context of a sentence which contains a named entity. Because words of a sentence usually have highest relativity together and when two words are separated further from each other, relativity between them will be less.

In this step, we generated a score for each element of a named entity vector. If the saved results for each named entity have some redundancy, the redundancies must be filtered and also the other results are filtered according a threshold. Then score of remaining results are normalized.

$$\bar{S} = \{s_1, \dots, s_l\} \quad (3)$$

Where $$S_l^m = \frac{\frac{score_i}{max_{i \in l}\, score} - T}{T}$$

S_l^m is an element of similarity vector that compute for each named entity in the sentence.

Correlation Matrix

With this assumption that all of the words in a meaningful sentence have semantic correlations, we use another parameter to compute relativity score with higher accuracy between each obtained URI of LOD and context of the text. This parameter is correlation matrix.

To compute correlation matrix elements, we computed correlation between RDF graphs that characterize URI result of NE_i and RDF graphs that characterize URI result of NE_j. Correlation of these two graphs is corresponding to number of direct links between them and also number of links to the same concept, entity or URI.

If r_l and $r_{l'}$ are two elements of two result vectors, o_l^i and $o_{l'}^j$ are two object in those result. Then $L(o_l^i)|_j$ is 1 if two URI's link to the same entity.

$$L(o_l^i)|_j = \begin{cases} 1 & o_l^i = o_{l'}^j \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Assume L (i, j) as number of r_l of NE_i to $r_{l'}$ of NE_j , then each element of correlation matrix compute as:

$$s_{mn}^{corr} = L_{\langle m,n \rangle} + L_{\langle m,n \rangle} + \sum_{r_i \in \overline{R_m}, r_i \in \overline{R_n}} L(o_l^i)|_j \quad (5)$$

The dimensions of this matrix are $l \times l'$.

Both normalized value of elements of this matrix and similarity vectors are used to compute relevance of each named entity.

VI. Sentence Relevance Vector

We generated a matrix of each element couples for two result vectors. In this matrix, rows are result vector of first named entity and columns are result vector of second named entity. Each elements of this matrix are computed as:

$$\delta_{ij} = \frac{\frac{s_{ij}^{corr} + s_i + s_j^{ne}}{3} - \min_{ij} S}{\max_{ij} S - \min_{ij} S} \quad (6)$$

Then all elements with a score less than threshold are eliminating:

$$MX_{ij} = \begin{cases} \delta_{ij} & \delta_{ij} > Threshold \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

If all elements in a row or a column of the matrix are zero, there is no agreement with any result of neighbourhood named entities. If the agreement of this URI with the context of sentence is less than acceptable value, URI is removed from result vector of this named entity.

None zero elements of this matrix make a new similarity vector and will be combined with result vector of next named entity. This procedure will be continued to compute the final similarity vector of this sentence.

Figure 1 shows part of computations similarity vector for four named entity in a sentence. The reference URIs for these named entity are:

Ingmar Bergman: http://dbpedia.org/resource/Ingmar_Bergman

Uppsala: http://www.geonames.org/2666199

Sweden: http://www4.wiwiss.fu-berlin.de/factbook/page/Sweden

Erik Bergman: http://dbpedia.org/resource/Erik_Bergman_%28Lutheran_minister%29

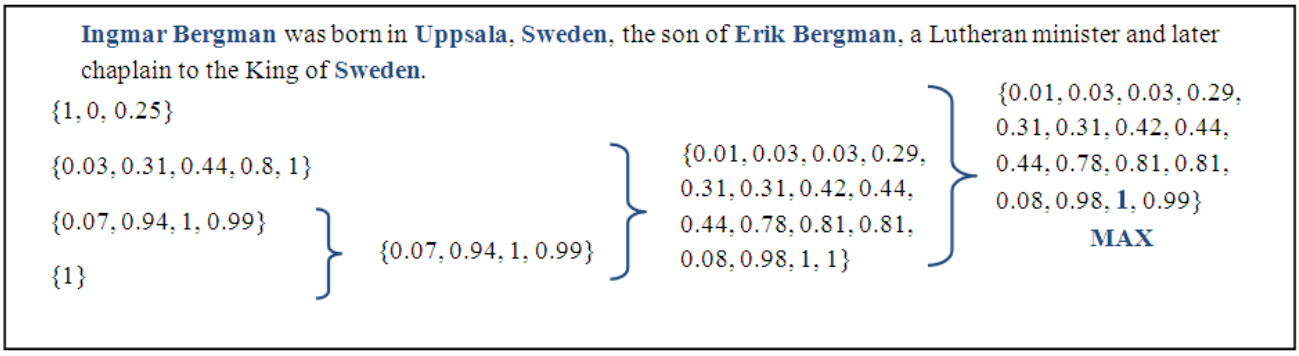

Figure 1. Example of compute final result vector in a sentence

The vector with 16 elements is the final similarity vector for this sentence. An element with maximum score is referring to a set of URIs that is best referent for these named entities in LOD.

