Semantic Search Engine Using Natural Language Processing

Article *in* Lecture Notes in Electrical Engineering · November 2015

DOI: 10.1007/978-3-319-07674-4_53

3 authors, including:

Sudhakar Pandiarajan
Indian Institute of Technology Kharagpur
9 PUBLICATIONS **268** CITATIONS

SEE PROFILE

Chapter 53
Semantic Search Engine Using Natural Language Processing

Sudhakar Pandiarajan, V.M. Yazhmozhi and P. Praveen kumar

Abstract The World Wide Web has become colossal and its growth is also dynamic. Most of the people rely on the search engines to retrieve and share information from various resources. All the results returned by search engines are not always relevant as it is retrieved from heterogeneous data sources. Moreover a naive user finds it difficult to confirm that the retrieved results are significant to the user query. Therefore semantic web plays a major role in interpreting the relevancy of search results. In this work, a novel algorithm is proposed for retrieving relevant documents using semantic web based on the concept of Natural Language processing (NLP). In this proposed system, NLP is used to analyse the user query in terms of Parts of Speech. The extracted terms are compared to the domain dictionary to identify the relevant domain of the user interest. On the other hand, the retrieved documents of the user query are investigated with the help Natural language processing to identify the relevant domain. Now the documents are ranked as per the relevancy of the contents against user query. The experimental result of the proposed algorithm indicates that the accuracy of the retrieved document is 97 %.

Keywords Semantic search engine · Natural language processing

S. Pandiarajan (✉) · V.M. Yazhmozhi · P. Praveen kumar
Kamaraj College of Engineering and Technology, Virudhunagar 626101, India
e-mail: sudhakarcse@kcetvnr.org

V.M. Yazhmozhi
e-mail: yazh.technovate@gmail.com

P. Praveen kumar
e-mail: praveenkumargen@kcetvnr.org

H.A. Sulaiman et al. (eds.), *Advanced Computer and Communication Engineering Technology*, Lecture Notes in Electrical Engineering 315,
DOI 10.1007/978-3-319-07674-4_53

53.1 Introduction

With the increasing use of World Wide Web as an essential source of information, there is a need to work with the Semantic Web [1], in order to reduce the irrelevant data obtained during the search. The existing search engines [2], [3] do not provide domain specific search and they simply perform keyword matching. Those search engines cannot understand the negative senses, Example: "I do not want clustering". The result set of the existing search engines for such a query would be related to clustering as they merely perform keyword matching and don't analyse the actual meaning of the user query. Other example of queries with negative senses includes "I want clustering algorithms except cobweb". The pro-posed approach using semantic search algorithm overcomes all the above stated pitfalls.

Semantic Web provides the users a comfort zone and reduces the wastage of time. The proposed work on Semantic search is accomplished by POS (Parts Of Speech) tagging using Natural Language Processing. Using POS Tagging, the proposed semantic search algorithm can understand what the user query conveys and hence it provides more relevant results to the user. The query entered by the user is POS tagged using Stanford Parser, and the tags for each word in the user query are obtained. For each tag obtained, if it is a noun it is added to the noun list (NL), and all other tags correspondingly. Each word in the NL is now compared with the word dictionary of each document that is obtained during pre-processing the document. If a match is found, then the word weight is incremented, and is added to the word weight list (WWL). The documents are sorted based on the WWL and added to the result set. Similarly, each sentence in the document is detected using OpenNLP, and its weight is incremented if all the nouns in the NL are matched. This weight of each sentence is added to the sentence weight list (SWL). The documents are then sorted based on the SWL and added to the result set. If there is any occurrence of negative word (e.g., not, except, NEITHER-NOR), then all the nouns in the NL are skipped and compared with the word dictionary of each document if verbs of possession occurs before those negative words. These documents are also added to the result set. The ranked documents from the result set are retrieved to the user. By this approach of Semantic search [4], [5], [6], the search results that are more relevant to the user's interest are provided. In the proposed work, pdf, word and html documents have been considered for analysis.

