Finding Aliases using Noun Detection Algorithm

Anuja Digambar Bharate
PG Student,
Department of Computer Engineering
JSPM's ICOER, Wagholi, Pune

ABSTRACT
Increased amount of internet users leads to collision in the names on the web forums and in many scenarios, which in turn leads to maximum number of users who are using their aliases in the web. This creates a difficulty in detecting the proper user. So, systems are suggested to identify their aliases using the entity graphs. But most of them are experimenting on the datasets; on the other hand very few systems exist to be worked on real entity. So implemented system put forward an idea of finding aliases on the real web data by using an enhanced web crawler which collects all sub URL's of the given seed URL, which is analyzed by the another baby crawler to fetch and parse the web data as human readable content using random walk relational theory. Alias graph is identified to be more efficient with the help of real relation entity graph on the collected web data.

General Terms
Pattern Recognition, Algorithms

Keywords
Web crawler, NLP, Entity relation, Random walk, Cauchy distribution.

1. INTRODUCTION
Since the information is wide over internet and various entities are found in documents like organization, products, location, festival name, people etc. Several applications like information retrieval, data mining, synonym generation, removing of duplicate names from the documents are fundamental for entity identification. In information extraction process, finding suitable information of given entity is a critical task. It will be infeasible to extract the person by using his name because sometimes person may have nick names also.

Dealing with entities leads to two main problems. First lexical ambiguity and second is referential ambiguity. Lexical ambiguity occurs when circumstances like entity share a same name and when one entity is referred by different names. People use alternate strings for some named entity. For example "Shahrukh Khan" also called as "srk". These alternate strings are called entity synonyms. If we ask for "DBMS" then "Database management system" may arise. Hence if proper identification of these aliases is done then final conclusion will contain more pages. Another important application of entity aliases identification is e-discovery. E-discovery is a process of collection, preparation and thus to produce a document in various criminal and civil cases. Consider an example which took place in one company where accident of microwave oven happened. In this case company has to open up to the investigation team all the needed information. In this type of cases, entities such as person, product and organization plays a vital role. Because these entities are described in reports, emails and another types of textual documents frequently. In such cases it is very challenging to find the long tail references of the entity. Here one thing is clear that investigation team does not want to leave a single reference also. These references may be engineer's name, accessories used when manufacturing of microwave shown in electronic documents.

Figure 1 give description of various real word entities present in different forms in documents. Left side entities are the basic one and right side are derived of that basic entities. Hence if aliases are not detected effectively then it leads to missing of evidences.

In entity extraction methods, a reference table is needed to maintain the aliases of the entity. Consider an example, if we are extracting a product name, very few chances are there for getting the accurate query string in the reference database. Hence aliases extraction systems should be give approximate matching. For alias identification, similarity matching approach can be used but it gives very poor efficiency. To overcome this drawback natural language processing techniques such as correlation is used. But for measuring correlation between the substring and database entity requires scanning of entire body. Because of this reason, it is not feasible to measure correlation when substring matching is done.

[3] Defines an approach for extraction of entities from the web. Author try to make it fully automated. The proposed system is based on the referential ambiguity which further combines with lexical pattern matching. For searching web pages, n based crawling method is used. [4] Describe an important approach for the discovery of entity synonyms. To overcome the previous hurdles, two novel similarity functions are used. Map Reduce framework is used to generate the large amount of synonyms. Also the approach is proposed to converts the long entity names into the short entity names.

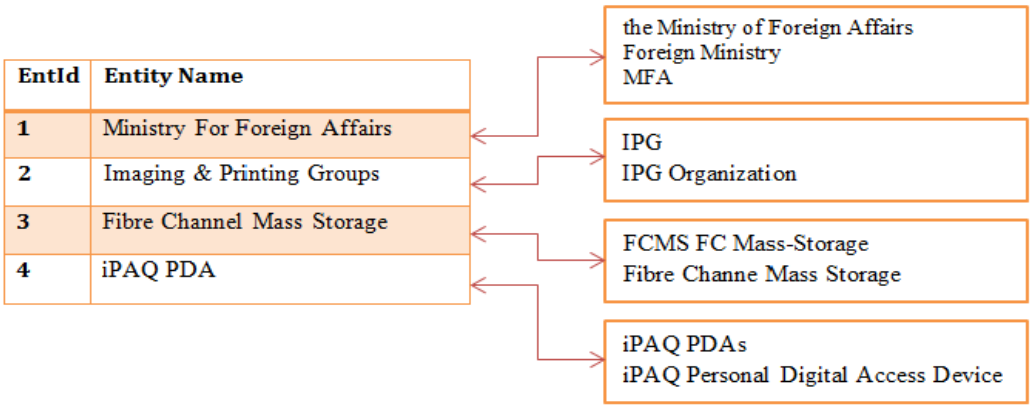

Figure 1: Real word Aliases examples [21]

