Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Survey On Building A Database Driven Reverse Dictionary

Akanksha Tiwari[1], Prof. Rekha P. Jadhav[2]
[1,2]G.H.Raisoni institute of engineering and technology(GHRIET),Pune

Abstract: Reverse dictionaries are widely used for a reference work that is organized by concepts, phrases, or the definitions of words. This paper describe the many challenges inherent in building a reverse lexicon, and map drawback to the well known abstract similarity problem The criterion web search engines are basic versions of system; they take benefit of huge scale which permits inferring general interest concerning documents from link information. This paper describe the basic study of database driven reverse dictionary using three large-scale dataset namely person names, general English words and biomedical concepts. This paper analyzes difficulties arising in the use of documents produced by Reverse dictionary.

Keywords: *Reverse Dictionaries(RD), Phrase, Lexicon, Database and WordNet.*

I. INTRODUTION

Since last decade, people have used dictionaries for two well-defined purpose. First is to find the meaning of specific word with their equivalent in another language. Second is to find the words listed alphabetically in specific language which contain their usage information , definition, phonetics, pronunciations and other linguistic features. When these ideas comes together we, understand , why this resource has not lost important and continue to be widely used around the world.

The change in technology evolution from last years , dictionaries are now available in electronic format An online dictionary [1] is a dictionary that is accessible via the Internet through a web browser. Basically two types of online dictionary .

1. Forward dictionary : Dictionary is one which maps from word to their definition. Example : 'chef' : is a person who is a highly skilled professional cook who is proficient in all aspects of food preparation.

2. Reverse dictionary : we already had the meaning or the idea but aren't too sure of the appropriate word, then reverse dictionary is the right one for use. Example : person who is a highly skilled professional cook who is proficient in all aspects of food preparation :- ' chef'.

In order to build a reverse dictionary, first a forward dictionary is needed. WordNet is the forward dictionary used in this work. WordNet [2] is a lexical database which is available online and provides a large repository of English lexical items. WordNet .Such dictionaries have become more approach discussed in [5] deal with the pre-creation of a context vector for each word in WordNet during the learning phase.

Three large datasets are used to build reverse dictionary such as, dataset s with person names, biomedical concept names, and general English words.Again classifying these dataset into the five databases simultaneously. Five database such as synonym db which give the relevant meaning for that important word, RMS db creates the parse tree for that dictionary definition , hyponym db a word that is

more specific or generic than a given input word , antonym db which gives opposite answer to given word and definitions db is describing the word briefly[8].
This paper contributes as II section Literature survey, consist of survey on the Reverse dictionary approach. and later discuss the problems and constraints with existing system. In section III representing future enhancement. In section IV consist of conclusion of the this survey.

II. LITERATURE SURVEY

In recent years many research has done over database driven reverse dictionary. the idea of arranging the vocabulary of language in reverse order is not new. Since 19th century the reverse dictionary were published, as simple collection of words. In 1915 in Russian language,the first reverse dictionary of modern language , compiled for the purpose of decoding the military news. In fifties and later many reverse dictionaries were published on many different languages. In traditional model for using dictionary, forward concept is implemented where it result in set of definition and it may produce a comprehensive phases. To facilitate forward concept, user provide reverse dictionary in which for any phases or word, the appropriate single word meaning is given. System will provide the relevant meaning even if that word is not available in the database. Virtually all attempts to study the similarity of concepts model concepts as single words[4]. Work in text classification for instance, surveyed in detail in [3], attempts to cluster documents as similar to one another if they contain co-occurring words (not phrases or sentences). Current word sense disambiguation approaches appears, still consider a single word at a time[5]-[7].Also there exists some work on multiword addresses ,the problem of finding the similarity of multiword phrases across a set of documents in Wikipedia[19].

Several studies addressed different paradigms for approximate dictionary matching. Bocek etal. (2007) presented the Fast Similarity Search (FastSS), an enhancement of the neighborhood generation algorithms, in which multiple variants of each string record are stored in a database[10].
Wang et al. (2009) further improved the technique of neighborhood generation by introducing partitioning and prefix pruning. Huynh et al. (2006) developed a solution to the k-mismatch problem in compressed suffix arrays. Liu et al. (2008) stored string records in a trie, and proposed a framework called TITAN[11].
These studies are specialized. Several researchers have presented refined similarity measures for strings (Winkler, 1999; Cohen et al., 2003; Bergsma and Kondrak, 2007; Davis et al., 2007). Although these studies are sometimes regarded as a research topic of approximate dictionary matching, they assume that two strings for the target of similarity computation are given; in other words, it is out of their scope to find strings in a large collection that are similar to a given string. Thus, it is a reasonable approach for an approximate dictionary matching to quickly collect candidate strings with a loose similarity threshold, and for a refined similarity measure to scrutinize each candidate string for the target application.

