# 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)

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

CK Kumbharana, Ph.D
Head, Guide
Department of Computer Science
Saurashtra University Rajkot, India

## ABSTRACT
A phonetic algorithm is an algorithm to identify words with similar pronounce and is used to index the words based on their pronunciation. Most of the algorithms are designed to work with English language. These algorithms are complex by nature due to many rules and exceptions in English pronunciation and change in evolving English language with adoption of many words from other languages. Also there are many differences between UK English and US English. Although due to many such circumferences there are many algorithms with different rules for identifying similar pronunciations words. In the presented research paper, the researcher has studied the different algorithm related to phonetics and developed a new algorithm and compared it with the existing algorithms.

## General Terms
Pronunciation, phonetic algorithm

## Keywords
Soundex, Metaphone, Homophone, Matching, text patterns.

## 1. INTRODUCTION
Automatic determination of similar pronunciation words is the crucial mechanism for the computer world. In this area numbers of algorithms are developed to solve the pronunciation problems. But by studying the different algorithms they may fails in some circumstances. The prime goal of the researcher is to study and identify some circumstances where the algorithm fails, So by taking different nearer 25 similar words, the researcher has implemented the algorithm and identified the problems and depending upon the deficiency I have designed and developed a working model and algorithm to identify similar phonetic words. And also the comparison is also performed based on the sample data set and concluded the analysis of the algorithms. I have studied various working algorithm which are as follows.

## 2. STUDY OF EXISTING ALGORITHMS

### 2.1 Soundex
This algorithm was originally developed Robert C. Russell and Margaret K. Odell in 1918[2]. It returns a four character string for the given word. The first character represents the starting alphabet of the inputted word and remaining three are digits depending upon the phonetic characters.

### 2.2 Daitch-mokotoff soundex
A variation of soundex D-M soundex 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[1,6].

### 2.3 Kolner phonetic
This algorithm is similar to soundex but designed for German words[1].

### 2.4 Metaphone, Double metaphone and Metaphone 3
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. This series of metaphone algorithms are suitable for most of the English words and these algorithms are the basis for many English spell checkers and dictionaries.

The working Mechanism of these algorithms is described below. [3]

#### *2.4.1 Metaphone Algorithms*
Original Metaphone codes use the 16 consonant symbols 0BFHJKLMNPRSTWXY. The '0' represents "th" (as an ASCII approximation of Θ), 'X' represents "sh" or "ch", and the others represent their usual English pronunciations. The vowels AEIOU are also used, but only at the beginning of the code. Original Metaphone contained many errors and was superseded by Double Metaphone, and in turn Double Metaphone and original Metaphone were superseded by Metaphone 3, which corrects thousands of miscoding that were be produced by the first two versions[3].

#### *2.4.2 Double Metaphone:*
The Double Metaphone phonetic encoding algorithm is the second generation of this algorithm. Its implementation was described in the June 2000 issue of C/C++ Users Journal. It makes a number of fundamental design improvements over the original Metaphone algorithm[3].

It is called "Double" because it can return both a primary and a secondary code for a string; this accounts for some ambiguous cases as well as for multiple variants of surnames with common ancestry. Double Metaphone tries to account for myriad irregularities in English of Slavic, Germanic, Celtic, Greek, French, Italian, Spanish, Chinese, and other

origin. Thus it uses a much more complex rule set for coding than its predecessor; for example, it tests for approximately 100 different contexts of the use of the letter C alone[3].

### *2.4.3 Metaphone 3:*

A professional version was released in October 2009, developed by the same author, Lawrence Philips[3]. It is a commercial product but is sold as source code. Metaphone 3 further improves phonetic encoding of words in the English language, non-English words familiar to Americans, and first names and family names commonly found in the United States.

All these algorithms were basically built to find out similar pronunciations names and surname stored in large databases but here an attempt is made to find out similar English words.

## 2.5 NYSIIS

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 developed by Western Airlines in 1977 for indexing and comparing homophonous names[1].

## 2.7 Caverphone

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 in data matching between late 19th century and early 20th century electoral rolls to commonly recognize the names[1].

By studying the above algorithms I have found the soundex (2.1) and metaphone (2.4) are widely used in the different applications. So, I have selected, these two algorithms and implemented them on the selected various words to find out the problems. By identifying the outcome I have developed the new algorithm and the comparison table is given as in Table 1.