The rest of the paper is organized as follows. Section 1 discusses some related work and section 2 presents the design of approach. The details of the results and some discussions have conducted on this approach are presented in section 3 as Results and Discussions. Sections 4 provide conclusion.

2. RELATED WORK

As the day passes large numbers of web pages are added to the internet. From this dynamic data useful information is extracted by using search engines. So the proper algorithms are needed to search useful information from this huge data. Web crawlers are the program or software's that are used to browse the WWW i.e. World Wide Web in efficient manner.

Due to nonstop changing nature of World Wide Web, it totally depends on the crawlers for its operations. By using web crawlers auto traversing of web is done and the links are followed by page to page. So it guarantees that no single page will get missed. [5] Express Focus: a supervised web-scale forum crawler that is use to parse likely forum content from web. It reports forum crawling problem to URL type recognition problem and shows how to implicitly leverage entry-index-thread path and design a method to learn ITF regexes explicitly. In order to provide better confidentiality and robust future over internet, [6] Narrates an approach that integrate TSSNB(To Small Sometime Naive) and ABS(Adaptive Blocking Schema) that detect the unwanted crawler and dynamically block the request in real time.

In order to display all the digital document and libraries that is in the form of Natural Language processing which are supported by linguistic resources, [5] invented a methodology that automatically learn natural language from the document of specific language and develop a resource that might be available for all other language. Further, it enables high level processing of document in that language and can be taken as a basis for future manual refinements. Web crawlers are basically categories in four parts [7].

- Focused Web Crawler
- Incremental Crawler
- Distributed Crawler
- Parallel Crawler

Focused web crawler downloads the relevant web pages. It collects the appropriate documents from the given query. Incremental crawler is dynamic crawler as it replaces the old documents with new documents. Parallel Crawler has parallel working nature to work in parallel manner. Distributed crawler works in distributed way to accomplish the task. [8] Discusses the approach that yields the best result than well-known breadth first crawler. The associated cost of the crawling can be effectively reduced by this method. The less crawler cost will give the more accurate results.

To bring the output pre-processing is required on the input data. Normalization is one of such method used for pre-processing. It is the process of bringing the word to its original form i.e. core form. This can be done by the stemming process. Normally normalization is done to find and remove the suffixes attached to the words in order to find the occurrences of the same word repeatedly in specific context (e.g. going and go gives the same meaning, computing and computed also have same meaning). Again the suffixes to be searched should be known in advance. Sometimes it may possible that normalization carried by the stemming process will change the meaning of the word (computing and computed will give compute), a solution on this is to use lemmatization. But a condition that doesn't have the linguistic knowledge in prior will support the stemming as a best method.

[10] Based on the context aware represents a stemming algorithm. The given approach is intended to reduce the morphological variation of the input query caused because of stemming process. The stated algorithm takes well known port stemmer algorithm as a base for development. The rule based approach is used to do so. Affix removal approach [11, 12, 13, 14, & 15] is one of the good approaches used for the purpose of stemming. This algorithm came under the classical approach of stemming technique. There are number of algorithms like Dawson stemmer, Lovins stemmer, Paice-Husk stemmer came under the same category. Lovins stemmer works on the principle of longest match. Dawson stemmer also makes use of principle of longest match and it replaces the recoding rule used in the Lovins stemmer to make it reliable. Paice-Husk stemmer is another algorithm for stemming word removal which finds out the answer in indefinite steps. Among all the above stated methods porter stemmer gain popularity because its performance over another algorithms.

[16] Illustrates the n- gram stemmer, an interesting and language independent method which makes use of string similarity approach. The basic idea behind the approach is that the similar word will have high proportions of n-grams in common. E.g. for n=2, n=3 words will be diagrams and trigrams respectively. N-gram technique is one of the common techniques used in the approaches stated in [13]. But one of the biggest disadvantage of the method is it requires the significant amount of memory. [17] Presents a new approach based on the Hidden Markov Model (HMMs) which are finite state automata where probability functions are used

for the rules between the transitions. Such method will not require the prior linguistic knowledge of the dataset.

