Construction of Semantic Map of Homophones for Achieving Desired Performance for Searching Homophones for a Given English Word from a Large Database of English Words

Vimal P. Parmar
Research Scholar,
Dept. of Comp. Science
Saurashtra University, Rajkot
Gujarat, India

C.K. Kumbharana, PhD
Head, Guide
Department of Computer Science
Saurashtra University Rajkot
Gujarat, India

ABSTRACT
Semantic map is a complex network of words or phrases which are related in some way. To search related words or phrases for a given English word from a large vocabulary of English language is a time consuming process for a computer. Each time to search related set of words or phrases requires massive processing of comparison and if the entire database is large enough it is required to implement some mechanism that makes searching efficient and fully utilizing the computing power. The words and phrases which are related may be synonyms, antonyms or homophones meaning having similar pronunciation with different spellings and different meanings. The researcher has made an effort to prepare such a network of homophones so that when a search for homophones is required for a given English word, the fast retrieval of result in form of set of homophones can be possible. To determine about the words whether they are homophones or not we require phonetic algorithms for phonetic similarity between words. Similar to indexing a mechanism is derived using an algorithm that is one time processing to prepare a semantic map and then to retrieve set of homophones from this semantic map of homophones. This one time processing for constructing a semantic map is also somehow time consuming processing but once it is constructed searching becomes more efficient.

General Terms
English word database, indexing, File handling

Keywords
Semantic map, Search Operation, Homophone, Linear Search and Binary Search, Algorithms, Array, Time Complexity, Algorithm Efficiency, performance analysis.

1. INTRODUCTION
Searching is fundamental operation in computer science. We always use searching techniques one way or another way to search information from large database. We search specific topic inside a book by searching related keywords from a book by use of the index given at the end of the book, we search the telephone number from telephone directory by searching name arranged alphabetically and we search the English spelling from dictionary. Search operation is involved in our day to day life. Computers are used to process bulk amount of data and searching information from these large database. Almost all computing applications involve searching. Searching homophones for a given word from a large list of English words is not a simple task. Number of phonetic algorithms exists to determine whether the given words have similar pronunciation or not[1][3]. If we want to search homophones from a large list of English words, the given word must be phonetically compared with every word of the large list which is a tedious task and wasting of computing power if we require frequently searching of homophones.

The effort is being made in this research paper to prepare a semantic map of homophone words. The preparation of semantic map network is one time processing which require some amount of time depending on the total number of English words database. The searching becomes more efficient compared to a raw searching of English homophone word by applying the phonetic algorithm with every word. Once the semantic map is prepared it keeps track of every related homophone using the indices of every word. The organization involves accessing of two file. One stores merely the listing of the words and the second with index of each word with indices of every homophone separated by comma.

Whenever a new word is encountered which is not listed in the database, it must be added to the database with its index numbers and all its indices of homophone words and updating all entries to which it is a homophone.

2. INTRODUCTION OF PHONETIC ALGORITHMS
Phonetic algorithms are used to determine the phonetic identity between words. Various algorithms are developed based on needs and languages. Some popular such algorithms are described in following section[3][4].

2.1 Soundex Algorithm
Soundex algorithm was originally developed by Robert C. Russell and Margaret K. Odell in 1918 which returns a four character string for the given word in which the first character is the starting alphabet of the given word and remaining three are digits representing the phonetic encoding[5].

2.2 Daitch-Mokotoff Soundex Algorithm
Daitch-mokotoff soundex is a variation of original soundex and named as D-M soundex which was designed in 1985 by Gary mokotoff and later improved by Randy Daitch to match surnames of Slavic and German languages and returns the six digit numeric code for the given word.

2.3 Kolner Soundex Algorithm

Kolner phonetic algorithm is similar to soundex but was designed for German words.

