Disambiguation of Named Entity with Supervised Technique over a Knowledge Base

Aarthi D, Viswanathan V

Abstract— Named Entity Disambiguation (Entity linking) is the task to link the entity mentioned in the query search with the appropriate entity in the repository without any name disambiguation. It can facilitate many tasks such as list of people or population in the repository and query processing and information retrieval. This task is a very challenging because of ambiguity and name conventions. In this paper we address a problem of named entity matching. In order to overcome this challenge we use Query search technique that implemented here is Name Dictionary based technique. The search key is extracted and compared with all the keys from the dictionary and the appropriate value is fetched and system throws as a result. Entity linking provides the information both explicitly and implicitly. Explicit linking provides the information beyond the knowledge base whereas implicit linking provides the information only from the knowledge base. Based on the information obtained we can also add the ratings and the comments. Based on the comments and the ratings the data that provided in the repository can also be managed. Our experiments shows the promising results in extracting the Candidate entities and graph based outcome if the user performs sequence of query search on single namesake.

Keywords: Named Entity, Candidate Entity, Disambiguation, Explicit Linking, Implicit Linking.

1. INTRODUCTION

The amount of information over web is increasing day to day proportionally amount of web data is also increasing. Large amount of data that available over the internet is in the form of Natural Language. Natural Language is highly ambiguous especially with the namesakes (i.e. different persons having same names). A named entity may have multiple names and a name could denote several different named entities. Knowledge sharing resources like Wikipedia have provided a large automated machine understandable knowledge bases. As too many people associated with single name it is a very hard task to differentiate the individuals. For example a single name may be associated with singer, football player, basketball player, actor etc. So it will be a hard task to differentiate one among them. A critical step to achieve this goal is to link named entity mentioned in the query search with the appropriate entity in the repository without any name disambiguation. [1] It can facilitate many tasks such as list of people or population in the repository and query processing and information retrieval.

Query search text field can accept both the query and the named entity alone. If the passed text field is query it extracts the named entity and matches with key pairs and provides the result.[2] [3] The data is maintained as a dictionary. Where the data is stored as a key value pairs. The namesake data can be inserted into the dictionary (table) as individual records by providing the key, value. Otherwise the collected namesake data is pre-processed and stored as a text file with the collection of key value pairs of namesake data separated by some delimiter. Now this processed data which is like a text file can be proceeded for bulk import of the namesake data. Instead of inserting the individual data by the bulk import bulk amount of data can be inserted at single step. The search technique that used will provide the resultant data along with the count of total population, candidates associated with search key and candidate search ratio. The result that obtained will be displayed as graph based on the ratio of population that present on the particular name and profession (i.e. particular namesake) data.

The rest of the paper is ordered as follows. In section 2 we discuss preceding works related to Name Entity recognition and knowledgebase. In section 3 we described the proposed system with candidate entity generation, Entity linking and building the knowledge base based on Name dictionary base technique with the real time dataset. Section 4 deals with the experimental discussions and how the search ratio will be calculated, resultant graph generation and experimental testing of the application. Section 5 deals with how the results are obtained and the screenshots of the results. Section 6 present conclusion and future works.

2. RELATED WORK

In Wikipedia when the disambiguation named entities occurs based on the page ranking which entity is referred mostly to that particular named entity the page will be redirected and the other name entities or namesakes are provided as a referable links in some special cases. When multiple entities in Wikipedia could be given the same name, a disambiguation page is created with all the named entities related to that particular name and when the query search made with that name or named entity it redirects to the disambiguation page and provides the result. It results in much time for the processing of that particular query because of page to page traversal.

2.1. Named Entity Recognition

Named entity recognition is a sub task in query processing and recognizing a named entity from the query itself will be the hard task. One of the previously proposed system mainly

Revised Version Manuscript Received on March, 25, 2019.

Aarthi D, School of Computing Science and Engineering, VIT University, Chennai, Tamilnadu, India

Viswanathan V, School of Computing Science and Engineering, VIT University, Chennai, Tamilnadu India

focused identify only the named entity[2] which is of noun form. But single name would be associated to many people. At this case disambiguation occurs in Named entity recognition. Other proposed system involves identifying of named entity and classification into predefined categories such as name of person, organization, locations etc. But this performs based on query log and topic model [4] [5] which stores log data.