In this method, the result vector of a named entity and the computed result vector in the past stage are combined and create a matrix. The results with the least relevance sentences are filtered and removed. In this method, the length of result vector never become so large and its length always are in an acceptable range and therefore computing time is small.

VII. NED In All Sentences of the Text

The similarity vector from the first sentence determines the relevance between each set of URIs. This vector is used as the first similarity vector for text and is combined with next sentence vector by the method that was discussed in the last section. Similarity vectors of the two first sentences are combined and a new set of result is obtained. Each set combining this result with the next sentence continues to end of text.

Assume number of disambiguated named entity is N and number of a sentence named entity is n. Each sentence similarity vector effect on text similarity vector by difference weights. This weighs are computed by comparing n and N.

If s_i be an element of similarity vector of text and s_j be an element of similarity vector of new sentence, then each element of similarity matrix compute as:

$$\delta_{ij} = \frac{\frac{s_{ij}^{corr} + ((1 - w_{ij}) \times s_i) + (w_{ij} \times s_j^{ne})}{2} - \min_{ij} \delta}{\max_{ij} \delta - \min_{ij} \delta} \quad (8)$$

Where $w_{ij} = \frac{n}{n+N}$

After disambiguating all sentences in text, we obtained a set of URIs with high score in final similarity vector. This set has the URI references to web of data for each named entity in text with highest agreement.

VIII. Evaluation

To evaluation our approach, we used a dataset that is randomly selected from Wikipedia English documents.

DBpedia is a semantic version of Wikipedia. Each Wikipedia article has a URI reference in DBpedia datasets. Triples in each URI include same information in the articles. The named entities in these articles are a subject or object in triples of the URIs. Each DBpedia URI has triples with OWL:sameas properties that is linked to other sources of LOD.

The number of entities in each article is between 1 and 52 and totally there are 519 named entities in these Wikipedia articles. For 55 named entities (10%), Sindice cannot found any URI reference. 374 named entities (91.66%) have correct URI reference and 39 others ones (8.44%) have not suitable URI reference.

In 70% of correct results (26 named entity), there is not any URI in set of final results. This means that all URI founded by Sindice have not enough information to compare with context of text for selecting suitable URI reference. For other incorrect results (13 named entity), a URI is found in final results set but is not a correct reference.

Therefore in 2% of all results, they guide user to false URI references and 98% of URI in results set are correct reference for named entities of text.

Table 1 shows comparison accuracy of our method with related work that uses Wikipedia articles for evaluation.

IX. Helpful Hints

In this paper we propose an approach for NED by using LOD as an encyclopaedia. Experiments on a set of randomly selected Wikipedia documents showed that this approach has a high accuracy for NED. The proposed solution references to about 91.66% of named entities in a text to a suitable URI referent.

The available domain articles in experimental set (e.g. mathematic, sports, religion, politics, movie) showed LOD can be used as a open domain and high scale knowledge base system.

TABLE I. ACCURACY COMPARISION

method	Maximum accuracy
Cucerzan	88.3%
Gentile et al	89.83%
Our method	91.66%

REFERENCES

[1] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[3] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] K. Elissa, “Title of paper if known,” unpublished.

[5] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[8] Electronic Publication: Digital Object Identifiers (DOIs). Article in a journal:

[9] D. Kornack and P. Rakic, “Cell Proliferation without Neurogenesis in Adult Primate Neocortex,” Science, vol. 294, Dec. 2001, pp. 2127-2130, doi:10.1126/science.1065467. Article in a conference proceedings:

[10] H. Goto, Y. Hasegawa, and M. Tanaka, “Efficient Scheduling Focusing on the Duality of MPL Representatives,” Proc. IEEE Symp. Computational Intelligence in Scheduling (SCIS 07), IEEE Press, Dec. 2007, pp. 57-64, doi:10.1109/SCIS.2007.357670.