ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

Building a ranking based reverse dictionary

Kesavaram.P.H, T.Muthamilselvan

Final year M.S(software engineering) student, School of Information Technology and Engineering, VIT University, Vellore, Tamilnadu, India

Assistant Professor (Senior), School of Information Technology and Engineering, VIT University,Vellore, Tamil nadu, India

ABSRACT: A traditional forward dictionary is where the user inputs a word and gets one or more definitions as output. Whereas, in a reverse dictionary the user enters the desired phrase with a logic and gets a certain number of words as outputs similar to the concept of the user input phrase. The user entered phrase need not necessarily be the same as in the definition, therefore we implement in such a way that the concept of the user input will be analyzed and corresponding words will be obtained as the result. 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.

KEYWORDS: dictionary, reverse, ranking, RMS, WordNet, data mining, term similarity, term importance

I. INTRODUCTION

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. The building of reverse dictionary addresses this issue.

For the Reverse dictionary the main concept that needs to be implemented is the Building the RMS [1]. RMS means Reverse Mapping Set. According to which, 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”The way to achieve this concept is to consider the key words in the user input, i.e, It is necessary to negate some common words and consider the important words such as nouns and verbs.For the inflected forms, An algorithm is used. According to which, the inflected forms will be converted into a root form. Example: “Sleeplessness during night time”, In this example the verb “Sleeplessness” plays a very important role. But it is an inflected form of “Sleep”. So we have to use the Algorithm to convert the inflected form “Sleeplessness” to its root form “sleep”.

In case there is no enough words or less than the target number of words, we can extend the search using Synonyms, hypernyms, hyponyms [1]. When there is not enough words, the synonym of each word needs to be considered. Example: the synonym of “sleep” will be “relax”. Therefore the R(Relax) will be n number of words. Therefore this allows us to expand the search result. Considering Antonyms, they play a very important part in the Reverse dictionary. Because when a user enters “Cannot sleep”, the negation word “Cannot” needs to be addressed. Therefore the word “cannot” has the same meaning as “ness” in “sleeplessness”. Thus we should still get the result as “Insomnia”.On the other hand, if there are more words than the threshold value, It can be reduced to the limit. That is, if the number of values must not exceed 10, but if we have “20”, 10 words need to be removed from the list. For this, we have to consider which word is found in most of the definitions, and the R(w)s must be removed . Example: R(what) is w1,w2,…,w10; whereas R(sleep) is w1 and w2. Thus words w1,w2,.. w10 of R(what) will be removed. By doing this, we finallyget a satisfying N number of results.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

II. RELATED WORKS

In a reverse dictionary a user input phrase is given and we receive certain number of words as output ranked with an algorithm. The related works to this reverse dictionary are the reverse dictionaries [3][4] which is built with certain drawbacks.Such work is in the RD case, where the concepts are obviously unknown already , especially in the case of user inputs, It is necessary to compute the user input during the run-time. Vector computation is known to be quite compute intensive. Hence the vectorization which is used in the current system, uses the LDA(Latent Difficult Association) and PCA(Principal Component Analysis) [11].The other related work to a reverse dictionary is the other existing features of a reverse dictionary where certain constraints are to be addressed.The two main constraints needs to be addressed.The first constraint is that, the user input will not match the one in the forward dictionary. Next, the response time needs to be similar to that of a forward dictionary.

The existing dictionary receives an input phrase and outputs many output words, therefore it can be tedious for the user to search one from it. Whereas, building a ranking based dictionary, allows the user to choose from a set of words which are closely related to each other.Example: In the existing reverse dictionary, the user inputs a phrase “unknown name” outputs over a 100 search results including “anonymous”, “nameless”, “incognito”, “jane doe”, “john doe”, “some”, “sky-blue pink”, “dark horse” etc. [4], But the most accurate result for “unknown name” shall be “anonymous”, “nameless” but “incognito” also contributes some primary meaning to the user input phrase. Whereas words like “sky-blue pink”, “challenge”, “key” doesn’t have anything to do with the concept but It is associated with the word “unknown”. Reverse mapping Set,RMS of t, denoted by R(t) is mapped to the “definition” in the dictionary .Every word that contains the definition, forms the suggested output. Finally, they are arranged according to the rank.