In query processing it scans the query and using Parts of Speech (POS) tagger[6] that is it identifies all the parts of speech of each word in the input query and extracts the patterns and using patterns it extracts the named entities. But in this proposed system without using any external libraries the input query will be divided into tokens and these tokens are stored as array of strings and all the stop words from the string array will be removed and only the main key word will be searched or compared with the keys from the dictionary.[7]

2.2. Expert System

Expert systems or Knowledge base is a basic component in the entity linking. It provides the information about the world entities and the relations between them. One of the existing system focuses on generating the expert system where the data is created by the ontologies. [8] But here in this proposed system without the creation of ontologies the namesake data is collected from the various sources and pre-processed. The namesake data that obtained is stored as the key value text file. Were the key is name followed by the profession and value will be the textual entity. These key values are separated by the delimiter. Thus created text file acts as the data set and it is then imported to the dictionary that created. One of the expert system Wikipedia provides a lot of data with respect to namesake data. It maintains a disambiguation pages for the namesake data in few special cases. If a query search is performed over that data it fetches and identifies as ambiguous data and redirects to the disambiguation page which takes a lot of time.

In contrast to overcome this issue in this paper we propose Name Dictionary based technique where entity linking system will hold all the combinations of redirected pages and disambiguation pages as a dictionary between the various names and all possible mapping entities. Thus using the dictionary the query that passed for the search will generate the appropriate candidate entity. The dictionary that made will contain the data or information as (Key, value) pairs.[9] [10] Where key is the name of the person followed by the designation and the value is the content of that particular named entity. The search query will hit the dictionary to particular key value and the associated text content to that particular key will be fetched and displayed.[1] [11]

3. PROPOSED SYSTEM

Named Entity Disambiguation can facilitate many tasks such as list of people or population in the repository and query processing and information retrieval. The main task is to link the entity mentioned in the query search with the appropriate entity in the repository without any name disambiguation. The query that passed for the query search technique, initially the query that entered is read undergoes trim then performed the tokenization[12] and the tokens that extracted are stored in the string array then performed the stop word removal [13][14] from the array of strings. By removal of the stop words we can extract the actual search key.

The extracted search key is then matched with the dictionary key and the appropriate value that associated with that key from the dictionary will be fetched and result will be displayed. That is if the key associated with only one value in the dictionary then only that value is displayed. Else if key associated with three values in the dictionary then those three values are fetched and displayed as result. Along with the data the total amount of the population in the repository (as Total candidates), the total no of candidates that are associated with the search key (as Total search candidates found) and also the candidate search ratio.[1]

Entity linking provides the information both explicitly and implicitly. Explicit linking provides the information beyond the knowledge base i.e. provides the external link associated to the topic and can view the additional information by navigating the provided link whereas implicit linking provide the information only from the knowledge base i.e. provides the data that is associated with the topic from the repository here also the external link will be provided but the user cannot navigate. The link will not be enabled here. Based on the information obtained we can also add the ratings and the comments. Based on the comments and the ratings the data that provided in the repository can also be managed. By managing it we can ensure the accuracy of the data that provided in the link. If the ratings to the specified URL are low or less then we can update or change or modify the specific topic URL and provide the apt data.

The input query is accepted from the “search candidate” option the query that obtained is undergone with series of actions for the query processing. After the series of steps performed the result (candidate entity) will be displayed. Along with the candidate entity Total candidates, Total search candidates, candidate search ratio will also be displayed.

The system mainly goes with the three stages.

- Candidate Entity Generation.
- Knowledge Base.
- Entity Linking.

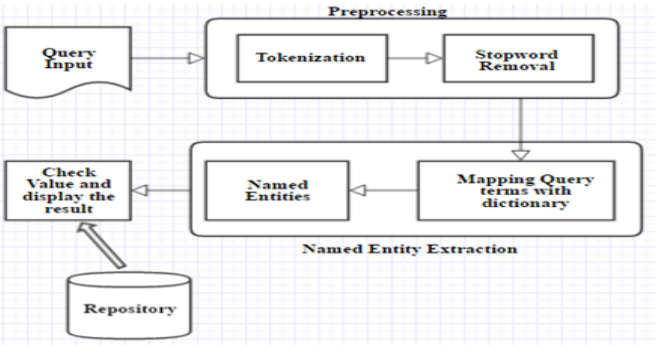
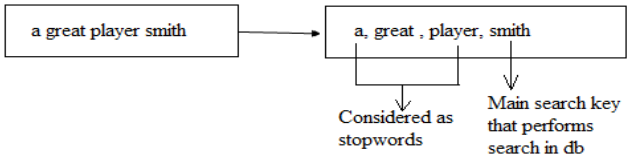