# 3. VARIOUS APPLICATIONS OF THE PHONETIC ALGORITHMS

3.1 Application of these algorithms can be incorporated into speech recognition system to identify the correct word from similar phonetic words. Although it is somehow difficult but can be achieved at some extent based on the context in which it is used.

3.2 In spelling correction to produce more than one correct words having similar pronounce.

3.3 Search applications can provide the set of related search terms when spell-mistake words are given to it by finding similar coded words.

3.4 Helpful for children to increase vocabulary of English words and to learn homophones having similar pronounce with different meaning.

3.5 Can be used to search and modify names from large databases with similar pronunciations.

3.6 These phonetic algorithms are also incorporated in Sql, Mysql, Oracle and Informatica like databases as well as PHP scripting. In database it is possible to search similar pronunciation data with different spells.

3.7 These types of algorithms can be used in one word input online based computerized test to assist the examinee when user just knows the phonetics but not the actual spelling of answer.

3.8 Many more applications can be possible in areas of pharmaceutical, trademark searching and oral securities.[5]

3.9 To build intelligent dictionary.

# 4. PROPOSED ALGORITHM FOR PHONETIC IDENTIFICATION OF ENGLISH WORDS

The proposed algorithm is designed and developed is basic algorithm and it is not comprehensive and complete. It requires modification to cover many of the English pronunciation rules. This algorithm is designed after studying soundex, metaphone and English pronunciation rules. Many exceptional cases of similar spellings are exist and for those other rules should be incorporated to enhance the algorithm. Most of the phonetic algorithms are designed to search names and surnames stored in database but the presented algorithm is designed to search actual English word, which has similar pronunciation but different meaning.
Following is the algorithm to determine the phonetic code of the given English text. Schematic diagram of algorithm is depicted in Figure – 1.

1. Start
2. Input English text or word
3. Convert the given English text into capital only to simplify the process.
4. Remove the repetitive subsequence characters to obtain the target text which contains only a single character by comparing each pair of subsequence characters. For example the given text "LETTER" will result in "LETER" after applying this process.
5. This process identifies the phonetic similarity between two or more words. It consists a set of English pronunciation rules after applying them we obtain the encoded English text that represents the phonetic code for the given text. If two words have identical phonetic code then we can interpret as phonetic similarity otherwise not. Encoding of text after applying rules can be made as per the programmers' choice. But here it is used as simple as metaphone algorithm returns the code. Following is the set of rules that are used to apply on text. Although the listing given below is not complete and comprehensive but it can be extended by adding more English pronunciation rules with some exception cases too. All the processes are independent here so as to extending the phonetic rules database does not impact on overall algorithm design.
   - 5.1 Replace each occurrence of CE, CI , CY → S
   - 5.2 Replace each occurrence of GE, GI, GY → J
   - 5.3 Replace each occurrence of WR → R
   - 5.4 Replace each occurrence of GN, KN, PN → N
   - 5.5 Replace each occurrence of CK → K
   - 5.6 Replace each occurrence of DGE → J

5.7 Replace each occurrence of OUL → U

5.8 Replace each occurrence of OUGH → F

5.9 Replace each occurrence of SH → S

5.10 Replace each occurrence of GHT → T

6. Remove all the vowels from the resultant target text except if it is the first character of the given text.
7. Display the encoded text representing phonetic code of the given text.
8. Finished.

By applying the above algorithm repetitively for the different words different output is obtained.

## 5. COMPARING RESULTS OF SOUNDEX, METAPHONE AND PROPOSED ALGORITHM WITH SELECTED DATA SET

Comparison of all these algorithms involves time analysis, space analysis, performance issues, accuracy and implementation. Time and space complexity are not the major issues even performance due to the advances in programming language and hardware enhancements. For such phonetic algorithm accuracy is the critical factor for the correctness of the algorithm, why because English language is too large with complex spelling structure having different meaning with similar pronunciation. Here to check the accuracy of algorithms, a selected proper data set is used as in table – 1.

Table – 1 demonstrates the output of three algorithms soundex, metaphone and newly developed algorithm with sample data set.

First column represents serial number, second column represents the sample test words, third column is the outcome of soundex algorithm, forth column is remarked as success or failure for soundex, fifth column is the outcome of metaphone algorithm, sixth column is remarked as success or failure for metaphone, seventh column is the outcome of proposed algorithm and eighth column is remarked as success or failure of the proposed algorithm.

