ISSN: 2321-7782 (Online)
Volume 3, Issue 2, February 2015
International Journal of Advance Research in Computer Science and Management Studies
Research Article / Survey Paper / Case Study
Available online at: www.ijarcsms.com

A Proposed System for A Scalable Database Driven Reverse Dictionary

Nagesh Racha[1]
Department of Information Technology
Isbm School of Technology,
Pune, India

Varsha Balghare[2]
Department of Information Technology
Isbm School of Technology,
Pune, India

Sabir Khan[3]
Department of Information Technology
Isbm School of Technology,
Pune, India

Prof. Seema Bhardwaj[4]
Department of Information Technology
Isbm School of Technology,
Pune, India

Abstract: The enormous availability of words in usage is really becoming a challenging task in finding the exact meaning of words. Sometimes people feel that we could have described a feeling or a situation much more concisely by using a single word, or a shorter phrase. At times people go on describing the situation or a feeling in long sentences. In such situation, reverse dictionary is a useful tool that gives a precise and appropriate word to one's thoughts. 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. A traditional dictionary makes use of forward concept i.e. it accepts input as a word and gives one or more definitions in terms of output. The concept of reverse dictionary is exactly opposite to that of traditional dictionary. It accepts input as a phrase or a sentence describing a concept or situation, and returns a set of precise and appropriate words that satisfy the meaning of the input phrase. In this paper, an implementation of reverse dictionary is proposed which helps the user to get relevant words to the input phrases.

Keywords: Data Mining, Dictionary, Reverse Mapping Set, search process, thesauruses.

I. INTRODUCTION

The proposed reverse dictionary uses the concept of reverse mapping i.e., given a phrase describing a desired concept, it provides words whose definitions match the entered definition phrase. This is not the case in regular forward dictionary. A regular dictionary maps the words to their definitions. For example, a regular dictionary helps the user to get the meaning of the word "regret". The regular dictionary shows the meaning of input word as "to feel sorry." On other hand, the reverse dictionary, offers the user an opportunity to enter the phrase "feeling of loss or longing for someone" as input, and can expect the word "regret" and possibly other words with similar meanings as output. Most of the techniques for the creation of reverse dictionary is based on the creation of multiple databases for synonyms, hyponyms, antonyms etc. In reverse dictionary, the user entered phrase need not necessarily be the same as in the definition, therefore the implementation is done in such a way that the concept of the user input will be considered and corresponding words will be obtained as the outcome. The output results will be ranked according to the perfect similarity of the word to the input phrase to the least possible similarity of the search concept. The reverse dictionary identifies a concept or idea hidden behind the input words or phrases.

In a reverse dictionary, the user input is unlikely to match the definition that is already present. For example: when a user wants to know the word for the concept "Inability to sleep". Whereas, the forward dictionary definition would be rather "sleeplessness during night". Therefore the user input phrase should be conceptually similar to the definitions but need not be exactly the same. This issue is addressed by the proposed reverse dictionary application by using the concept called as building the RMS (Reverse Mapping Set) [1]. According to the concept of RMS, the word which is found in the definition of another word, the latter is mapped to the former [2]. Example, the forward dictionary definition of “insomnia” may be “Sleeplessness during night”. Therefore RMS of sleeplessness, i.e, R (Sleeplessness) will be “insomnia.” This concept can be achieved by considering the key words in the user input. So it is necessary to negate some common words and consider the important words such as nouns and verbs from user input.

For some cases, there are no enough words or less than the target number of words. In such cases the search can be extended using Synonyms and hypernyms. When there are not enough words, the synonym of each word needs to be considered. Example: the synonym of “sleep” will be “relax”. Therefore for word Relax there will be n number of words by using the concept of RMS. In this way search result can be expanded. Consideration of Antonyms plays a very important part in the Reverse dictionary. For example, when a user enters “Cannot sleep”, the negation word “Cannot” need to be considered. Therefore the word “cannot” has the same meaning as “ness” in “sleeplessness”.

II. Proposed System

Report the formation of the WordStar Reverse Dictionary (WRD), a scalable- driven RD system that attempts to address the core issues identified above. The WRD not only full fills new functional ideas sketched above, it does so at an order of magnitude performance and scale improvement over the greatest concept similarity measurement structures available without impacting key quality. This also reveals that the WRD is in solution quality than the two scalable RDs available. Reverse dictionary system is based on the concept that a phrase that conceptually dense a word should bear a resemblance to the words genuine definition, if not matching the exact disputes, then at least conceptually similar. Consider, for example, the following concept phrase: the tower is made of steel girders crises crossed to make it stronger. Based on such a phrase, a reverse dictionary should return words such as mercerize, iron work, and shove.