Fig 1. System Architecture

3.1 Candidate Entity Generation

In Candidate entity generation first the input query is obtained and performed the series of actions i.e. Trim, String array and Stop words removal. Last extract the actual word & Compare. By performing the trim the spaces will be removed and splits as individual words that are stored as string array. Now this string array is subjected to stop words removal and finally the named entity (actual word) is obtained. Thus the obtained key word is then subjected to compare with dictionary that created. The named entity hits the dictionary as a key and the associated text to the key will be fetched as a value from the dictionary and the result will be displayed.[15] [16][17]



We can also pass queries like:

1. What about a famous person Sachin.
2. A great singer
3. A famous football player etc.

Whatever the query may be passed it follows the above steps as mentioned and extracts the exact key word and matches that key word with in the database and provides the actual result.

3.2 Knowledge Base

Knowledge base contains the named entities and text collection with the specified named entities. Knowledge base may have same name entity but different textual entity. For example, as shown in Table1. Knowledge base will be in the form of dictionary as a key value pair. The technique used is Name Dictionary Based technique. The named entity column will act as a key and the associated text column will acts as a value. If k1 is the key that fetched the associated k1.value will be displayed as a result. It consists of multiple data associated with same name and also multiple persons with same profession.[18] [19][20]

Name Entity	Textual Entity
Michael Jordan	Michael Jordan a football player born in 1970
Michael I. Jordan (professor)	Michael Jordan a professor in reputed university
Michael Jordan (scientist)	Michael Jordan a professor and an American scientist.
Michael Jordan (mycologist)	Michael Jordan is an English mycologist, author of Encyclopaedia.

Table1. Namesake information of a particular person

The data that added to the knowledge base in two ways:

a) Inserting the record individually or
b) By the Bulk Import of data.

It is the Candidate entity generation duty to map with the exact one and that exact results are the candidate entities.

3.3 Entity Linking

The main goal of the Entity linking is to map each textual entity to its corresponding name entity in the knowledge base. Some entity mentioned in the text does not have corresponding entity record in the knowledge base such kind of entities will be act as unlinked entities and given as null. In the entity linking the user has two options:

3.3.1 Explicit Entity Linking

Explicit linking provides the information beyond the knowledge base. Based on the topic chosen from the list of topics like (petroleum, coal) and the country like (India, USA) the information regarding the chosen topic and the URL with respect to that topic will be viewed. The URL can be accessed or viewed provided the internet net availability.

3.3.2. Implicit Entity Linking

Implicit linking provide the information only from the knowledge base. Based on the topic chosen the information that present in the database (or) repository will only be viewed. It also shows the URL but the URL can't be accessed. In entity linking based on the information obtained the user can provide the comments and the ratings to the respective topic that made for search. . Based on the comments and the ratings the data that provided in the repository can also be updated or deleted.

4. EXPERIMENTAL DISCUSSIONS

The application is developed in java environment under JSP (Java Servlet Pages) [21]. Sqlyog is the database tool [22] where all the database tables sit along with the offline dictionary table. In Candidate Entity generation the query that passed will processed and generates the candidate entities. Along with that it also displays the values of Total Candidates, Total Search Candidates found and also the Candidate Search Ratio (CSR).

$$CSR = \frac{\text{Total Search Candidates found}}{\text{Total Candidates}} * 100 \qquad \text{(Eq.1)}$$

f the user performs the search on a single namesake data consecutively the consecutive search results will be stored in the database table and this can be viewed as a plotted graph over that particular namesake. These results are generated by the graph based approach under the supervised ranking method. [23]

Most of the named entities are disambiguated. For example a named entity "SUN" can refer to the star in solar system, "Sun Developer Network" who developed java and "Sun-Hwa Kwon" a fictional character on television series. It is very hard to differentiate this kind of named entities. These kind of ambiguous data can be easily disambiguated by the proposed technique. By using the proposed technique the

experiments generated the disambiguated candidate entities.

Using Selenium IDE the experiments had also made to generate the test scripts for the all the modules. All the test scripts ran successfully and achieved the appropriate results. [24] The application holds good only if the named entity data available in English language. It does not support for the other languages. The application does not need any existing data sets as the data is added manually. The application holds good only if the data that inserting into the dictionary is well pre-processed. The application generates the accurate result as both the name and the profession together considered as a key. It also generates the accurate candidate ratio for the search query. The application also generates absolute graph for the consequent search of a single namesake data.