53.1.1 Outline of the Paper

Section 53.2 presents the various works on semantic web and natural language processing supportive to the proposed research. Section 53.3 describes the Architectural design of the proposed scheme. Section 53.4 illustrates the experimental results and Performance Evaluation. Section 53.5 depicts the conclusion and future work.

53.2 Related Works

In [7, 8], Mukhopadhyay et.al has ogy which is made effective by mapping the instances and the classes. Even thought the results of this research prove that the performance are good than the regular search engine, the results are proved within the domain only. In [9], Cafarella et al. developed a search engine using natural language processing in which it out performs well in terms of producing relevant information using natural language processing. In [10], Karpagam had proposed a framework which is based on ontology to build the semantic search engine. The author finds the relevant document for the user query using the techniques like word stemming, ontology matching, weight assignment, rank calculation. If the approach is extended to a larger data set, the weight assignment and rank calculation will become tedious. In [11], Jiang et al. developed a semantic search engine that overcomes the problem of knowledge overhead by a query interface. In [12], Lei et al. proposed a semantic web portal that has been designed to ensure the quality of the extracted metadata and it also facilitates for data querying. In [13], Kruse et al. had used WordNet, a lexical database to find the word senses in order to achieve semantic. But WordNet does not provide a classification of word senses in technical terms. In [14], Lara et al. had proposed a hybrid searching technology which is the combination of ontology and traditional keyword based matching in which the drawbacks of keyword based search like stop words removal in the user query was reflected. In [15], Madhu had presented a survey on the search engine's generation and role of the search engines in the intelligent web in which it insists the necessity to build semantic search engines. In [16], Kalaivani had proposed a question answering system based on the semantic searching terminology and natural language processing technique. In [11], Jiang et al. developed a full text search engine has been designed to exploit ontological knowledge for document retrieval. In [17], Kerschberg et al. recommended a methodology has been developed to capture the semantics of the users search intent into target search queries of the existing search engine.

53.3 Architectural Design

In the proposed work, Structured and Unstructured documents are maintained separately. Whenever a user query arises, it is given to Stanford Parser for POS tagging. Based on POS Tagging [18] NN (Noun, Singular), NNS (Noun, Plural), NNP (Proper Noun, Singular), JJ (Adjective) and IN (Preposition) are extracted from user query and stored in a term list table. Decisions are taken based on the accompanying diagram. If the user query belongs to only one domain then all

