Enhance the Reverse Dictionary Using Jaccard Indexing Algorithm

J.Mohan, PG Scholar
Department of Computer Science and Engineering, Bharath University , Chennai, India
Mohan191@gmail.com

2.Dr.K.P. Kaliyamurthie *Prof & Head*
Department of Computer Science and Engineering, Bharath University , Chennai, India
kpkaliyamurthie@gmail.com

***Abstract*— in today's business environment people working in the different areas often struck because unable use appropriate word. Some using the traditional dictionary usually containing a list of word arranged in the alphabetical words along with their meanings. It usually maps word to their definitions and it will not accept the word in non standard order. So we introduce the concept called Reverse dictionary. Reverse dictionary take input phrase from the user and establish the relation between them, and display the single word meaning for given. If the application find no match is available then it display list of words with the relevant meaning. The results are produced much faster without compromise the accuracy of our implementation.**

Key words: Dictionaries, thesauruses, lexicon, search process,

1. INTRODUCTION

In this article we are going to apply the concept of Data Mining to create the reverse dictionary. The amount of word stored in normal dictionary is growing day by day. At the same time using the dictionary wants to express their ideas or thought by using more particular words in their profession. But in the regular dictionary only allow you to search the word to their definition. People no longer satisfied with word arranged in alphabetic order. Reverse dictionary allow the people to solve these constraints.

When user entered input into the dictionary. It then applies Knowledge Discovery in Database (KDD). KDD is the process of identify useful patterns from large and complex data sets [12]. This KDD process includes 1) Accepting the data from the user 2) Preprocessing include applying the stemming algorithm to remove the error and unwanted data from the user input. 3) Transformation is a process of converting or shortened the given word into the root word. 4) Apply the set of algorithm for mining process. These algorithms expand the given input to make the result more accurate. 5) Now evaluate the given output. Apply the Jaccard similarity to identify and choose the words with high similarity value and display the result. Reverse dictionary enable automated text identification and useful for the people working in the field of lexicography.

II. RELATED WORK

Normally end users become impatient if a website takes longer than 4-5 seconds to respond to a request. [4]

Porter Stemming algorithm widely used for removing commoner and reduce word to the single root in the English language. Using this algorithm we generate more generic word that has better visibility. [3]

Upon receiving a user input in the reverse dictionary search the normal dictionary at its database and selects the words whose definitions are similar to this user input. These words then form the output of this reverse dictionary lookup.[1]

In the reverse dictionary which contain very little contextual information (often consisting of fewer than ten words). The lack of contextual information in the reverse dictionary case adds to the difficulty of addressing this problem. To avoid those problem use the hypernyms density representation leads to significantly more accurate and more comprehensible rules.[9]

In the reverse dictionary case, since our concepts are not known beforehand. In notably within the case of user inputs, we want to figure the user input construct vector at runtime and so afterward figure the gap between this and also the wordbook construct vectors (where these vectors may be computed a priori). Vector computation is thought to be quite computing intensive despite the fact that abundant effort has been exhausted in building economical schemes. The results of this effort are many revealed variants of LSI such as pLSI and Latent Dirichlet Allocation (LDA).[13]

Researchers have attempted to enhance the scale of the efficient variants by creating parallelized and distributed avatars. In this paper we are going to enhance the Wordster Reverse Dictionary (WRD), a database-driven reverse dictionary system that attempts to address the core issues identified above.

The WRD not solely fulfils new useful objectives printed higher than it will thus at associate degree order of magnitude performance and scale improvement over the most effective thought similarity schemes obtainable while not impacting resolution quality.