2.4 Metaphone Algorithms

Metaphone family of algorithms are suitable for most of the English words and these algorithms are the basis for many English spell checkers and dictionaries. First metaphone algorithm was developed by Lawrence Phillips in 1990. Later variation of metaphone by him was double metaphone and incorporating other languages too. In 2009 he released the third version of metaphone which achieves accuracy of 99% of English words[1][3].

2.5 Nysiis Algorithm

NYSIIS means New York state Identification and Intelligence System which is known as NYSIIS phonetic algorithms developed in 1970 which has achieved increased accuracy on soundex.

2.6 Match Rating Approach

The match rating Approach (MRA) is a phonetic algorithms whichwas developed by Western Airlines in 1977 for indexing and comparing homophonous names. MRA uses distance calculation between two words.

2.7 Caverphone Algorithm

The Caverphone phonetic algorithm was developed by David Hood at the University of Otago in New Zealand in 2002 and revised in 2004 and was created for data matching between late 19th century and early 20th century electoral rolls to commonly recognize the names and surnames.

All this algorithms have their own advantages and characteristics. Any algorithm or combination of these algorithms can be used for better accuracy in identifying homophones. The combined effect of these algorithms proves better performance. These algorithms are used to determine the family of homophones and binding them together in form of semantic map which we call in simple words as network of homophones.

3. INTRODUCTION SEMANTIC MAP

Semantic map is a graphical representation of related words. This representation is easier to understand the relationship between words. The semantic map is also known as network or web of related words. Such a map is useful for understanding the concepts, learning and familiar with vocabulary. The word is written in a circle connected with the other words through arcs. Semantic map can be constructed for synonyms, antonyms or phrases which have similar conceptual meaning[9]. Here in this research paper a semantic map of homophones is created from a large list of English words. General semantic map is depicted in following figure 1.

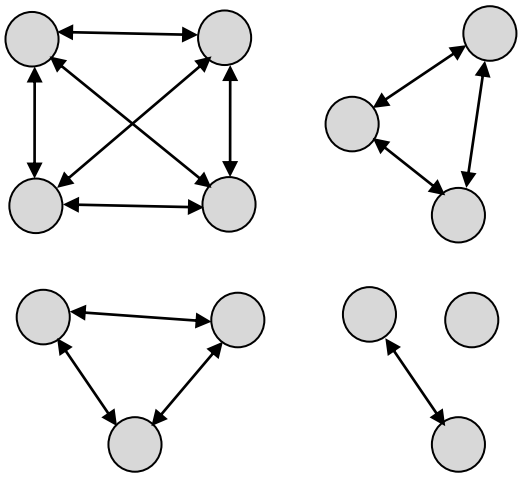

Figure 1: Semantic map representation

The bidirectional arcs joining circles represent the relationship and grouping of similar relationships. The circles or nodes representing the word are bound together. The isolated circle identifies the node which has no relationship. Different semantic map of words can be constructed base on the relationship among the words.

4. THE NEED AND REPRESENTATION STRUCTURE OF HOMOPHONE SEMANTIC MAP NETWORK

If we use three phonetic algorithms for finding set of homophones for a given English word from a large English word list, then each word of the large word list must be phonetically compared with the given word using all three algorithms. The process of searching homophones becomes time consuming and requires more comparisons and wasting of the computing power. For example if we have English word list of 70000 words then the given word must be compared with each of the 70000 words using three algorithms. So to utilize the potential computing power we require some mechanism that makes searching faster and efficient. The mechanism modeled here by the researcher is the semantic map construct which is one time processing for preparing a network of homophone words called knowledge base. Preparation of such semantic map again require amount of time but it is prepared once and can be searched as many number of times. This makes searching efficient. Structure of such semantic map network for homophone words for example for the word “sign” is depicted in following figure 2.