Yass stemmer [18] stands for Yet another Suffix Striper and another stemming method based on the statistical and corpus method. Also it will not require the prior language knowledge and it is language independent. [14, 19, 20] tries to elaborate are some of the hybrid approaches of stemming word removal. It includes Linguistic Lexical Validation Stemmer, Corpus Based Stemmer, and Context Sensitive Stemmer. A study [13] gives a methodology of Linguistic Lexical Validation Stemmer. The main motto behind the method is to reduce the stemming errors and increase the accuracy of the overall system. Krovetz gives one dictionary that can validate the correctness of the suffix words. Hence dictionary lookup is done once stemmer is over in specific order. In this stemmer the spelling mistakes and meaningless stems are transforms into the nearest words.

Corpus Based Stemmer was being proposed by the [17] where author tries to overcome some of the disadvantages of the well-known stemmer Port Stemmer. The biggest problem with the port stemmer is that sometime it generates the stems which are not real words. This problem is easily overcome by the stated stemmer. Context Sensitive Stemmer fins an interesting method of stemming because morphological variants necessary for the search are predicted before the query is submitted to the search engines. This experiment dramatically reduces the unwanted expansions. Also precision can be increase too much by the method.

3. PROPOSED SYSTEM

In this section, we describe our framework to find aliases in dynamic web pages with the below mentioned stages as shown in figure 2.

Stage 1: In this step we are creating a web crawler which accepts a seed URL of the web site and searches it's all links.

Web crawlers are an essential component to search engines but running a web crawler is a challenging task. There are tricky performance and reliability issues as well as there are some social issues. Crawling is the most broken structure application as it involves interacting with hundreds of thousands of web servers and various name servers, which are all beyond the control of the system.

Web crawling speed is not only depends on the speed of one's own Internet connection, but also on the speed of the sites that are to be crawled. Especially if one is a crawling site from multiple servers, if many downloads are done in parallel, then total crawling time can be significantly reduced.

At the core, the numerous applications for Web crawlers, are all fundamentally the same. Web crawlers work by following process: Download the Web page. Parse the downloaded page and retrieve all the links.

Repeat the process, for each link retrieved. The Web crawler can be used for crawling the web pages through a whole site on the Inter/Intranet. When we specify a seed URL and the Crawler follows all links found in that HTML page. This usually leads to find more links, which will be followed again. A site can be seen as a tree-structure, the root is the seed URL; all links in that root-HTML-page are direct sons of the root. Succeeding links are then sons of the previous sons. The web crawler is based on the depth first algorithm as shown in below Algorithm.

Algorithm 1: DFS (G, v)

Input: graph G and a start vertex v of G

Output: labeling of the edges of G in the connected component of v as discovery edges and back edges

setLabel (v, VISITED)

for all e ∈ G.incidentEdges (v)

if getLabel (e) = UNEXPLORED

w ← opposite (v,e)

if getLabel (w) = UNEXPLORED

setLabel (e, DISCOVERY)

DFS (G, w)

else

set Label (e, BACK)

Here in our proposed method we developed a web crawler using java programming language, where we used multithreading feature extensively and also used java html parser to parse the web pages. And finally we store all collected web links in the live vector.