III. PROPOSED SYSTEM

Reverse dictionary allow the user to enter the word in non standard order which is not possible in the standard dictionary. But most reverse dictionary suffer from the concept similarity problem (CSP) and vectorization is done prior only vector distance is calculated. To avoid problem mentioned above. We need to access the information stored in the number of database. Apply the knowledge discovery in database (KDD). KDD is the process of identify useful patterns from large and complex data sets. This process includes

1) Reverse dictionary accept the input from the user and display the set of similar words.
2) We must create mapping for all the word appear in the definition phrase in the dictionary.
3) Now apply the Porter Stemming algorithm to remove unwanted word or reduce any particular word.[3]
4) Convert the user input into the most generic form of the word possible.
5) Convert all the negation word into the Antonyms words
6) Due to less conceptual information available from the user input. We can add synonym set along with the user input.
7) Hypernyms is a super ordinate word provides the better visibility for the user data.
8) Hyponyms is a sub ordinate word provides more specific word for the user input.
9) Expand the user input for better visibility.
10) Create the parse tree.
11) Execute query using the Boolean expression consist of n number of term and produce the output
12) Sort the result using the Jaccard similarity indexing and display the result in the descending order.

IV. Architecture of Reverse dictionary

A. Porter Stemming Algorithm

This algorithm widely used for removing commoner and reduces word to the single root in the English language. Example: connect, connected, connecting, connection, connections into the single root word Connect [11]. When the user enter the input initial it remove the Punctuation, number, symbol and then it split the sentence into the words and remove the stop word from the user input. And then it Remove the suffixes helps in the field of information retrieval system. Information retrieval is usually document contain the vector of words.[2]

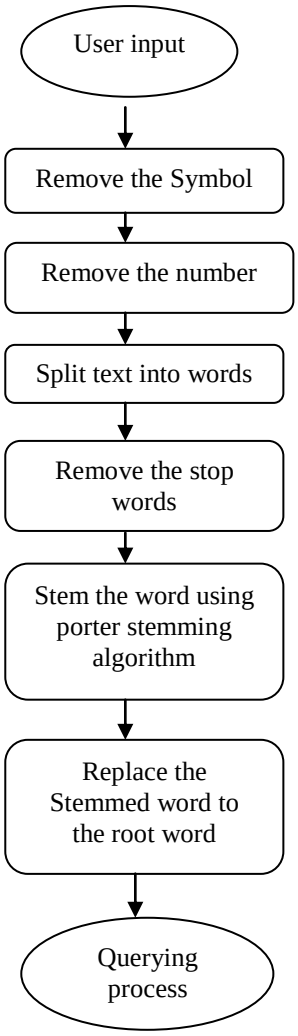

Figure: 1

Step 1: Remove the punctuation from the given sentence.
Step 2: Remove the specific symbol from the given sentence.
Step 3: Remove the number from the given sentence.
Step 4: Now split the sentence into the words
Step 5: Remove the common English words
Step 6: Remaining words are shortened to the single root word for better visibility.
Step 7: Now send those words to the querying process.

Create R mappings for all terms appearing in the sense phrases (definitions) in D. [1]

B. Preprocessing:

Remove unwanted words:

After getting input from the user we need to generate the query that produces the sufficient number of output. We need

to remove all the common words, symbol and number from the given input.Now convert all the negation words into the equivalent antonyms words in the English.

Split the words:

Split the entire user given input into the words.

Example: Suppose User entered "Tell about yourself"
Covert into {Tell,
About,
Yourself}

Convert Negation word:

Convert all the Negation word into the equivalent Antonyms word and join with the existing word using union.

Example: Suppose user enters "not creative" we need to convert into the antonym set like Antonyms (Creative) = {sterile, unimaginative, uninspired, and uninventive}. [11]

Generate the table:

After split the user input into the word. Now search each word separately in the dictionary and find the list of attribute in the table for each word.

Table 1

Noun	Verb	Adjective	Adverbs

Now store the table in the temporary database. We can get above attribute in the WorldNet Dictionary [12].

C. Transformation

Add Synonyms:

We need to find the synonym for each input word. It will increase the visibility and able to identify the similarity between data set.

Example:
Synonym [creative] = {fanciful, notional, fictive, imaginative, inventive, yeasty}.[11]

Add Hypernyms:

Hypernyms are super ordinate word. It is particular more generic word and able to wide our area of finding similarity of our dataset. Hypernyms are widely used in the text classifications. [3]

Example:
User input is 'painter' then

Hypernyms (painter) = {"artist"}

Artist is more generic form of the word painter.