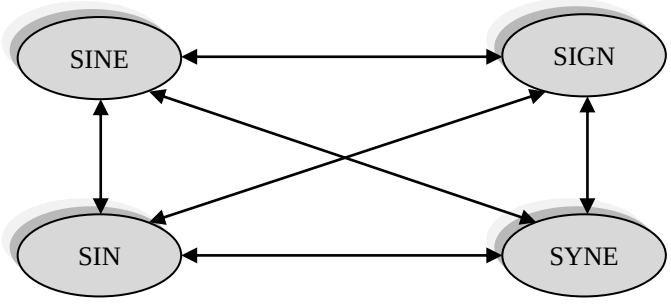

Figure 2 : Semantic map Structure

Each word in above example is related with every other word from the large list of words. There may be more other words of homophone family for the above example. Here the effort is made to create network of all the words in large database. If

no homophone is found for the word it will be treated as isolated word. Once such a complex word network is prepared searching becomes efficient.

5. CONSTRUCTING HOMOPHONE SEMANTIC MAP NETWORK

The model is designed and developed for constructing homophone semantic map network. Word list of nearly 70000 words is taken to form a semantic map network. At beginning of the process first word is taken from the English word list. It then will be compared with all the words in the word database using phonetic algorithm(s) to determine the phonetic similarity. If a match is found then its index or sequence number in database is recorded. Once the comparison process is completed then all the matched indices which were recorded previously are stored in another word list file along with the word and its index as zero. This process is repeated for all the words in the database. The new file that is created is duplicate of the first file but now with an index of each word and the indices of all it homophone words. So the constructed file represents the semantic map which relates words with one another in form of homophone similarity. Using these two file, one for the original merely word list and second with indices makes searching homophone more efficient then the direct searching homophones from original database file which requires to compare every word with given word for determining phonetic similarity. Because when searching a homophone family for the given word, only first match in database is found then the indices in the second file represent the family of the homophones that can directly be searched from the database. This process is more efficient than the direct searching approach.

Following figure 3 represents a schematic of the entire process.

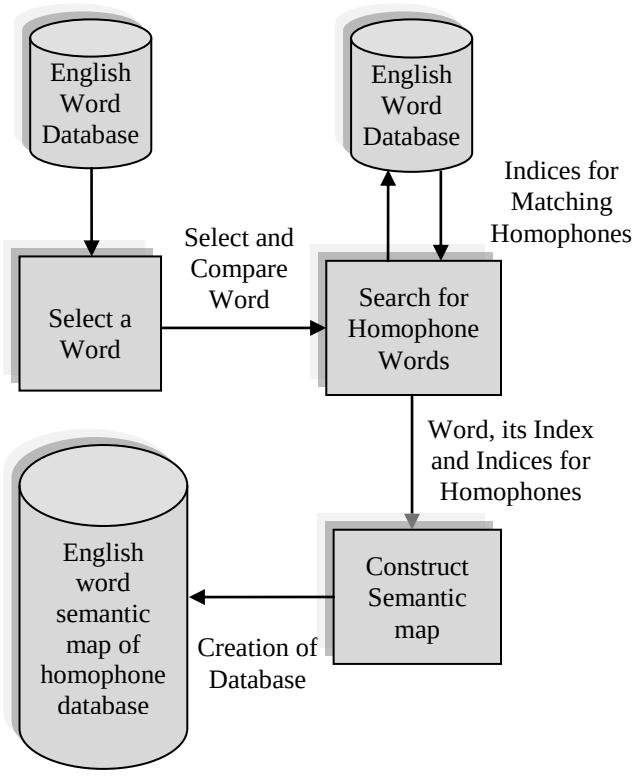

Figure 3: Semantic map Construction Model

From the above figure, first word is taken from large database of English words. Then it is phonetically compared with all the words in database and the indices of that words are recorded. Once the first word is compared with all the words in database than that word with it index and all the indices of its homophones are used to construct the semantic map. Each record consists of two types of indices, one is the index of itself and other is a set of indices of its homophones. The entire record of index number of the word, word itself and comma separated indices of its homophone is stored again in new file of database. Thus the homophones indices for first word are recorded but the words at all the found indices have the same homophone family including the first one.