5. RESULTS

Based on the query that submitted is processed by the Name Dictionary Based technique and named entity is extracted from the query. The extracted named entity will hit the dictionary table that maintained and compares the named entity with all the (keys). The appropriate (key. Value) that matched with the extracted named entity is fetched as a result as shown in (Fig.3).

In entity linking (explicit, implicit) based on the search topic and country specified by the user will be compared with the data table and associated data is fetched as result and then navigated to ratings and comment page as shown in (Fig 4,5,6,7). On the consequent search of the same namesake data will be stored in the table. Considering that table based on the name and the professions the result will be displayed as graph based on the ratio of population that present on the particular name and profession as shown in (Fig. 8).

Enter Query Text What about a famous persc

Search Candidate Entities Reset

Fig 2. Search Candidate

The figure (Fig 2) depicting the query text that entered. [What about a famous person Sachin]

Candidate Entities Details

Matched Candidate Entity	Sachin Tendulkar
Matched Candidate Entity	sachin (singer)

TOTAL CANDIDATES=58.0
TOTAL SEARCHED CANDIDATES FOUND=2.0

CANDIDATE SEARCHED RATIO::3.4482758620689653

Fig 3. Search Candidate fetched result.

Based on the search performed the appropriate value from the repository is fetched as shown in the figure and also displays the total candidates, total searched candidates found and candidate search ratio.

The Fig 4. is depicting the outcome of explicit linking based on the topic selected that is (topic name- petroleum; country- USA). User can view the data externally by clicking the above link in image.

The Fig 5. Depicting that the user can provide the ratings and comments to the topic that viewed in explicit linking. (Ratings as: Very good; Good; Average; Poor.)

The Fig 6. Depicting the outcome of implicit linking based on the topic selected that is (topic name- petroleum; country- USA). User can view the data only internally by clicking the find more option. This only provides the data that available in the database. Here the URL will not be enabled.

SELECTED EXPLICIT TOPIC DETAILS

petroleum USA

Topic Name	Petroleum	Country Name	USA
Type	Oil	URL(Explicit Relationship)	http://www.gassigns.org/usapetro.htm
About Topic	USA Petroleum station at Barrington and San Vicente Blvd. in Brentwood, CA. All have huge American flags to attract attention. I believe the	Total Views (Rank)	23

Fig 4. Explicit Entity Linking

You selected for http://www.gassigns.org/usapetro.htm !!1 Do u Want to open site ??? Click Here -> http://www.gassigns.org/usapetro.htm

Select Parameter	Average
URL	http://www.gassigns.org/us
Mid	100
	Submit Reset

Fig 5. Rating for explicit linking.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Fig 6. Implicit Entity Linking.

Fig 7. Depicting the comment and rating form implicit linking.

Fig 7. Depicting that the user can provide the ratings and comments to the topic that viewed in implicit linking.

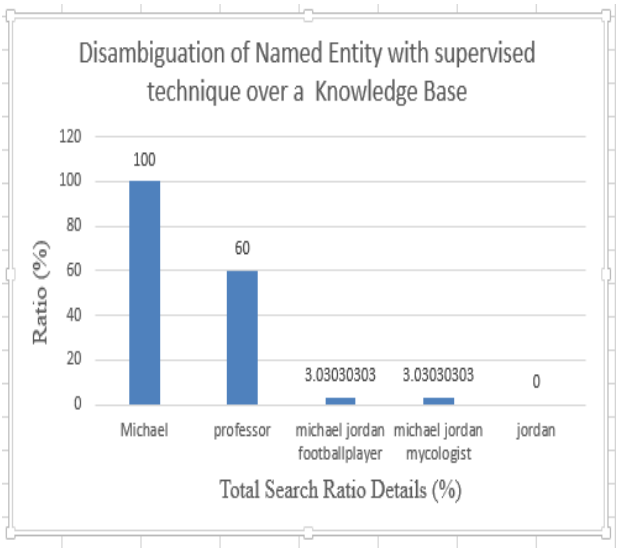


Fig 8. Result of consecutive namesake search.

The Fig 8. Depicting the outcome of the search on a single namesake data consecutively.

6. CONCLUSION AND FUTURE ENHANCEMENT

In this article we have implemented Candidate entity generation using Name Dictionary based technique for generating the candidate entity and we also implemented entity linking both explicitly and implicitly. We have implemented this by utilizing the three modules of entity linking (i.e. Candidate Entity Generation, Knowledge Base, and Entity Linking). Although there are many other techniques to deal with entity linking and candidate entity generation we have implemented by the above stated technique with the relevant data repository. We also generated the candidate ranking by the supervised ranking methods using the graph based approach for the candidate entities that formed. We point out some promising research direction in entity linking and candidate entity generation.