III. PROPOSED APPROACH FOR REVERSE DICTIONARY

In a reverse dictionary the user enters the desired phrase with a logic and gets a certain number of words as the output.To do so, we have to first build the RMS(Reverse Mapping Set). Building an RMS means to find a set of words in whose definitions any word ‘w’ is found. Example: The word “sleep” is found in 4 definitions belonging to 4 words. Therefore R(sleep) will be “slumber”, “sopor”, “nap”, ”rest”. These words must be manually entered for each word. The RMS of the words can be found from the wordnet [2][6] dictionary.The stop words like “whereas, whenever, however, very” etc needs to be negated as they don’t form a very important part of theprocess. Whereas, Antonyms are needed to be addressed. Example: When the word sleep is followed by “cannot”, the antonym of “sleep”, which is “wake up” should be considered for the search process.

The search can be enhanced by considering the synonyms, hypernyms and hyponyms of that particular word. When the words do not yield enough output. The synonym, hypernym and hyponym of that particular word is considered and the RMS of those words also form a part of the output words.A synonym is the other possible meaning of the word, where as a hypernym is the common class under which the word occurs. Example: The word “parrot” belongs to “birds”, therefore the hypernymof “parrot” is birds. Whereas the hyponym is the other similar birds like “macaw” etc.Considering synonyms, hypernyms and hyponyms will increase the number of output words. i.e, if “parrot” doesn’t yield enough results; the synonym, hyponym and hypernym of parrot will yield more results.

The final step is to sort out the results. This is done when there are more number of output words. Example: In the existing reverse dictionary, the user inputs a phrase “discrimination based on colour” outputs over a 100 search results including “racism”, “classism”, “judgemental”, “colour bar”, “colour line”, “red”, “nepotism”, “fair” etc. But the most accurate result for “discrimination based on colour” shall be “racism”, but “colour line”, “colour bar” also contributes some primary meaning to the user input phrase. Whereas words like “fair”, “red” doesn’t have anything to do with the concept but It is associated with the word “colour” [3]. In order to avoid such unnecessary results we decrease the number of search

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

words.By decreasing the number of outputs, we finally get the number of words which are closely associated with the search concept. Therefore, according to the previous example, when a user enters a phrase “discrimination based on colour”, the output shall be “racism”, “prejudice”, “colour line”, “colour bar”, “nepotism” etc.Inorder 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 [1]. Example: “discrimination based on colour”, the word “racism” consists of both “discrimination” and “colour”, therefore it must be given more priority.we arrange the output words in the descending order by the means of number of words in whose definitions that word is present. Example: R(how) gives 5 words, and R(Sleep) gives 3 words. The 3 words which the word “sleep” yields is considered rather than “how”.

A. System Architecture

The Reverse dictionary is a computer application which takes the user input phrase and gives the corresponding words as output.