So all the records at that indices must have the homophone word indices including first word. Each record must have the homophone indices except its own index because a word is homophone of itself and it is not necessary to store the index of itself as homophone index.

If we assume fetching the first word from database, comparing it with all other words and storing back record to the database require nearly one second of processing than to prepare a semantic map of entire database it requires 70000 second for a database consisting of 70000 words. It means it requires almost $70000/3600 \approx 19.44$ hrs of computer processing. The process to prepare a semantic map is time consuming process but it is one time processing and optimization can be applied in preparation process.

Once the semantic map is prepared searching becomes efficient as comparisons are reduced and we have a set of homophone indices through which direct search can be performed. Compare to time require to search it just requires few space to create a same database file with indices.

6. HOMOPHONE SEMANTIC MAP FILE STRUCTURE

The file structure of semantic map consists two types of file one with raw English words and second with the indices. General structure for first file is raw English words. Each record merely consists of English word. Using this file an index file is constructed whose file structure is described as follows.

Index_N Word_N Index_N1, Index_N2, Index_N3……

Index_N1 Word_N1 Index_N, Index_N2, Index_N3……

Index_N2 Word_N2 Index_N, Index_N1,Index_N3 … …

Index_N3 Word_N3 Index_N, Index_N1,Index_N2 … …

… …

Index_N is the index number of the word, Word_N is the spelling as taken from the first file and located atindex Index_N and Index_N1, Index_N2, Index_N3 … … are the indices of all the homophones of the word Word_N.

For example the file with sample data set of few records of English words having content listed as follows.

The indices listed in the following example may vary depending on the original database file of English words. The shown English words are taken as homophones and 11th entry has no homophone found so it does not contain homophone indices. Each listed entry must be processed to search homophones. To reduce the processing and optimizing the algorithm, homophone indices of every found homophone must be updated so that it is not necessary to search the same homophone family searched previously.

0.	Cash 1
1.	Case 0
2.	Four 3
3.	For 2
4.	Sell 5, 6
5.	Cell 4, 6
6.	Sail 4, 5
7.	Leave 8
8.	Live 7
9.	Knight 10
10.	Night 9
11.	Psychology

Figure 4: Semantic map File Structure Example

7. ALGORITHM FOR CONSTRUCTING SEMANTIC MAP HOMOPHONE WORD NETWORK

Algorithm processes by initiating connection with first database file for reading all the words from word list. It also creates a second database file to store the words along with the indices.

Pick up one word from the first file, its index is zero and compare it with all the words in the same file for phonetic equality. If match is found its index is recorded and added that index at the end of index of first word followed by word itself in second file.

Also to improve the algorithm efficiency all the matched indices words entry in second file are updated with all the indices except its own index, as a word is homophone of itself so no need to record its own index. Same process is repeated for the second word, third word and so on for all the words in database.

Algorithm is optimized such that if homophone is found say at Nth entry, then the same searching process for the word at index N is not repeated, because homophone family is already found for the entry at N index and is already updated. The resultant second file represents the homophone semantic map in form of a network of related words and using this map searching becomes more efficient.

Algorithm with optimization of reducing processing for searching homophones is listed as follows.

Step 1 : Start for preparing, connecting and fetching the original word list from the database file and preparing an index file which contains the index followed by the word followed by the homophone indices. Count the total number of words say N in the original word list file to process all the words.

Step 2 : Repeat for i = 0 to N - 1.

Step 3 : Repeat for j = 0 to N - 1.

Step 4 : Check whether the word(i) is already being processed?

If yes then

Move to step 2 by updating value of i.

Else

Compare for homophone word(i) with word(j) where i does not equal to j.

If homophone word(j) found for word(i) then

Update the entry in index file by appending the index j at the word located at index i. Keep track of index j to update its entry in index file later using an index array say ind[ ] and total homophones found currently as TOTAL.

else

Update the value of j to fetch the next word for comparison.