This application can be deployed over webserver and can be used as a search engine. The offline dictionary that maintained with the namesakes data is only in English language we can also make other dictionaries for other languages like German, French etc.

REFERENCES

1. W. Shen, J. Wang, and J. Han, “Entity linking with a knowledge base: Issues, techniques, and solutions,” IEEE Trans. Knowl. Data Eng., vol. 27, no. 2, pp. 443–460, 2015.
2. J. Guo, G. Xu, X. Cheng, and H. Li, “Named entity recognition in query,” Sigir, pp. 267–274, 2009.
3. [3] E. Agichtein and L. Gravano, “Snowball: Extracting Relations from Large Plain-Text Collections,” Proc. fifth ACM Conf. Digit. Libr. - DL ’00, vol. I, no. 58, pp. 85–94, 2000.
4. R. Wongso and D. Suhartono, “A Literature Review of Question Answering System using Named Entity Recognition,” pp. 274–277, 2016.
5. D. Nadeau and S. Sekine, “A survey of named entity recognition and classification,” Lingvisticae Investig., vol. 30, no. 1, pp. 3–26, 2007.]
6. Y. Sari, M. F. Hassan, and N. Zamin, “Creating extraction pattern by combining part of speech tagger and grammatical parser,” ICCTD 2009 - 2009 Int. Conf. Comput. Technol. Dev., vol. 1, pp. 515–519, 2009.
7. J. Zhu, “An adaptive approach for web scale named entity recognition,” Web Soc. 2009. SWS’09. 1st IEEE Symp., pp. 41–46, 2009.
8. A. Bellandi, S. Nasoni, A. Tommasi, and C. Zavattari, “Ontology-driven relation extraction by pattern discovery,” 2nd Int. Conf. Information, Process. Knowl. Manag. eKNOW 2010, pp. 1–6, 2010.
9. “Sporcle.” [Online]. Available: https://www.sporcle.com/games/Torgo/same-name-different-person.
10. “Namesake persons.” http://www.ebaumsworld.com/pictures/same-name-different-person/84311825/.
11. A. Moro, A. Raganato, and R. Navigli, “Entity Linking meets Word Sense Disambiguation : a Unified Approach,” Trans. Assoc. Comput. Linguist., vol. 2, pp. 231–244, 2014.

12. M. Tkatchenko, A. Ulanov, and A. Simanovsky, "Classifying wikipedia entities into fine-grained classes," Proc. - Int. Conf. Data Eng., pp. 212–217, 2011.
13. "Stopwords removal." http://stackoverflow.com/questions/27685839/removing-stopwords-from-a-string-in-java.
14. N. Phiwngam and T. Senivongse, "Knowledge Enhancement of Text and Visualization Based on DBpedia Dataset," Inf. Sci. Control Eng. (ICISCE), 2016 3rd Int. Conf., pp. 433–438, 2016.
15. P. G. Jose, S. Chatterjee, M. Patodia, S. Kabra, and A. Nath, "Hash and Salt based Steganographic Approach with Modified LSB Encoding," Int. J. Innov. Res. Comput. Commun. Eng., vol. 4, no. 6, pp. 2257–2263, 2016.
16. W. Wu, H. Li, H. Wang, and K. Q. Zhu, "Probase: A probabilistic taxonomy for text understanding," Proc. 2012 ACM SIGMOD …, pp. 481–492, 2012.
17. S. Mohammed, B. Abdellah, and O. El Beqqali, "based Tweet Entity Linking," pp. 3–9, 2016.
18. "same first names." https://www.quora.com/What-are-the-most-common-first-names-of-U-S-Presidents
19. "Famous people with same real names." [Online]. Available: http://www.ranker.com/list/famous-people-with-the-same-name/celebrity-lists.
20. "menta_floss." [Online]. Available: http://mentalfloss.com/article/58702/11-notable-people-who-shared-their-names-famous-contemporaries.
21. "Adding external jars", https://jsumon.wordpress.com/2009/11/24/adding-external-jar-or-library.
22. "Working with sqlyog", http://etutorials.org/Programming/PHP+MYSQL.
23. W. Shen, J. Wang, P. Luo, and M. Wang, "A graph-based approach for ontology population with named entities," Proc. 21st ACM Int. Conf. Inf. Knowl. Manag. - CIKM '12, p. 345, 2012.
24. "Testing with Selenium IDE." http://toolsqa.com/selenium-ide/download-and-install-selenium-ide/