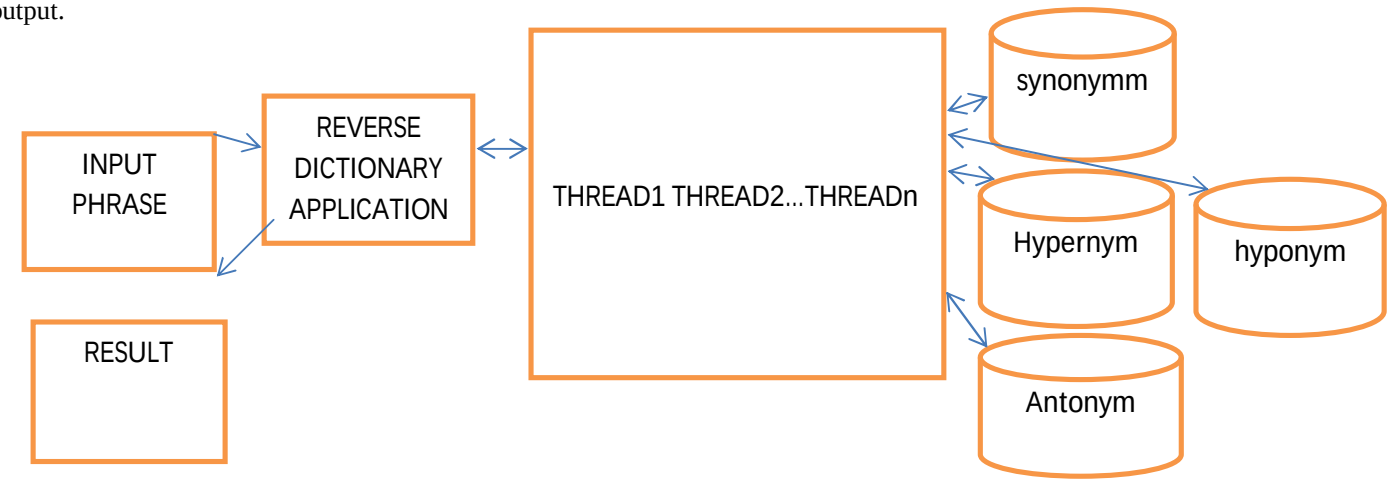


Fig.1 System architecture of reverse dictionary

The RMS contains a set of mappings, the dictionary definitions andparse trees [8] 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, we consider three characteristics. First, A frequently accessed data is stored, which lets the thread to get therequired 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

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.

IV. SELECTING THE OUTPUT BASED ON RANKING

The reverse dictionary outputs a number of words as a result for the given input phrase. To acquire more precise words, it is necessary to rank the outputs based on the similarity to the search concept.A set of words in decreasing similarity to the user input phrase, based on syntactical similarity of the definitions of the corresponding word as compared to the user input phrase.Inorder to build a similar ranking, It is necessary to assign a similarity measure for each sense and user input phrase [8], i.e, the sense phrase and the user input phrase pair. Simply considering the pairwise similarity will provide a meaningful measure of similarity whereas the order of the words in a phrase is important. Example: consider these two phrases. “The man catches a fish”, “the fish catches the man”. The both sets of phrase contain the same nouns and verbs but the meaning of these two sentences varies very much [6].Therefore, it is necessary to have the measure ofthe similarity of a term and importance of the term. That is, to measure how critical the term is in the context of the phrase [9][14].Let us consider the term a as “man” and b as “fish”. That is, a and b are two words from the user input phrase.

Term similarity :It is necessary to consider various functions to use in evaluating the similarity of two terms a and b which returns the least common ancestor shared by both a and b found in the WordNet[2][6] hierarchy. Using these two functions, we consider a function of similarity that describes how alike the two termsa,b are. If a,b is larger than a and b are similar to their least common ancestor in the hierarchy.Term importance: Let us consider, the term importance t, the word in the user input phrase and p, the phrase. In the already mentioned example, “the man catches the fish”, the subject is more important than the object. To find out the importance of each term we generate the parsertreee, e.g, OpenLP [5],that can return the grammatical representation of the sentence. Therefore, a parser tree of p, pt(p) and depth d of a term, where pt(p) has a depth d(pt(P)) in total, to obtain a measure of the importance of t to the meaning of P [10]. Weconsider a function for term importance between the word in the user input phrase, t and the phrase,p.Therefore, if the value of t is higher in the parse tree, the term importance of t,Pwill be larger.

The final similarity measure factor generates each word pair a and b, where “a” belongs to “S”, sense phrase and“b” belongs to the user input phrase. Therefore, this factor as an input to the algorithm which is responsible for the similarity of the string to get a similarity measure of a phraseS and U for a sense phrase and a user input phrase [6]. Finally,Werank the outputs similar to U, and return the maximum number of outputs according to the threshold.