A. Dataset

Reverse Dictionary Application is a software element that captures a user phrase as input and returns theoretically connected words as output. It requires large amount of dataset to get accurate meaning of the word. There exists a three large datasets and simultaneously database for synonyms, hyponyms and antonyms.

1. Person name: This dataset comprises actor names extracted from the IMDB database6. We used all actor names (1,098,022 strings; 18 MB) from the file actors.list.gz.The average number of letter trigrams in the strings is 17.2. The total number of trigrams is 42,180. The system generated index files of 83 MB in 56.6 s.

Table 1 : Literature survey on Reverse Dictionary

Author/Year	Method	Dataset	Remark
Yuhua Li, David McLean, Zuhair A. Bandar 2006 (IEEE) [13]	Sentence-similarity	Lexical dataset	varied sentence pair data set with human ratings and an improvement to the algorithm to disambiguate word sense using the surrounding words to give a little contextual information
Naoaki Okazaki and Jun'ichi Tsujii 2010 [8]	CP Merge algorithm and n-grams feature approach.	Personal names, general English words and biomedical names.	solved ῑ overlap joins by checking approximately half of the inverted lists with cosine similarity and threshold α= 0.7).
Anindya Datta and Kaushik Dutta March 2013 (IEEE) [1]	concept similarity problem (CSP)	Dictionary contain synonyms, antonym and hyponyms .	propose a set of methods for building and querying a reverse dictionary, and describe a set of experiments that show the quality of results.
Oscar Méndez, Marco A. Moreno-Armendáriz 2013 (IEEE) [14]	Semantic approach	WordNet, semantic dataset	applying algebraic analysis on dataset then filtering process and a ranking phase. Finally, a predefined number of output target words are displayed

2. GoogleWeb1T unigrams: This dataset consists of English word unigrams included in the Google Web1T corpus (LDC2006T13). We used all word unigrams (13,588,391 strings; 121 MB) in the corpus after removing the frequency information. The average number of letter trigrams in the strings is 10.3. The total number of trigrams is 301,459. The system generated index files of 601 MB in 551.7s[15].

3. UMLS: This dataset consists of English names and descriptions of biomedical concepts included in the Unified Medical Language System (UMLS). We extracted all
English concept names (5,216,323 strings; 212 MB) from MRCONSO.RRF.aa.gz and MRCONSO.RRF.ab.gz in UMLS Release 2009AA. The average number of letter trigrams in the strings is 43.6. The total number of trigrams is 171,596[14].

4. Synonym set: Set of similar meaning words example talk: {speak, utter, mouth}.

5. Antonym set: A set of conceptually opposite or negated terms for t. For example, pleasant might consist of {“unpleasant,” “unhappy”}.

6. Hypernym set: A set of conceptually more general terms describing. For example (red) might consist of {“color”}.

7. Hyponym Set: A set of conceptually more specific terms describing. For example (red) might consist of {“maroon”, “crimson”}.

B. Building the Reverse Mapping Set

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. The basic architecture of database driven reverse dictionary is as shown in figure 1.
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(clever) will be "intelligent", "bright", "smart", "brilliant". 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 "am", "are", "however" ,"where" etc. needs to be negated as they don’t form a very important part of the process. Whereas, Antonyms are needed to be addressed.
Example: When the word 'clever' is followed by “not”, the antonym of “clever”, which is “stupid” should be considered for the search process.

C. Querying the Reverse mapping

It describes the use of R indexes, to respond to user input phrases. When a user input phrase U is received, first extract the core terms from U. The next step is to apply stemming. Stemming is done in order to convert a term to its base form. For example, if the input phrase given is “hopping animal” and when stemming is applied, ‘hopping’ will get converted to its base form ‘hop’. Stemming is done through a standard stemming algorithm.