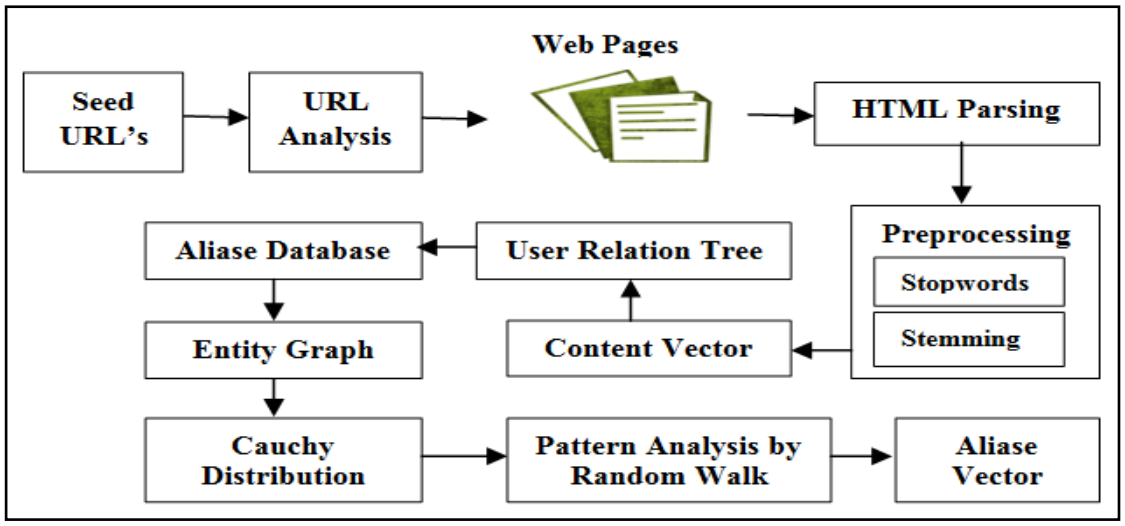

Figure 2: System Architecture

Stage 2: This is the one of the most crucial phase of our experiment, where our system interacts with the live web page. And then by using a well-designed baby web crawler our system is enable to fetch the data of the web page and then parse all the HTML tags from the web page. Only human readable data is extracted from the web page and also many advertisements contents are also vomited in this phase. And this well parsed data will be stored in vector and then it is passing for preprocessing as stated in the next session.

Stage 3: This is the step where we are preprocessing is conducted, where string is processed to its basic meaning words by the following four main activities: Sentence Segmentation, Removing Stop Word, Tokenization and Word Stemming.

- *Sentence segmentation* is boundary detection and separating source text into sentence.
- *Tokenization* means it separate the input query into individual words.
- *Stop word removal :*In any document narration the conjunction words does not play much role in the meaning of the document, so by discarding these words (like: is, the, for, an) from the documents which greatly reduces the overhead of processing
- *Stemming:* Many of the elongated words in the English language generally fail to provide proper meaning in the given scenario and also they increases the computational time. So it is necessary to bring the words to their base form by replacing its extended characters with desired characters (Example: studied is reference to study, where "ied" is replaced with "y").

Stage 4: As mentioned earlier entity relation is created base on the names. So now our aim is to find the noun from the extracted text form the web pages.

Noun extraction from the entered text plays a vital role in identification of the perception of the web pages according to the user. This is done by comparing each word of the query with the dictionary collected for almost 1, 00,000 words of English language. The details of this process can be shown in the below algorithm.

Algorithm 2: Noun Detection

Input: Query String

Output: Noun Words

Step 0: Start

Step 1: Read string

Step 2: divide string into words on space and store in a vector V

Step 3: Identify the duplicate words in the vector and remove them

Step 4: **for** i=0 to N (Where N is length of V)

Step 5: for ith word of N check for its occurrence in Dictionary

Step 6: if present then return true

Step 7: else return false

Step 8: stop

Stage 5: After Extracting noun an entity graph will be created with respect to the user and the web links in a two dimensions vector form.

Stage 6: Here in this system by using random walk algorithm and Cauchy distribution method which mentioned in equation 1. A fine grained Aliases vector is generated which shows all possible relational vector from one user to another.

The proposed system can be expressed using following mathematical model.

1. Let S={ } be as system for Entity Relationship Graph
2. Identify Input as I={ Ui }Where U=Seed URLsS= {I}
3. Identify A as Output i.e. Aliase VectorS= {I, A}
4. Identify Process P S= {I, A, P} P= { W_p, P_r, R_t ,E_g, R_w }

Where W_p = Web Information Parsing

P_r =Preprocessing

R_t =User Relation Tree

E_g = Entity Graph

R_w =Pattern analysis by Random Walk

5. $S = \{Q, R, W_p, P_r, T_i, T_s, N_d, F_l\}$

The union of all subset of S Gives the final proposed system.

The Cauchy distribution can be done by calculating the turning angle in random walk with the below mentioned equation

$$\emptyset = [2 \times \arctan\{\frac{(1-\rho)\times\tan\pi\times(r-0.5))}{1+\rho}\}]\text{----------(1)}$$

Where,

$\emptyset$ = Turning angle

ρ = Uniform distribution factor

r= Random variable

4. RESULTS AND DISCUSSIONS

The effectiveness of implemented system some experiments are conducted on java based windows machine. To measure the performance of the system we set the bench mark by selecting real world web pages as the input to the system.