Step 5 : Once all the words are compared, inner loop with index j is terminated.

Now update all the entries found as homophone whose indices are stored in an index array ind[ ] using the following procedure and TOTAL keeps track of total number of homophones found for word(i).

Repeat for k = 0 to TOTAL – 1

Update the entries of words indexed at ind[ ] by appending all the indices stored in ind[0] to ind[TOTAL-1] except its own index found in ind[ ] because it is not required to store index of its own as homophone index. Obviously the word is itself a homophone of itself.

Update the value of i to search homophones for the next word and repeat the processing from the step 3.

Step - 6 : After termination of outer loop indexed by i, close both the database files. The wordlist file with the indices represents the group of homophones in form of semantic map bound together through indices.

Step – 7 : Finished.

8. CONCLUSION

The algortihm can be implemented using any programming language supporiting database and file handling capabilities. Implemetation for preparing a semantic map as per algorithm requires hours of time due to large database of words and large number of comparisons to find set of homophones, but it is just a one time processing. Once it is created searching a family of homophones becomes more efficient using both of the files compared to direct searching from large database of words.

Further, if new word is added in a list then same process of comparison with all the words in first file is repeated and accordingly if first match is found than all the homophone indices areappended at the end of new word and all the homophone index entries are updated by appending the index of new word. So that the semantic map becomes consistent for new word also.

It possible to construct such semantic map for different kind of relationship between words and can be useful for other applications. Many application have adopted this technique to relate contextual relationship between word. Thus using this technique a database is shaped in a knowledge base that answers based on the related terms. Number of such possibilities exist depending on what kind of relationship is found. At last it is still possible to optomize the described algorithm in form of its processing to achieve the same result.

9. REFERENCES

[1] Analysis and Comparative Study on Phonetic Matching Techniques Rima Shah, Dheeraj Kumar Singh International Journal of Computer Applications (0975 – 8887) Volume 87 – No.9, February 2014

[2] Name and Address matching strategy – White Paper Series Truth Technologies December 2010

[3] Vimal P. Parmar, Dr. CK Kumbharana "Study Existing Various Phonetic Algorithms and Designing and Development of a working model for the New Developed Algorithm and Comparison by implementing it with Existing Algorithm(s)" International Journal of Computer Applications (IJCA) ISSN: 0975 – 8887 Volume 98 / Number 19 (ISBN: 973-93-80883-19-1) DOI : 10.5120/17295-7795

[4] Vimal P. Parmar, Apurva K. Pandya, Dr. CK Kumbharana "Determining the Character Replacement Rules and Implementing Them for Phonetic Identification of Given Words to Identify Similar Pronunciation Words" Futuristic Trends on Computation Analysis and Knowledge Management (ABLAZE) 2015 International Conference at Greater Noida, India Pages : 272-277 Print ISBN : 978-1-4799-8432-9 DOI : 10.1109/ABLAZE.2015.7155010 Publisher : IEEE

[5] Binary Search Tree Balancing Methods: A Critical Study Suri Pushpa1, Prasad Vinod IJCSNS International Journal of Computer Science and Network Security, VOL.7 No.8, August 2007

[6] The Semantic Map of the Spatial Domain and Related Functions- Wei Wang University of California, Los Angeles- Language and Linguistics 16(3) 465–500 DOI: 10.1177/1606822X15569169

[7] The representation of homophones: Evidence from remediation- Britta Biedermann, Gerhard Blanken, Lyndsey Nickels- APHASIOLOGY, 2002, *16* (10/11), 1115–1136

[8] The homophone effect during visual word recognition in children: an fMRI study- Sharlene D. Newman, Psychological Research DOI 10.1007/s00426-011-0347-2

[9] Semantic relationship - www.iva.dk/bh/lifeboat_ko/CONCEPTS/semantic_relations.htm