V.IMPLEMENTATION AND RESULT ANALYSIS

The Reverse dictionary is implemented where the user phrase is taken as the input and a set of words suggested are obtained as the result. The output results which are obtained can be of any number. Therefore, it is necessary to set a minimum and a maximum value. According to the setup of a minimum value, we can expand the output results. Say, we setup the minimum value to be 7 but we get only 5 results, using the expansion of the query, we can easily get more than 7 words as output. We do this by considering the synonym, and if we still do not get enough outputs, we consider the hyponyms and hypernyms. In case there is more number of outputs than the maximum number defined. We use another technique. The number of outputs is arranged in an order so that the word which has more number of occurrences in the definitions and those corresponding words are placed on top. Then that word should be removed from the list. By doing this, we reduce the number of words until it reaches the maximum limit. Practical implementation shows that, when the number of suggestions or the output increases, the accuracy of the output results gradually will decrease. When we consider the number of output words being more, the accuracy of the words to that of the user input phrase will be less. Above a certain number of suggestions, the accuracy becomes stable.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

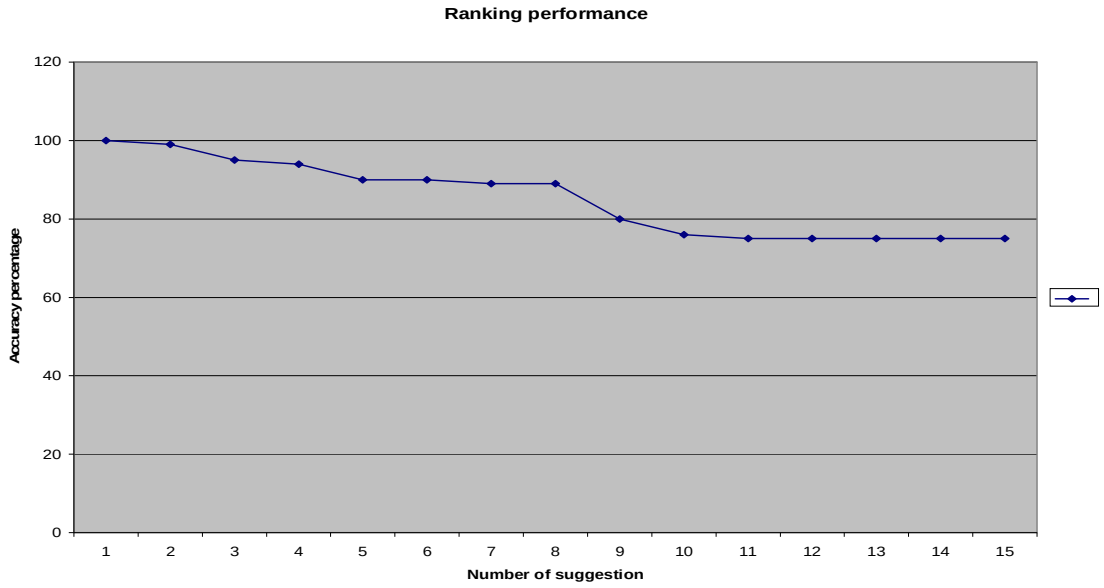
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

NUMBER OF SUGGESTION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
ACCURACY PERSENTAGE	100	99	95	94	90	90	89	89	80	76	75	75	75	75	75

Table 1: Result of ranking performance



V. FUTURE WORK

The reverse dictionary works in such a way that, given an input phrase, an word related to that phrase is given as output. For the inflected forms, an algorithm is used. According to which, the inflected forms will be converted into a root form. Example: “Sleeplessness during night time”, in this example the verb “Sleeplessness” plays a very important role. But it is an inflected form of “Sleep”. So we have to use the Algorithm to convert the inflected form “Sleeplessness” to its root form “sleep”. For which, an algorithm called Porter-stemming algorithm is used.According to which a word which is in its inflected form is converted to its root form. As already seen in the example, when the user input phrase consists of “flying”, it must be converted to its root form, which is “fly”, so that it can be searched within the database.