The performance of the system is determined by examining that how many relevant Aliases entity vectors are formed based on the Random walk theorem.

To measure this, precision and recall are considered as the best measuring techniques. So precision can be defined as the ratio of the number of relevant Aliases identified to the total number of irrelevant and relevant Aliases identified. It is usually expressed as a percentage. This gives the information about the relative effectiveness of the system.

Whereas Recall is the ratio of the number of relevant Aliases are identified to the total number of relevant Aliases identified. It is usually expressed as a percentage. This gives the information about the absolute accuracy of the system.

The advantage of having two measures like precision and recall is that one is more important than the other in many circumstances.

For more clarity let we assign

• A = The number of relevant Aliases identified,

• B = The number of relevant Aliases not identified, and

• C = The number of irrelevant Aliases identified.

So, Precision = (A / (A+ C))*100

And Recall = (A / (A+ B))*100

No. Of Web Pages → / Parameter ↓	1	2	3	4	5	6	7	8	9	10
Precision (In %)	100	89	89	89	86	93	86	93	89	86
Recall (In %)	98	90	83	83	86	83	79	77	72	87

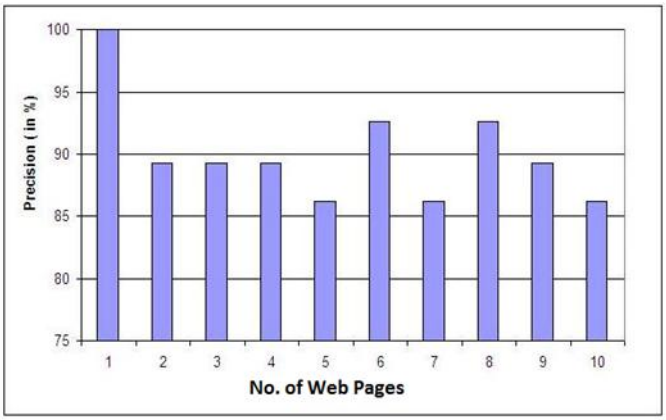

Figure 3. Average precision of the proposed approach

In Figure 3, we observe that the tendency of average precision for the identified Aliases are high compared to other systems.

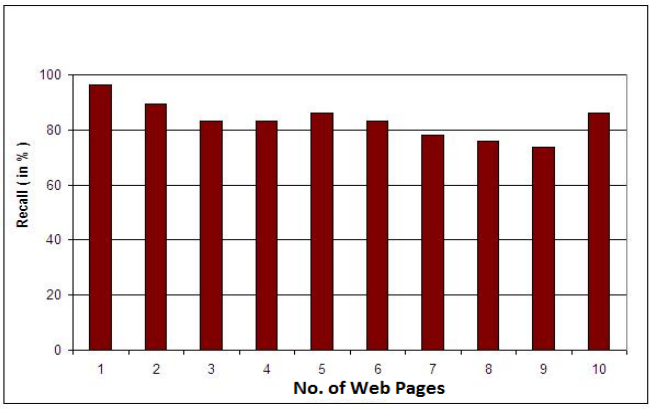

Figure 4. Average Recall of the proposed approach

In Figure 4, we observe that the tendency of average Recall for the identified Aliases are high compared to other system. So this shows that our proposed system is achieving high accuracy than any other method.

5. CONCLUSION

The proposed system successfully designs a recursive multi-threaded web crawler which actually takes a seed URL from the online web pages and crawl it systematically to collect its entire sub URLs. Then another baby crawler in the system crawls each and every collected web page to get the parsed information of the each web page. Parsing web page is bit challenging job which is catalyzed by the java HTML parser. Proposed system extracts the important features (like Noun) from the parsed web content and then this will be used to create entity graph, then this graph will be successfully used by the random walk algorithm to find most appropriate aliases vector from the live web pages.

The proposed system can be enhanced as an effective API that can be easily integrate with any system where there is a need to find aliases.

6. ACKNOWLEDGMENTS

I would like to extend my sincere thanks to Asst. Prof. S. R. Todmal for his guidance, encouragement and support throughout the course of this work. I would also like to thanks to all staff members for their unconditional support and guidance.