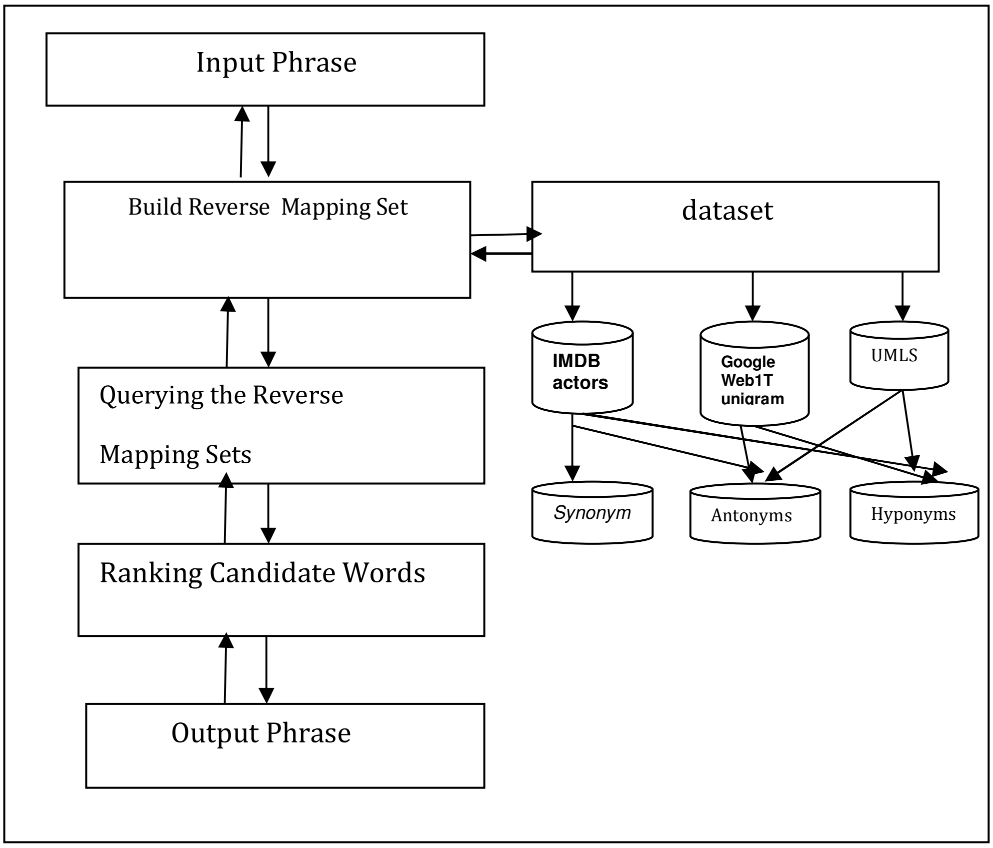

Figure1: Database Driven Reverse Dictionary

After stemming, consult the appropriate R indexes (ie. RMS) of these terms extracted from the user input phrase to find out the candidate words. Given an input phrase " a small town " extract the core terms : “small” and “town” (the term “a” is a stop word and hence it is ignored). Then consult the appropriate R indexes, R (small) and R (town) and will return words in whose definition “small” and “town” occurs simultaneously. Each word becomes a candidate word. A tunable input parameter α is defined which represents the minimum number of candidate words needed to stop processing and return output. If the first step discussed above does not generate a sufficient number of candidate words (W) according to α, then expand the query Q to include synonyms, hypernyms and hyponyms of the terms in Q. When threshold number of candidate words have been found out, then sort the results based on similarity to U and return top β Ws where β is an input parameter representing the maximum number of words to be returned as output.

C. Ranking candidate words

Here semantic similarity of definition of candidate words “S” found is compared with the user input phrase U. On the basis of that, sorts a set of output words in the order of decreasing similarity to U as compared to S. There is a need to assign a similarity measure for each (S,U) pair, where U is the user input phrase and S is the definition of candidate words found out[8].

Here it is necessary to compute both term similarity and term importance. Compute term similarity between two terms based on their location in the WordNet hierarchy. The WordNet hierarchy organizes words in English language from general at root to most specific at leaf nodes. Consider the LCA (Least Common Ancestor) of the two terms and if the LCA of two terms in this hierarchy is root, then those two terms will have little similarity. If the LCA is more deeper, two terms will have greater similarity. Define a similarity function to compute the similarity between two terms ‘a’ and ‘b’

$$\rho(a,b) = \frac{2*E(A(a,b)}{(E(a)+E(b)} \qquad (1)$$

Where “b” is the term in the user input phrase U and “a” is the term in the sense phrase S.A(a,b) return the LCA shared by both a and b in the WordNet hierarchy. E[A(a,b)] is the depth of LCA. E(a) and E(b) return the depth of terms “a” and “b” respectively. Value of ρ(a,b) will be larger for more similar terms. It is essential to consider the importance of each term in the phrase. For Example, Consider two phrases “ the fox who bit the man” and “the man who bit the fox”. These two phrases contain similar words but convey different meanings. So it is important to consider the sequence of words in a phrase. To generate the importance of each term, a parser can be used. OpenNLP[12] parser is used in this work. The parser returns the grammatical structure of a sentence. A parser return a parse tree for a given input phrase. In a parse tree, the terms in the phrase that add most to its meaning appears higher than those words that add less to its meaning[18].