Add Hyponyms:

Hyponyms are the sub ordinate word. It more specific word for any particular word. It will be used if we get large number of output. It helps to find the more similar word in our dictionary.

Example:
Hyponyms (painter) = {colorist, cubist, distortionist, impressionist}

D. Evaluation of query:

Now evaluate the query. Evaluation is a process of searching the word in the table1. Evaluation query includes the synonym, hypernyms along with the user input for the better visibility.

Algorithm 2:
Step 1: Get the input from the user

Step 2: Apply the potter stemming algorithm to the input and get the list of words from the stemming algorithm

Step 3: Convert the negation words into the antonym word.

Step 4: Get the list of Nouns, Adjective, Verb and Adverbs and stored in the database table.

Step 5: Expand the user input by including the similar lexicon like Synonyms of the given word and also include hyponyms and hypernyms.

Step 6: We need to evaluate the query by merging the different lexicon using the intersection and similar lexicon using the union.

Step 7: Now execute the query and search the similar word in the database table mentioned in step4.

Step 8: Sort the result using the Jaccard similarity indexing.

Step 9: Display the output.

E. Sorting the Result:

Jaccard Indexing

Jaccard indexing is used for comparing the similarity of the any given sample set. Jaccard algorithm is one of the simplest method for evaluate the similarity of the particular sample to the given sample set.[14]

$$J(A, B) = A \cap B / AUB$$

If you what to find the similarity between two words A, B then the mapping range is from [0, 1]. Usually similar words have a higher similarity value. Similarity value are usually expressed as

$$\textbf{Jaccard (A, B)} \sum \textbf{[0, 1]}$$

- Jaccard (A, B) =1 mean that word are alike.

- Jaccard(A,B)=0 mean that word are not alike at all

User input now checking against the table 1. Initial combine the pair of word and identifies the similar in the table1. Then add plus one word and again searched in the dictionary. Which English word that has some similar to the user input displayed first and remaining words are sorted in the descending order.

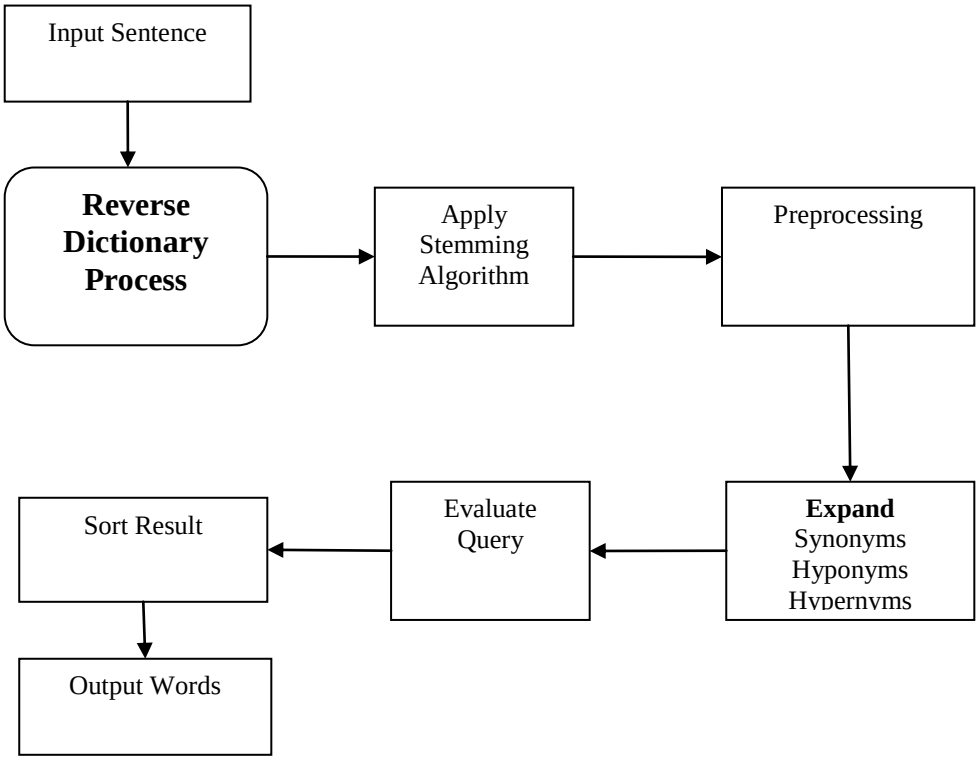

Figure 2: Architecture for Reverse Dictionary

V. EXPERIMENT AND RESULT:

In this experiment environment consist of Netbeans IDE and WordNet database stored in the text files[12].