Following figure shows the proposed architecture of our system.

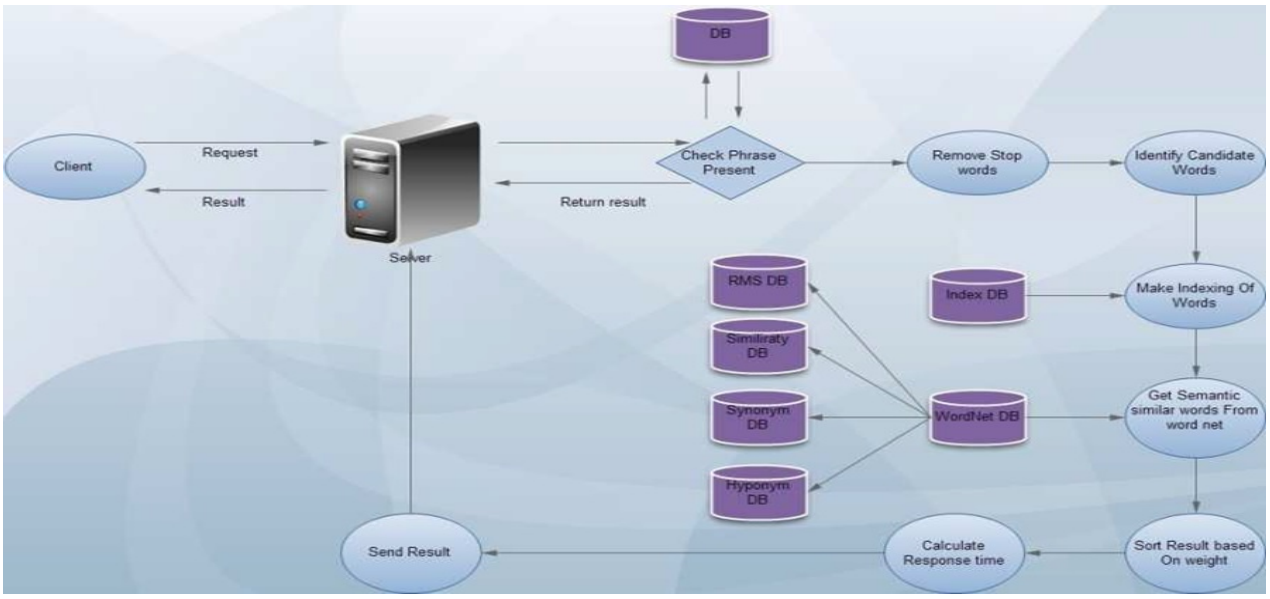


Fig.1 System Architecture

This architecture has three objectives designed to ensure maximum scalability of the system.

1. 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.

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

3. 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.

Following are some of the system features:

1. *Forward Mapping:*

A forward mapping designs all the senses for a particular word phrase.

2. *Reverse Mapping:*

Reverse mapping applies to terms and is expressed as a reverse map set (RMS).Creating a reverse mapping set is infeasible. Thus to create this reverse for every relevant term in dictionary is time consuming. The cost of creating corpus has no effect on run time performance.

3. *Query Building:*

Different algorithms are used for executing and expanding the systems query to and the conceptual similar terms of the type Set Type which can be synonyms, hyponyms or hypernyms. Thus we need to measure the terms similarity across two terms and a term importance it is how critical the term is in the context of the phrase of which it is a part.

III. ALGORITHMS USED

1 K-Means Algorithm:

K-means clustering tends to and clusters of comparable spatial extent, while the expectation- maximization mechanism allows clusters to have different shapes.

Description:

Given a set of observations (x_1; x_2; x_3; ::::; x_n) where each observation is d-dimensional real vector k-means clustering partition the no. of observation into K cluster (i<=n) sets Where S =S1; S2; S3:::::::::S_k so as to minimize the within cluster sum of square i.e

$$j = \operatorname{argmin}_s \sum_{i=1}^{k} \sum_{x \in S_i} ((x - y_i))^2 \quad \ldots\ldots\ldots\ldots\ldots (1)$$

Where y_i is mean of the point of $s_{i,}$

Regarding computational complexity, finding the optimal solution of k-means clustering problem for observation in d-dimensions:

1. NP hard in general Euclidean d even for two Clusters.

2. NP hard for a general no. of cluster k even in plane.

2 Algorithm BuildRMS:

RMS stands for Reverse Mapping Set. It is a mapping algorithm designed to map the word to words of similar meaning. It improves the quality of word mapped i.e. not vulnerable to the input phrase. For an input dictionary D a mapping R is created for all term appearing in the sense phrase. The RMS algorithm describes this reverse mapping pattern.