E. Problem and Constraints:

- Problem of approximate dictionary matching.
- It is necessary to compute both term similarity and term importance.
- It does not scale well—for a dictionary containing more than 100,000 defined words, where each word may have multiple definitions; it would require potentially hundreds of thousands of queries to return a result.
- To demonstrate the efficiency of the algorithm on three large-scale datasets with person names, biomedical concept names, and general English words.
- Provide significant improvements in performance scale without sacrificing
- Solution quality but for larger query, it is slow.

III. FUTURE ENHANCEMENT

It is natural to extend this study to compressing and decompressing inverted lists for reducing disk space and for improving query performance .use of K-means clustering algorithm for searching the queries Even though, 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 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. Also we can introducing the wild card characters in user query .Try to make emoticon based dictionary using semantic orientation.

IV. CONCLUSION

Reverse Dictionary Application is a software element that captures a user phrase as input, and returns theoretically connected words as output. The database driven approach can provide significant improvements in performance scale without sacrificing the quality of the result. In this survey paper, we study different approaches that need to construct the database driven reverse dictionary. we describe the significant challenges inherent in building a reverse dictionary, and map the problem to the well-known conceptual similarity problem , the methods for building and querying a reverse dictionary.

V.ACKNOWLEDGMENT

The authors would like to thank for his valuable help throughout their research. They also thank the referees for their suggestions in improving this paper.

REFERENCES

[1] Anindya Datta, Ryan Shaw, Debra VanderMeer and Kaushik Dutta (2013) „Building a Scalable Database-Driven Reverse Dictionary"-VOL. 25, NO. 3, pp.528-540
[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] H. Cui, R. Sun, K. Li, M.-Y. Kan, and T.-S. Chua, "Question Answering Passage Retrieval Using Dependency Relations," Proc. 28th Ann. Int"l ACM SIGIR Conf. Research and Development in Information Retrieval, pp. 400-407, 2005.
[5] T. Dao and T. Simpson, "Measuring Similarity between Sentences," http://opensvn.csie.org/WordNetDotNet/trunk/Projects/Thanh/Paper/WordNetDotNet_Semantic_Similarity.pdf (last accessed 16 Oct. 2009), 2009.
[6] X. Liu and W. Croft, "Passage Retrieval Based on Language Models," Proc. 11th Int"l Conf. Information and Knowledge Management, pp. 375-382, 2002.
[7]F. Sebastiani, "Machine Learning in Automated Text Categorization,"(2002) ACM Computing Surveys, vol. 34, no. 1, pp. 1-47.
[8]Naoaki Okazaki and Jun'ichi Tsujii,"Simple and Efficient Algorithm for Approximate Dictionary Matching",Coling 2010,ICLL,pp 851-859.
[9]Dietterich, "Machine Learning Research" , vol. 3, pp. 993-1022, Mar.2003.
[10]Wang, Wei, Chuan Xiao, Xuemin Lin, and Chengqi Zhang. 2009. Efficient approximate entity extraction with edit distance constraints. In SIGMOD '09: Proceedings of the 35th SIGMOD International Conference on Management of Data, pages 759–770.
[11] Winkler, William E. 1999. The state of record linkage and current research problems. Technical Report R99/04, Statistics of Income Division, Internal Revenue Service Publication.
[12]Li, Chen, Bin Wang, and Xiaochun Yang. 2007.Vgram: improving performance of approximate queries on string collections using variable-length grams. In VLDB '07: Proceedings of the 33rd International Conference on Very Large Data Bases,pages 303–314.
[13] Yuhua Li, David McLean, Zuhair A. Bandar, James D. O'Shea, and Keeley Crockett Sentence Similarity Based on Semantic Nets and Corpus Statistics IEEE transactions on knowledge and data engineering, vol. 18, no. 8, August 2006.
[14] Oscar Méndez, Hiram Calvo, Marco A. Moreno-Armendáriz A Reverse Dictionary Based on Semantic Analysis Using WordNet Advances in Artificial Intelligence and Its Applications Lecture Notes in Computer Science Volume 8265, 2013, pp 275-285 Springer 2013.
[16]M.Porter,"The Porter Stemming Algorithm,"http://tartarus.org/martin/PorterStemmer/, 2009.
SITE References
[17] http://dictionary.reference.com/reverse
[18] http://www.onelook.com/