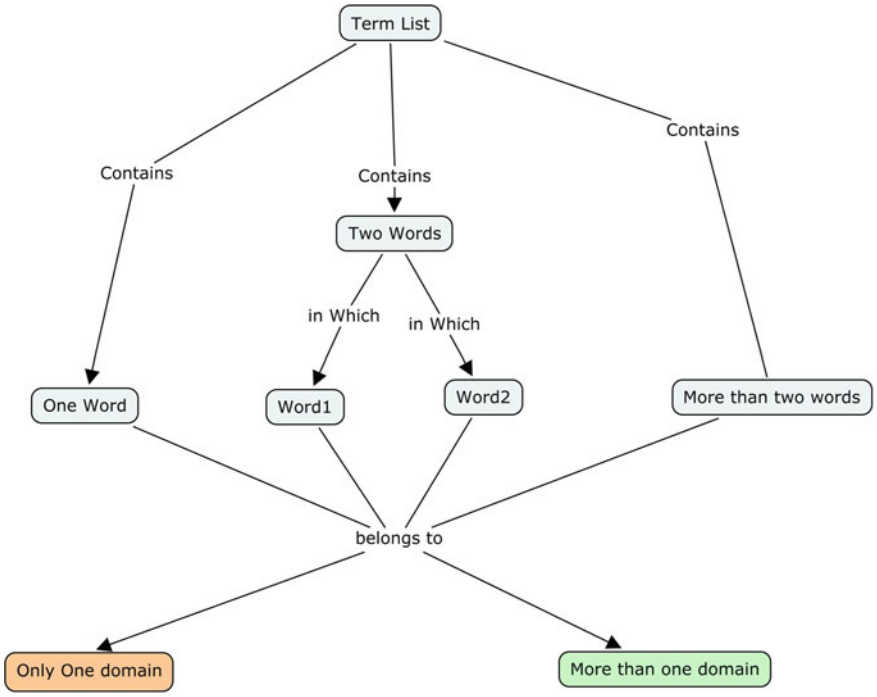

Fig. 53.1 Domain classification based on term list

the documents in the repository are compared against the domain dictionary and the results are computed. On the other hand, if the user query belongs to more than one domain then the dominant domain is computed based on Fig. 53.1. There are 2 possible cases can occur in selecting dominant domain

Case 1: If Most of the Words belong to one domain then the same domain is taken as dominant domain.
Case 2: If Equal number of Matches found with two domains then the choice are given to the user to confirm the dominant domain.

Once the dominant domain is selected, each document is split into sentences and each sentence is further divided into words. Each word is compared with term list and domain dictionary for matching. If a match is found then the "sentence weight" is incremented. The same process is continued for the whole document and cumulative document weight is calculated. If the user query contains negative words, then a negative ag is set to the user query and the process will request the

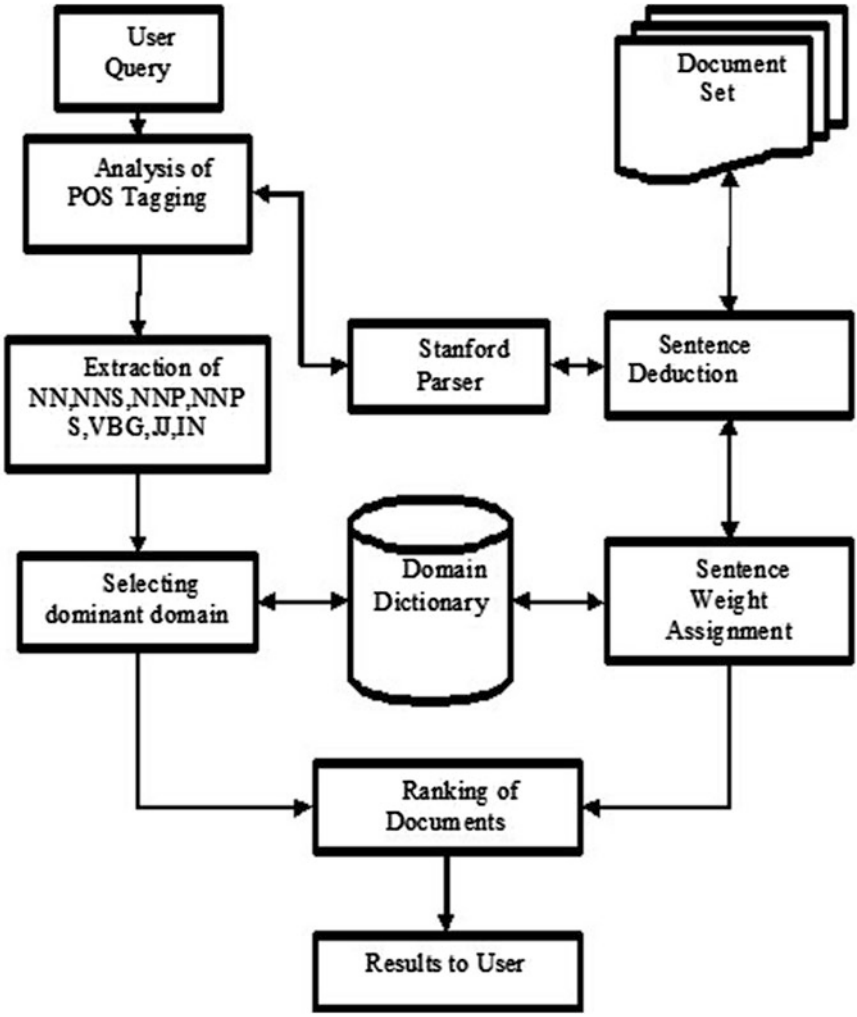

Fig. 53.2 Architectural design of the proposed approach

user to re-enter the direct query for better retrieval of results. In this way the accuracy of the retrieval is brought out. Once all the documents weight is calculated the highest weight of the document is ranked first and given to user as a good match in the retrieved results (Fig. 53.2).