Phase1 User Input	Phase 2 Stemming Algorithm	Phase 3 Transformation and result

In the following database, we take a sample of 5000 words and using our algorithm to execute the result. [12]

Phase1:
User Input: 'someone who works metal'

Phase2:

Input: 'someone who' is removed from the user input and apply the stemming algorithm.

Works is further reduced to work

Output is split into {metal, work}

Phase 3:
Transformation :

Input: (metal U work)

Now find the Noun, verb, adjective and adverbs of the words that has either metal or work and store it in the temporary table.

Phase 4:
Sort the result
Jaccard {metal, work} = metal ∩ work/metal U work

The similarity of the Jaccard indexing lies between (0, 1). Now search in the temporary table and identifies the most similar word. If the Jaccard index closes one mean that word has the highest similarity. Now sort those words in the descending order.

Result:

Blacksmith
Metal worker
Miller
Jack
Scab
Swage
Forge
Boilermaker

The above words are display instant without comprises the accuracy of the solution.

V. CONCLUSION AND FUTURE WORK

This paper provides set of algorithm for implementing the reverse dictionary and displays the best quality result. Our experiment show it provide better quality result without comprise the performance. Wordster reverse dictionary is one of the best reverse dictionary comparing the existing reverse dictionary like Dictionary.com [15] and OneLook.com[16]. We can further improve the performance of this dictionary by implementing the k mean clustering method. K means clustering classify the word more effective than Jaccard Indexing and replace in your future work.

***K*-means clustering** is widely used of cluster analysis which aims to partition *n* observations into *K* clusters in which each observation belongs to the cluster with the nearest mean.[3]. Item are moved along the clusters until the desired set is reached. By using this algorithm high degree of similar word reached one set and high dissimilar word are moved to another set. It will continue until no more changes in any set.[14].

REFERENCES

[1] Ryan Shaw ,Anindya Datta,Debra VanderMeer and Kaushik Dutta "Building a Scalable Database Driven Reverse Dictionary" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 25, NO. 3, MARCH 2013
[2] "Algorithm for Suffix Stripping",M.F. Potter, Computer Laboratory, corn exchange street, Cambridge, 2013
[3] Dileep Reddy Chintala and E.Madhusudhana Reddy "An Approach to Enhance the CPI Using Porter Stemming Algorithm," Volume 3, Issue 7, July 2013
[4] Forrester Consulting, "Ecommerce Web Site Performance Today,"http://www.akamai.com/2seconds, Aug. 2009.
[5] 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
[6] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity," Proc. Nat'l Conf. Artificial Intelligence, 2006
[7] F. Sebastiani, "Machine Learning in Automated Text Categorization," ACM Computing Surveys, vol. 34, no. 1, pp. 1-47, Mar. 2002
[8] N. Segata and E. Blanzieri, "Fast Local Support Vector Machines for Large Datasets," Proc. Int'l Conf. Machine Learning and Data Mining in Pattern Recognition, July 2009
[9] Sam Scott and Stan Matwin "Text Classification Using WordNet Hypernyms," Volume 1, August 2006
[10] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval.ACM Press, 2011.Mar. 2003.
[11] http://www.synonym.com/antonyms/
[12] http://wordnet.princeton.edu/
[13] D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," J. Machine Learning Research, vol. 3, pp. 993-1022,
[14] Margaret H. Dunham, "Data Mining Introductory and Advanced Topics", Pearson Education, 2003 , Second Edition.
[15] http://dictionary.reference.com/
[16] http://www.onelook.com/