But there are certain spelling changes which cannot be avoided. Example: “Ponies”, “flies”, ”skies” are the inflected forms of “Pony”, “fly”, “sky”. But according to the stemming algorithm, the result would be “Poni”. Therefore there is a “i” in the place of “y”. This calls for a spelling check. The method to overcome this problem would be spelling check. As soon as the inflected form is converted to its root form, the root form must be spell checked, and when it is an “i” instead of “y”, it can be auto corrected before checking them with the database.The integration of spell checker can be useful inorder to provide efficient results as the output.Eventhough, a meaningful information regarding the implementation of the existing reverse dictionaries cannot be provided, with the help of some corrections, an effective reverse dictionary, which gets a user input

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 4, April 2014

phrase and outputs a set of words, according to the priority and also in the ascending order of the words from the most conceptually similar to the least can be obtained.

VI. CONCLUSION

In this paper, we discuss the ways to build a reverse dictionary and also note down the existing problems that occur in the process. We, therefore, suggest a group of methods for constructing and inputing a reverse dictionary, and show the result's quality and also the run time and scalability. The quality of this approach shows a greater improvement in the quality and run time as compared to the existing reverse dictionaries, which are onelook.com and dictionary.com. Hence, upon receiving a user's search concept, the reverse dictionary will successfully output the relevant set of words.

REFERENCES

[1]Ryan Shaw, Member, IEEE, AnindyaDatta, Member, IEEE, Debra VanderMeer, Member, IEEE, and KaushikDutta, Member, IEEE," Building a Scalable Database-Driven Reverse Dictionary",vol.25, pp. 528-540,2013.
[2] G.Miller, C.Fellbaum, R. Tengi, P. Wakefield, and H. Langone,"Wordnet Lexical Database," http://wordnet.princeton.edu/wordnet/download/, 2009.
[3] Dictionary.com, LLC, "Reverse Dictionary,"http://dictionary.reference.com/reverse, 2009.
[4] OneLook.com, "Onelook.com Reverse Dictionary," http://www.onelook.com/, 2009.
[5] O.S. Project "Opennlp," http://opennlp.sourceforge.net/, 2009.
[6] T.Daoand T. Simpson, "Measuring Similarity between Sentences,"
http://opensvn.csie.org/WordNetDotNet/trunk/Projects/Thanh/Paper/WordNetDotNet_Semantic_Similarity.pdf, 2009.
[7]http://en.wikipedia.org/wiki/Data_mining
[8] J. Earley, "An Efficient Context-Free Parsing Algorithm," Comm.ACM,vol. 13, no. 2, pp. 94-102, 1970.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.
[9] D. Lin, "An Information-Theoretic Definition of Similarity,"Proc.Int'l Conf. Machine Learning, pp. 296-298, 1998
[10] Z. Wu and M. Palmer, "Verbs Semantics and Lexical Selection,"Proc. 32nd Ann. Meeting Assoc. for Computational Linguistics,pp. 133-138, 1994.
[11] D. Widdows and K. Ferraro, "Semantic Vectors," http://code.google.com/p/semanticvectors/, 2010.
[12] N. Segata and E. Blanzieri, "Fast Local Support Vector Machinesfor Large Datasets," Proc. Int'l Conf. Machine Learning and DataMining in Pattern Recognition,pp. 295-310 ,July 2009.
[13] U. of Pennsylvania, "The Penn Treebank Project," http://www.cis.upenn.edu/ treebank/, 2009.
[14] R. Mihalcea, C. Corley, and C. Strapparava, "Corpus-Based and Knowledge-Based Measures of Text Semantic Similarity,"Proc.Nat'l Conf. Artificial Intelligence,pp. 775-780,2006.
[15] C. Manning, P. Raghavan, and H. Schutze, "Introduction to Information Retrieval",.Cambridge Univ. Press, 2008.