7. REFERENCES

[1] Pavalam S. M., S. V. Kashmir Raja , Felix K. Akorli and Jawahar M., "A Survey of Web Crawler Algorithms". IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 6, No 1, November 2011 ISSN (Online): 1694-0814.

[2] C. A. Pina-Garcia, Dongbing Gu, "Scraping Global Threats in Facebook Through Movement Patterns Generated by Random Walks". 4th Computer Science and Electronic Engineering Conference (CEEC), 2012.

[3] Snehal S. Shinde, P. R. Devale "Automated Entity Alias Evocation from Web". International Journal of Recent Technology and Engineering (IJRTE)ISSN: 2277-3878, Volume-1, Issue-5, November 2012.

[4] Kaushik Chakrabarti, Surajit Chaudhuri, Tao Cheng, Dong Xin "A Framework for Robust Discovery of Entity Synonyms". ACM 978-1-4503-1462-6 /12/08,2012.

[5] M.V.Prabath Kumar, "FOCUS: Learning to Crawl Internet Forums". International Journal of Emerging Engineering Research and Technology Volume 2, Issue 3, PP 239-245, June 2014.

[6] DeXiang Zhang, DiFan Zhang and Xun Liu, "A Novel Malicious Web Crawler Detector: Performance and Evaluation". IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013.

[7] Trupti V. Udapure, Ravindra D. Kale, Rajesh C. Dharmik "Study of Web Crawler and its Different Types". IOSR Journal of Computer Engineering (IOSR-JCE), ISSN: 2278-8727Volume 16, Issue 1, Ver. VI ,PP 01-05, Feb. 2014.

[8] Ricardo Baeza-Yates, Ricardo Baeza-Yates "Crawling a Country: Better Strategies than BreadthFirst for Web Page Ordering". 14th international conference on WWW, Pages 864-872, ACM,2005.

[9] Michal Konkol and Miloslav Konopík , "Named Entity Recognition for Highly Inflectional Languages: Effects of Various Lemmatization and Stemming Approaches". LNAI 8655, pp. 267–274, 2014.

[10] K.K. Agbele, A.O. Adesina, N.A. Azeez , A.P. Abidoye "Context-Aware Stemming Algorithm for Semantically Related Root Words".Afr J Comp & ICT, ISSN 2006-1781,2012.

[11] J. B. Lovins, "Development of a Stemming Algorithm". Mechanical Translation and Computational Linguistics, vol.11, no. 12, pp: 22-31,1968.

[12] J. Dawson, "Suffix removal and word conflation".ALLC Bulletin, vol. 2, no. 3, pp: 33-46,1974.

[13] M. Porter, "An Algorithm for Suffix Stripping. Program". vol. 14, no. 3, pp: 130 – 137, 1980.

[14] Wahiba Ben Abdessalem Karaa, "A New Stemmer To Improve Information Retrieval". International Journal of Network Security & Its Applications (IJNSA), Vol.5, No.4, July 2013.

[15] D. Paice Chris. (1990). Another Stemmer. ACM SIGIR Forum, Volume 24, No. 3, pp: 56-61.

[16] R. Krovetz.. "Viewing morphology as an inference process". In Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Pittsburgh, PA, USA – June 27 th –July 01, 1993, pp: 191-202.

[17] M. Melucci and N. Orio. (2003), "A novel method for stemmer generation based on hidden Markov models". Proceedings of the 12th international conference on Information and knowledge management, New Orleans, LA, USA, ACM Nov 2003 – 08, pp:131-138.

[18] M. Prasenjit, M. Mandar, K. Swapan K. Parui, K. Gobinda, M. Pabitra and D. Kalyankumar,"YASS: Yet another suffix stripper".ACM Transactions on Information Systems. vol. 25, no. 4, article 18, 2007

[19] J. Xu, W.B. Croft, "Corpus-based stemming using co-occurrence of word variants", ACM Transactions on Information Systems,1998, vol. 16, no. 1, pp: 61-81.

[20] P. Funchun, A. Nawaaz, L. Xin and L. Yumao, "Context sensitive stemming for web search". Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval Amsterdam, July 23 – 27, 2007, pp: 639-646.

[21] Lili Jiang, Ping Luo, Jianyong Wang, Yuhong Xiong, Bingduan Lin, Min Wang, Ning An,"GRIAS: an Entity-Relation Graph based Framework for Discovering Entity Aliases".2013 IEEE 13th International Conference on Data Mining.