Algorithm 1 - POS Tagging using NLP
INPUT : User query
OUTPUT : Resultant Document Set
METHOD : POS Tagging IN NLP

Step 1 : Initialize the Resultant document set RES = { }
Step 2 : Initialize nounlist NL= { }
Step 3 : Initialize sentencelist SL={ }
Step 4 : Initialize negativelist NGL= { }
Step 5 : Initialize NEGLIST= { not, no,neither, nor, except... }
Step 6 : Initialize neg ag=0, word weight=0, sentence weight=0;
Step 7 : POS Tag the user query using stanford parser.
Step 8 : foreach tag obtained
Step 8a : If tag is NN, NNS, NNP, NNPS, JJ, VBG then add the appropriate word in the query to NL
Step 8b: If tag is RB,DT,CC and the word is in NEGLIST then set neg ag=1
Step 9 : If NL.count=1 and PL.count=0 and neg ag=0 then nd the concept under which the single noun in NL occurs in Domain Dictionary.
Step 10 : If the single noun occurs under more than one concept, display all concepts.
Step 11 : As per the user selection, add that selected concept too in NL.
Step 12 : Detect sentences in the web documents using OpenNLP and store in SL
Step 13 : foreach sentence sent in SL
Step 14 : If all nouns in NL occur within a sentence
Step 15 : Increment weight for each match.
Step 16 : Store sentence weight of each document into SentenceWeightList SWL Step 17 : Sort the documents according to sentence weight and store it in the RES.
Step 18 : Compare each entry in the NL with word dictionary of each document
Step 18a: If a match is found then increment word weight with the count attribute's value.
Step 19 : Store word weight of each document into WordWeightList WWL.
Step 20 : Sort the documents according to word weight and if that document is not in RES store it in RES
Step 21 : If neg ag=1 then skip all nouns in NL and compare with the word dictionary of each document
Step 21a : Store such documents into RES
Step 22 : Retrieve the ranked documents fD1,.Dng in RES to the user

53.4 Experimental Results and Discussion

The experiment was conducted using American national corpus (anc) [19, 20] by in-creasing the number of documents in the repository gradually. The main factor considered for evaluation is accuracy rather than the speed. Three different scenarios are considered in evaluating the performance of the proposed system.

A confusion matrix is created with various combinations of Relevant documents (RD) and Conflicting documents (CD) to analyse the system. Performance evaluation of the proposed approach is done based on the classification context scenario. Precision, Recall, Accuracy and F-measure are the major measures used for classification based performance. Precision is the probability that measure a retrieved document is relevant to the context. Precision is calculated based on the formula

$$\text{Precision} = \frac{TP}{TP + FP}$$

where

TP is true positive (Correctly retrieved)
TN is true negative (Correctly rejected)
FP is false positive (Incorrectly retrieved)
FN is false negative (Incorrectly rejected).

Recall is the probability that calculates a relevant document is retrieved in a search process. Recall is calculated based on the formula

$$\text{Recall} = \frac{TP}{TP + FN}$$

Accuracy is calculated based on the formula

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

First scenario creates a large number of Relevant Documents against the small number of Conflicting documents. The Second scenario creates an equal number of Relevant and Conflicting documents. The Third scenario creates a small number of Relevant Documents against with large number of Conflicting documents. All the three cases, the document set contents are increased gradually (Tables 53.1, 53.2, 53.3).

The accuracy and recall of the system with respect to the above stated 3 cases is plotted in Figs. 53.3 and 53.4 respectively.

All the performance measures of the system clearly highlights that the results produced by the system in terms of all aspects are better compared with existing systems.

Table 53.1 Relevant documents are higher than the conflicting documents

No. of documents	Accuracy	F-measure	Recall	Precision
100	0.978	0.965	0.974	0.962
200	0.975	0.963	0.972	0.96
300	0.974	0.964	0.973	0.961
400	0.974	0.962	0.971	0.957
500	0.97	0.963	0.973	0.961
600	0.967	0.962	0.974	0.964
700	0.966	0.96	0.972	0.961
800	0.968	0.962	0.973	0.962
900	0.97	0.96	0.972	0.961
1000	0.968	0.952	0.965	0.95
1500	0.964	0.958	0.962	0.951
2000	0.962	0.956	0.96	0.95
3000	0.963	0.956	0.961	0.951
5000	0.962	0.955	0.959	0.95

Table 53.2 Relevant documents are equal to the conflicting documents

No. of documents	Accuracy	F-measure	Recall	Precision
100	0.975	0.963	0.972	0.96
200	0.974	0.961	0.97	0.958
300	0.972	0.963	0.967	0.951
400	0.974	0.96	0.962	0.955
500	0.972	0.962	0.963	0.952
600	0.97	0.963	0.965	0.952
700	0.965	0.961	0.962	0.949
800	0.968	0.962	0.964	0.95
900	0.964	0.96	0.96	0.948
1000	0.964	0.952	0.962	0.95
1500	0.962	0.95	0.96	0.949
2000	0.96	0.955	0.958	0.948
3000	0.96	0.952	0.955	0.95
5000	0.961	0.951	0.954	0.95

Table 53.3 Relevant documents are smaller than the conflicting documents

No. of documents	Accuracy	F-measure	Recall	Precision
100	0.972	0.958	0.965	0.954
200	0.97	0.957	0.967	0.953
300	0.968	0.953	0.962	0.95
400	0.973	0.952	0.961	0.952
500	0.97	0.954	0.964	0.952
600	0.965	0.951	0.96	0.95
700	0.964	0.948	0.961	0.951
800	0.965	0.95	0.962	0.953
900	0.965	0.948	0.962	0.951
1000	0.96	0.952	0.958	0.95
1500	0.961	0.947	0.956	0.945
2000	0.958	0.952	0.957	0.946
3000	0.959	0.955	0.958	0.95
5000	0.96	0.954	0.958	0.951

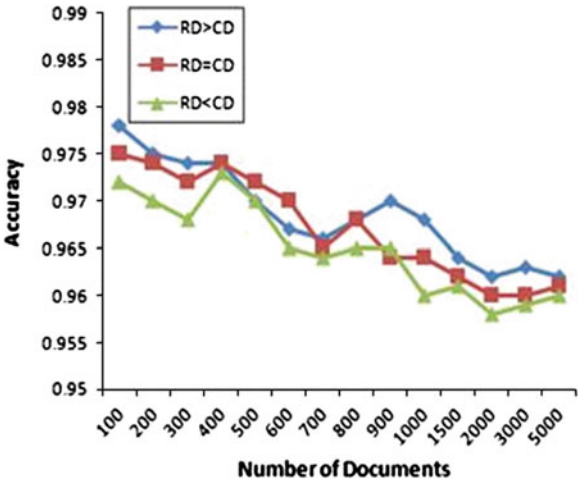

Fig. 53.3 Shows that the accuracy of the proposed systems is maintained between 96 and 98% irrespective of the number of documents

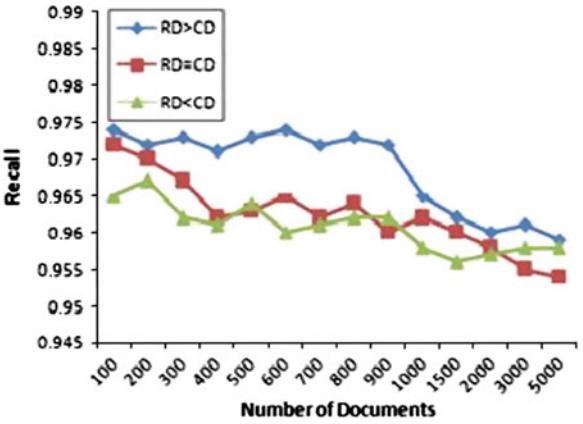

Fig. 53.4 Presents the recall measure of the proposed system against various numbers of document sets. Irrespective of the documents combinations the proposed system outperforms well in between 95.5 and 97.5%

53.5 Conclusion

In this study a new approach is proposed based on Natural language processing to understand and classify the documents based on the user query. The experimental results of the proposed systems point out that the accuracy of this system is vary in between 95 and 97 % in terms of relevancy. However the time taken to classify a document is little high compared with existing search engines. In future, our system will address the existing drawbacks to compete with other search engines in terms of the time factor.

References

1. http://swoogle.umbc.edu/
2. http://hakia.com/
3. https://duckduckgo.com/
4. Yu, L.: A Developers Guide to the Semantic Web. Springer, Berlin (2011). ISBN: 9783642159695
5. Antoniou, G., Groth, P., van Harmelen, F., Hoekstra, R.: A Semantic Web Primer, 3rd edn. ISBN: 0262018284
6. Segaran, T., Taylor, J., Evans, C., O'Reilly (2009) Programming the Semantic Web (2009). ISBN: 9780596802066
7. Mukhopadhyay, D., Banik, A., Mukherjee, S., Bhat-tacharya, J., Kim, Y.-C.: A Domain Specific Ontology Based Semantic Web Search Engine. In: Proceedings of the 7th International Workshop (MSPT), p. 8189, Feb 5 2007. ISSN 1975-5635, 89-8801-90-0
8. FinKelstein, L., Gabrilovich, E., Matias, Y.: Placing search in context: the concept revisited. In: Proceedings of the 10th International Conference on World Wide Web, pp. 406–414 (2001)
9. Cafarella, M.J., Etzioni, O.: A search engine for natural language processing. In: Proceedings of the International World Wide Web Conference Committee (IW3C2), ACM 1595930469/05/0005 (2005)

10. Karpagam, G.R., Uma Maheswari, J.: A conceptual framework for ontology based information retrieval. Int. J. Eng. Sci. Technol. **2**(10), 5679–5688 (2010)
11. Jiang X., Tan, A.-H.: OntoSearch: a full-text search engine for the semantic web
12. Lei, Y., Uren, V., Motta, E.: SemSearch: a search engine for the semantic web. In: Proceedings of the International Conference on Knowledge Engineering and Knowledge Management Managing Knowledge in a World of Networks, pp. 238–245. Springer, Berlin (2006)
13. Kruse, P.M., Naujoks, A., Rsner, D., Kunze, M.: Clever search: a WordNet based wrapper for internet search engines, computing research repository CORR, vol. Abs/cs/050 (2005)
14. Lara, R., Han, S.-K., Lausen, H., Stollberg, M., Ding, Y., Fensel, D.: An evaluation of semantic web portals. In: Proceedings of the IADIS Applied Computing International Conference 2004, Lisabon, Mar 23–26 2004
15. Madhu, G., Govardhan, A., Rajinikanth, T.V.: Intelligent semantic web search engines: a brief survey. Int. J. Web Semant. Technol. **2**(1), 34–42 (2011)
16. Kalaivani, S., Duraiswamy, K.: Personalized semantic search based intelligent question answering system using semantic web and domain ontology. In: Proceedings of the International Conference on Advanced Computer Technology (IJCA), pp. 15–17 (2011)
17. Kerschberg, L., Kim, W., Scime, A.: Intelligent web search via personalizable Meta—search agents
18. Stanford Parser: http://nlp.stanford.edu/software/index.shtml
19. http://www.anc.org/data/masc/downloads/data-download/
20. Lei, Y., Lopez, V., Motta, E.: An infrastructure for building se-mantic web portals. In: Proceedings of the International Workshop on Web Information Systems Modeling (WISM), pp. 283–308 (2006)