3 Algorithm GenerateQuery:

Here we generate a query for all the SetType that are mean to be used for mapping and retrieval of reversed term for the given input phrase. Here Query Q is generated for all other algorithm that is returned to get the meaning for the given set of terms in the phrase.The is the building algorithm for the SetType and Sorted queries.

4 Algorithm ExecuteQuery:

For a given query Q if u have phrase that contain terms t_1; t_2; t_3: tk, it performs AND/OR operations in query. If it performs OR operation then the terms of the phrase are union with reverse term and if it performs AND operation then the term of the phrase 11 Building A Scalable Database Driven Dictionary intersect with the reverse term and we returned the union or intersection of the reverse term.

5 Algorithm Expand Antonyms:

Given: A query Q of the form t1; t2; t3; tk, it creates a copy of the query and perform negation to create a sub query to replace all the terms and negated terms. If copy of the query is not equals to the copy of the original query the return copied query or else return its negated terms.

6 Algorithm ExpandQuery:

Given: A query Q of the form t1; t2; t3; ::::::tk, we perform AND/OR operation for all t_i in the query If AND is perform in SetType as synonyms, antonyms and hyponyms, hypernyms to create a subquery q for the above SetType respectively. For OR the term are replaced in query q from Q and at last it return ExecuteQuery.

7 Algorithm SortResults:

Create an empty list K and all the term is arranged in order of its retrieval priority for ease mapping. The sorted term are arrange according to it searched priority i.e term importance, semantic and weighted similarity factor to generate a candidate set that must be ranked using mathematical computation.

IV. RESULT ANALYSIS

Following are the system screenshots:

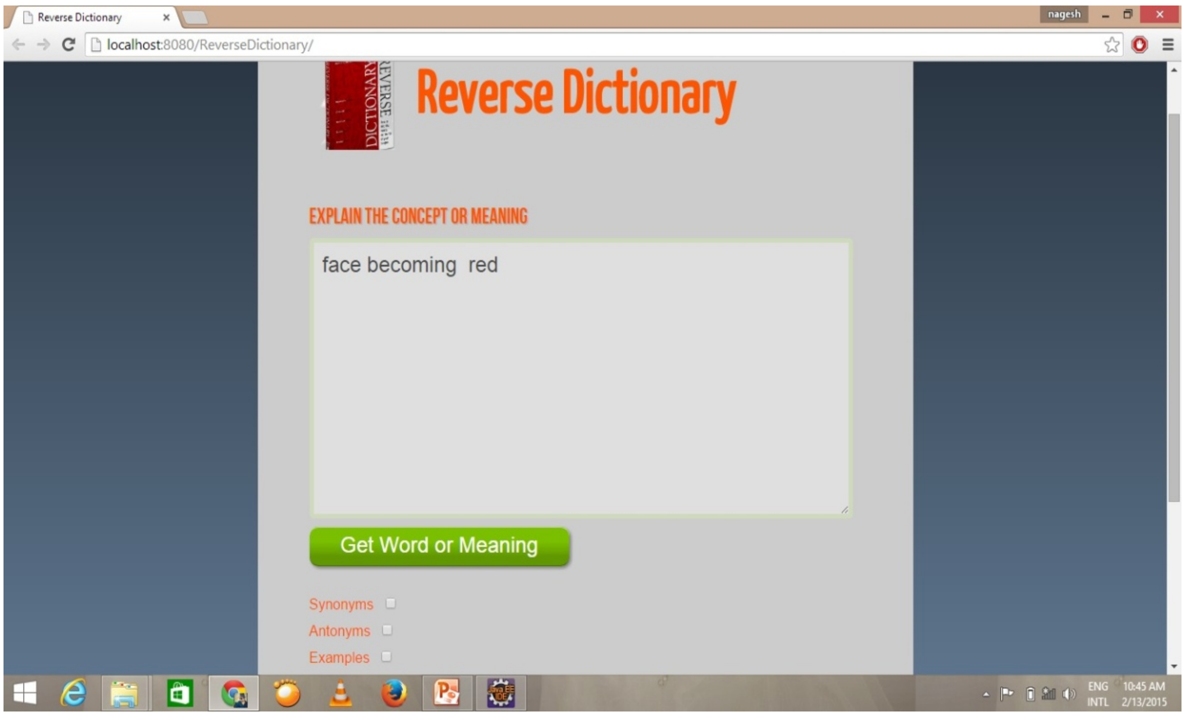
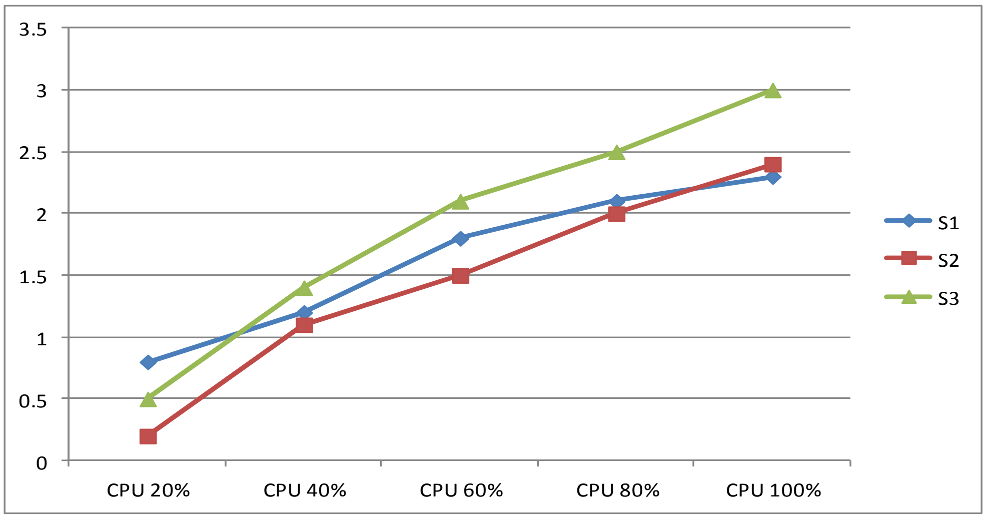