By testing the result we can easily identify whether the words have identity or not. In Table – 1 similar words are remarked with "✓" and dissimilar words are remarked with "×" and highlighted background where the algorithm fails. We can compare such algorithm with basic markov algorithm of string substitution which is widely used in theory of computation. Also compiler uses such mechanism to convert string of text that is source program into string of binary code that results into a machine language code.

The success of the algorithm relies on whether the inputted names are truly identified or not.
From table – 1 it is observed that the words with serial number 1, 2, 3, 5, 6, 7, 8, 11, 12, 16, 18, 19, 21, 23, 24, 25 are identified correctly by all the three algorithms. But highlighted words in table – 1 are not identified correctly by these algorithms. Non-identified words by different algorithms in table – 1 are summarized in table – 2

**Table : 2 Summary of non identified words**

| Sr. no | Success (✓) / Failure (×) of the algorithm | | |
|---|---|---|---|
| | Soundex algorithm | Metaphone algorithm | Proposed algorithm |
| 4 | × | × | ✓ |
| 9 | ✓ | × | ✓ |
| 10 | ✓ | × | ✓ |
| 13 | × | ✓ | ✓ |
| 14 | × | ✓ | ✓ |
| 15 | ✓ | × | ✓ |
| 17 | ✓ | × | ✓ |
| 20 | ✓ | × | ✓ |
| 22 | ✓ | × | × |

Using soundex algorithm, both "piece" and "peace" (sr. no 2 in table – 1) return the same string "P200" that proves the success of the algorithm, while "would" (sr. no 4 in table – 1) yields "W430" and "wood" results in "W300" that proves the failure of algorithm. Using metaphone algorithm, "would" and "wood" (sr. no 4 in table – 1) returns "WLT" and "WT", that proves the failure of the algorithm. But proposed new algorithm results both as "WD", means the algorithm is success.

From table – 2 it is observed that the soundex algorithm fails three times, metaphone algorithms fails seven times where as proposed algorithm fails only once. Hence, the proposed algorithm is efficient compared to soundex and metaphone algorithms. It is also observed that for the words with sr. no. 22, only soundex proves success and for the words with sr. no 4, only proposed algorithms proves success.

**Table 1 : Experimenting algorithms with sample data set**

| No. | Word | Soundex Return | Rem | Metaphone Return | Rem | Proposed algorithm Return | Rem |
|---|---|---|---|---|---|---|---|
| 1. | Week | W200 | ✓ | WK | ✓ | WK | ✓ |
| | Weak | W200 | ✓ | WK | ✓ | WK | ✓ |
| 2. | Piece | P200 | ✓ | PS | ✓ | PS | ✓ |
| | Peace | P200 | ✓ | PS | ✓ | PS | ✓ |
| 3. | Bed | B300 | ✓ | BT | ✓ | BD | ✓ |
| | Bad | B300 | ✓ | BT | ✓ | BD | ✓ |
| 4. | Would | W430 | × | WLT | × | WD | ✓ |
| | Wood | W300 | × | WT | × | WD | ✓ |
| 5. | Sun | S500 | ✓ | SN | ✓ | SN | ✓ |
| | Son | S500 | ✓ | SN | ✓ | SN | ✓ |
| 6. | Ship | S100 | ✓ | XP | ✓ | SP | ✓ |
| | Sheep | S100 | ✓ | XP | ✓ | SP | ✓ |
| 7. | Later | L360 | ✓ | LTR | ✓ | LTR | ✓ |
| | Letter | L360 | ✓ | LTR | ✓ | LTR | ✓ |
| 8. | Low | L000 | ✓ | L | ✓ | LW | ✓ |
| | Law | L000 | ✓ | L | ✓ | LW | ✓ |
| 9. | She | S000 | ✓ | X | × | S | ✓ |
| | See | S000 | ✓ | S | × | S | ✓ |
| | Sea | S000 | ✓ | X | × | S | ✓ |
| 10. | Case | C200 | ✓ | KS | × | CS | ✓ |
| | Cash | C200 | ✓ | KX | × | CS | ✓ |
| 11. | Of | O100 | ✓ | OF | ✓ | OF | ✓ |
| | Off | O100 | ✓ | OF | ✓ | OF | ✓ |
| 12. | Live | L100 | ✓ | LF | ