www.ijecs.in
International Journal Of Engineering And Computer Science ISSN: 2319-7242
Volume 5 Issue 11 Nov. 2016, Page No. 18827-18831

Improving the Reverse Dictionary Using Scalable Dataset

Mrs. R. Carolene Praveena, Mrs. A. Manjula,
Asst. Professor, Department of Computer Science,
Research Scholar, Department of Computer Science,
Sri Jayendra Saraswathy Maha Vidyalaya College of Arts & Science. \.
Coimbatore.
Sri Jayendra Saraswathy Maha Vidyalaya College of Arts & Science.
Coimbatore

ABSTRACT

A reverse dictionary takes a user input phrase describing the desired concept, and returns a set of candidate words that satisfy the input phrase. The user entered phrase need not necessarily be the same as in the definition, therefore it is implemented in such a way that the concept of the user input will be considered and corresponding words will be obtained as the outcome. The experimental results show that the proposed approach provides significantly higher quality than existing reverse dictionaries.

Keywords – dictionary, reverse, RMS.

1. INTRODUCTION

The reverse dictionary describes the concept that one can enter a single word, phrase, or a few words to get the suitable equivalent word of set of words. In this paper, similarity is applied, using similarity for computing the relevancy between concepts verses and the documents or verses in English collections.

The contributions can be summarized as follows: (i) the resulting details can then be used to select the top ranked words. Based on this main verse, it can calculate the concepts similarity for every phrase. (ii) It can create the Reverse Mapping Set (RMS).

The RMS sets contains synonyms set, antonyms set, hypernyms set and hyponym sets and also includes stop words sets. (iii) The stop word set contains a, be, some, too, very, who, the, in, of, and, that, with, this… etc. (iv) A reverse dictionary is a dictionary organized in a non-standard order that provides the user with information that would be difficult to obtain from a traditionally alphabetized dictionary.

The outline representative known methods for the problem of write optimized data storage. The Reverse Dictionary Application (RDA) is a software module that takes a user phrase (U) as input, and returns a set of conceptually related words as output. The scalability of this approach depends on the RMS queries on the RMS databases and the queries related to Word Net hierarchy from hyponym-hypernym database. Finally it can measure the run time response for retrieving the results for every phrases and sentences.

In this paper, the reverse dictionary is used to predict similarity between the concept and phrases. The resulting details can then be used to select the top ranked words. Based on this main verse, it can calculate the concepts similarity for every phrase. It can create the Reverse Mapping Set (RMS). The RMS sets contains synonyms set, antonyms set, hypernyms set and hyponym sets and also includes stop words sets. The stop word set contains a, be, some, too, very, who, the, in, of, and, that, with, this… etc.

2. RELATED WORK

Some of the related works are discussed below In [2], the authors discussed about Latent dirichlet allocation (LDA) a generative probabilistic model for collections of discrete data such as text corpora.

The authors of [5][6] discussed Probabilistic Latent Semantic Indexing which describes five character-level passage Evaluation. The authors have demonstrated some of the problems that can arrive in passage retrieval

evaluation and presented five retrieval Methods that can correct some of these problems.

In [11] the authors discussed about the multi-document summarization using mutual reinforcement and relevance propagation models to group the document set into several topic themes and then these are clustered according to the query.

The authors J. Kim and K. Candan in [9] discussed the similarity between Words that was concerned with the syntactic similarity of two strings.

The authors K. Naresh Kumar & Dr. Rajender Nath in [19] discussed about the automatic text summarization that is used to summarize the source text into its shorter version by preserving its information content and overall meaning.

3. SYSTEM ARCHITECTURE

The Reverse dictionary is a computer application which takes the user input phrase and gives the corresponding words as output. The RMS contains a set of mappings, the dictionary definitions and parse trees for definitions. The database of synonyms which consists of the set of synonym for individual words in the user input phrase. The hypernym/hyponym database, which consists the hyponym and hypernym sets for each individual word in the user input phrase, whereas the Antonym database consists of the set of antonym for each word in the user input phrase. The actual definition for a word is in the forward dictionary. For the scalability of the system, I consider three characteristics. First, A frequently accessed data is stored, which lets the thread to get the required data without enquiring a database. Second, the implementing of a thread pool allows for simultaneous retrieval of synonym, hyponym and hypernym, and RMS sets for words of the user input phrase. Third, a separate database will increase the parallel processing and increase system capability .The algorithm to create a query executes the process upon receiving the user input phrase, then it calls the algorithm responsible for the sorting of the results, which requires accessing full definitions for each word. The Result sorting algorithm which allows the program to sort the output words according to the relevance of the user input word. To avoid too many words, the words are grouped together with other words, and the set of two words are found in the definition of a word. First, the words are arranged in a decreasing order, therefore the words which occur in many definitions and yields many words, that particular word will be deleted. In case, there are still more results than the threshold value, the words are again reduced until it is less than or equal to the threshold value. Finally, the words with more precise meaning to the user input phrase will be displayed, followed by several other words which are more likely to be associated with the search concept.

Finally filter the result that occurs as output according to Noun, verb, adjective, adverb form to get the appropriate sorted result. This is shown in Fig 1.

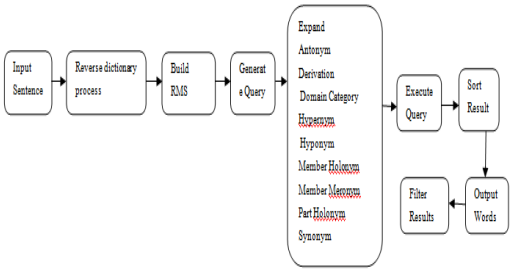

Fig 1: Reverse Dictionary System Architecture

4. IMPLEMENTATION AND RESULT ANALYSIS

The Reverse Dictionary Application (RDA) is a software module that takes a user phrase (U) as input, and returns a set of conceptually related words as output. To perform the processing, the RDA needs access to information stored in a set of databases:

- The RMS DB, which contains a table of mappings as well as dictionary definitions and computed parse trees for definitions.
- The Synonym DB, which contains the synonym set for each term ^t;
- The Hyponym/Hypernym DB, which contains the hyponym and hypernym relationship sets for each term ^t;
- The Antonym DB, which contains the antonym set for each term ^t;
- The actual dictionary definitions for each word in the dictionary.

DOI: 10.18535/ijecs/v5i11.19

- The mappings for the RMS, synonyms, hyponyms, hypernyms, and antonyms are stored as integer mappings, where each word in the WordNet dictionary is represented by a unique integer. This both condenses the size of the mapping sets, and allows for very fast processing of similarity comparisons, as compared to string processing. This architecture has three characteristics designed to ensure maximum scalability of the system.

$$\rho(a,b) = \frac{2 \times E(A(a,b))}{(E(a) + E(b))}.$$

First, a cache stores frequently accessed data, which allows a thread to access needed data without contacting a database. It is well known that some terms occur more frequently than others. The synonym, hyponym, hypernym, and RMS sets of these popular terms will be stored in the cache and the query execution in the database will be avoided.

Second, the implementation of a thread pool allows for parallel retrieval of synonym, hyponym, hypernym, and RMS sets for terms.

Synonym set: A set of conceptually related terms for t. Wsyn (t) = {t1; t2; . . . tj; . . . ; tn}, where tj is a synonym of t, as defined in the dictionary. For example, Wsyn (talk) might consist of the set of words {speak, utter, mouth, verbalize}.

Hypernym set: A set of conceptually more general terms describing t. Whyr (t) = {t1; t2; . . . tj; . . . ; tn}, where tj is a hypernym of t, as defined in the dictionary. For example, Whyr (red) might consist of {"color"}.

Hyponym Set: A set of conceptually more specific terms describing t. Whyo (t) = ft1; t2; . . . tj; . . . ; tn}, where tj is a hyponym of t, as defined in the dictionary. For example, Whyo(red) might consist of {"maroon," "crimson"}.

Negation word set: Negation is also a concern. For example, a user might enter an input phrase using the word "not," e.g., when an antonym of the negated term would be more precise.

Third, separate databases increase the opportunity for parallel processing, and increase system scalability. If a single machine is not capable of handling the necessary loads, the database can easily be further distributed across multiple servers using partitioning methods to improve overall system scalability.

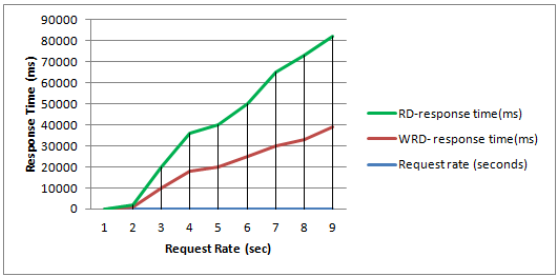

Fig 2: Architecture Diagram

This architecture has three characteristics designed to ensure maximum scalability of the system which is shown in Fig 2, First, a cache stores frequently accessed data, which allows a thread to access needed data without contacting a database. Second, the implementation of a thread pool allows for parallel retrieval of synonym, hyponym, hypernym, and RMS sets for terms. Third, separate databases increase the opportunity for parallel processing, and increase system scalability.

Results demonstrate the scalability and responsiveness of the approach with an existing approach, and then demonstrate the quality of solution compared to an existing approach.

Evaluation consists of four parts, First demonstrate the responsiveness of the WRD system in comparison to the SVM approach. Then demonstrated how the WRD system scales with the availability of more hardware resources in comparison to the SVM, LSI, and LDA. In the demonstration it is shown that the WRD performance and scale is substantially better than any of the existing approaches. The WRD is compared with some of the existing RD systems available and found that the accuracy of the WRD is significantly better than that of existing RD systems.

The curves all show classic exponential growth with increasing load. However, the response time of the system is about an order of magnitude better than the response time for the SVM approach, which requires vector similarity computation of the user defined phrase across the context vectors for all clusters. In contrast, the approach identifies a limited number of candidate words using the RMS and word-relationship data sets, and computes full similarity only against the definitions of these words in the final step. This requires significantly less processing than the SVM approach. This is shown in Fig3.

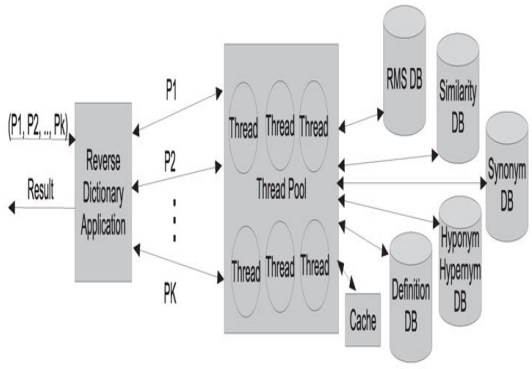

Fig 3: Response time performance as request rate increases.

5. CONCLUSION

In this work, Reverse Dictionary method provides improvements in performance scale quality. RMS runs efficiently compared to other existing methods. Reverse Dictionary with RMS takes a big step for finding a large class of emerging words. The results of this research work demonstrate that the proposed scheme can achieve high performance and quality.

REFERENCES

[1] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. ACM Press, 2011.

[2]D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," J. Machine Learning Research, vol. 3, pp. 993-1022, Mar. 2003.

[3] J. Carlberger, H. Dalianis, M. Hassel, and O. Knutsson, "Improving Precision in Information Retrieval for Swedish Using Stemming," Technical Report IPLab-194, TRITA-NA-P0116, Interaction and Presentation Laboratory, Royal Inst. of Technology and Stockholm Univ., Aug. 2001.

[4] V. Hatzivassiloglou, J. Klavans, and E. Eskin, "Detecting Text Similarity over Short Passages: Exploring Linguistic Feature Combinations via Machine Learning," Proc. Joint SIGDAT Conf. Empirical Methods in Natural Language Processing and Very Large Corpora, pp. 203-212, June 1999.

[5] T. Hofmann, "Probabilistic Latent Semantic Indexing," Proc. Int'l Conf. Research and Development in Information Retrieval (SIGIR), pp. 50-57, 1999.

[6] T. Hofmann, "Probabilistic Latent Semantic Indexing," SIGIR '99: Proc. 22nd Ann. Int'l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 50-57, 1999.

[7]T.Joachims,"Svmlight,"http://svmlight.joachims.org/, 2008.

[8]T.Joachims,"Svm^{multiclass},"http://svmlight.joachims.org/ svm_multiclass.html, 2008.

[9] J. Kim and K. Candan, "Cp/cv: Concept Similarity Mining without Frequency Information from Domain Describing Taxonomies," Proc. ACM Conf. Information and Knowledge Management, 2006.

[10] T. Korneius, J. Laurikkala, and M. Juhola, "On Principal Compnent Analysis, Cosine and Euclidean Measures in In-formation Retrieval," Information Sciences, vol. 177.

[11] Poonam P.Bariet. ," Multi-Document Text Summarization using Mutual Reinforcement and Relevance Propagation Models Added with Query and Features Profile", in International Journal of Advanced Computer Research, ISSN (print): 2249-7277, ISSN (online): 2277-7970, Volume-3, Number-3, Issue-11, pp. 59-63, September-2013.

[12] Ohm Sornil et. al.," An Automatic Text Summ-arization Approach using Content-Based and Graph-Based

Characteristics", in Cybernetics and Intelligent Systems, Print ISBN: 1-4244-0023-6, pp. 1-6, 2006.

[13] Wooncheol Jung et. al.," Automatic Text Summarization Using Two-Step Sentence Extraction ", in Information Retrieval Technology Lecture Notes in Computer Science , springer, pp. 71-81, Online ISBN 978-3-540-31871-2, Volume 3411, pp 71-81, 2005.

[14] S. Suneetha, "Automatic Text Summarization: The Current State of the art,"in *International Journal of Science and Advanced Technology*, ISSN 2221-8386, vol. 1, no. 9, pp. 283-293, 2011.

[15] Zhimin Chen et. al.," Research on Query-based Automatic Summarization of Webpage", in IEEE ISECS International Colloquium on Computing, Communication, Control, and Management, ISSN: 978-1-4244-4246-1/09, pp. 173-176,2009.

[16] Eduard Hovy et. al.," Automated Text Summarization in SUMMARIST", in proceeding of TIPSTER '98 Proceedings of a workshop on held at Baltimore, Maryland, pp. 197-214, doi> 10.3115/ 1119089.1119121, 1999.

[17] CemAksoy et. al.," Semantic Argument Frequency-Based Multi-Document Summarization", in The 24th International Symposium on Computer and Inform-ation Sciences, ISCIS, pp. 460-464, North Cyprus. IEEE 2009.

[18] Arman Kiani –B et. al.," Automatic Text Summarization Using: Hybrid Fuzzy GA-GP", in IEEE International Conference on Fuzzy Systems Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada, ISSN: 0-7803-9489-5/06, pp. 977-983, 2006.

[19] Naresh Kumar et. al., "Summarization of Search Results Based On Concept Segmentation", in international conference on data acquisition tran-sfer, processing and management (ICDATPM-2014), ISBN:978-81-924212-6-1, pp. 100-105, 2014.

[20] XiaoyanCai and Wenjie Li "Mu Manifold-Ranking Based Relevance propagation Model for Query-Focused Multi-Document Summarization" IEEE Transa speech, and language processing, vol. 20, no. 5, july 2012.

[21] X. J.Wang, J. W. Yang, and J. G. Xiao, "Manifold-rankingbased topic focused multi-document summarization," in Proc. 18th IJCAI Conf., pp. 2903–2908, 2007.

[22] X. J.Wan, J. W. Yang, and J. G. Xiao, "Manifold-ranking based topic focused multi-document summarization," in Proc. 18th IJCAI Conf., 2007, pp.2903–2908.

[23] S. Harabagiu and F. Lacatusu, "Topic themes for multi document summarization," in Proc. 28th SIGIR Conf., 2005, pp. 202–209.

[24] Wan X. and Yang J. 2006 "Improved Affinity Graph based Multi-Document Summarization." [4] R. X.Y. Cai, W.J. Li, in "Simultaneous ranking and clustering of sentences: a reinforcement approach to multi-document summarization, 2010,".

[25] Xiaojun Wan and Jianwu Yang "Multi-Document Summarization Using Cluster-Based Link Analysis 2008" [6] Xiaoyan Cai, Wenjie Li "A spectral analysis approach to document summarization: Clustering and ranking sentences simultaneously "inproc X. Cai W. Li / Information Sciences 181 (2011) 3816–3827.

[26] B. Cretu, Z. Chen, T. Uchimoto and K. Miya, "Automatic Summarization Based on Sentence Extraction: A Statistical Approach," International Journal of Applied Electromagnetics and Mechanics, IOS Press, vol. 13, no. 1-4, pp. 19-23, 2002.

[27] Jun'ichi Fukumoto, "Multi-Document Summarization Using Document Set Type Classification," Proceedings of NTCIR- 4, Tokyo, pp. 412-416, 2004.