Fig.2 System Architecture

Above figure shows the home page of our system. On this page, user will enter the phrase of words and click on the button to get the required output. For example, if user enters “unable to sleep” then our system will perform Reverse mapping and will get the appropriate and concise word for the entered phrase, like for above input our system will give output as “insomnia.”



In above graph, the efficiency of the proposed system is shown. On X-axis, working efficiency of CPU is plotted and on Y-axis, performance of algorithm is plotted. Three different color lines show the analysis of our system with other existing system. For example in above graph green color line i.e. s3 show our system performance, s2 show the performance of dictionary.com and s1 show the performance of other software on web. So from above analysis we are able to show that our proposed system is better than the other existing system if our system gets the appropriate specification.

V. Conclusion

Thus, in this paper an application for Reverse Dictionary is proposed that analyses the sentences, identify the phrases and returns a set of appropriate words. In the proposed system, the search can be extended by the idea of synonyms, antonyms and hypernyms. The proposed system helps the user to put the thoughts in his/her mind in a more précised manner with appropriate word rather than being verbose and very descriptive. The proposed application of Reverse Dictionary has significant application for those who work closely with words and also in the general field of conceptual search. Mainly, the target users for this application would be linguists, poets, anthropologists and forensics specialist examining a damaged text that had only the final portion of a particular word preserved.

References

1. Ryan Shaw, Member, IEEE, AnindyaDatta, Member, IEEE, Debra Vander Meer, Member, IEEE, and KaushikDutta, Member, IEEE,” Building a Scalable Database-Driven Reverse Dictionary”, 2013.
2. G.Miller, C.Fellbaum, R. Tengi, P. Wakefield, and H. Langone,“Wordnet Lexical Database,” http://wordnet.princeton.edu/wordnet/download/, 2009.
3. Balachandra Reddy Kandukuri, Ramakrishna Paturi V, Dr. Atanu Rakshit, "Data mining and clustering", IEEE International Conference on Services Computing, pp. 517- 520, September 2009.
4. T. Hofmann, Probabilistic Latent Semantic Indexing, SIGIR 99: Proc. 22nd Ann. Intl ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 50-57, 1999.
5. 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.
6. R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval. ACM Press, 2011.
7. T. Korneius, J. Laurikkala, and M. Juhola, On Principal Component Analysis, Cosine and Euclidean Measures in Information Retrieval, Information Sciences, vol. 177, pp. 4893-4905, 2007.
8. D.M. Blei, A.Y. Ng, and M.I. Jordan, Latent Dirichlet Allocation, J. Machine Learning Research, vol. 3, pp. 993-1022, Mar. 2003.
9. T. Joachims, Svmlight, http://svmlight.joachims.org/, 2008 and T. Joachims, Svm multiclass, http://svmlight.joachims.org/ svmmulticlass:html; 2008.
10. E. Gabrilovich and S. Markovitch, “Wikipedia-Based Semantic Interpretation for Natural Language Processing,” J. Artificial Intelligence Research, vol. 34, no. 1, pp. 443-498, 2009.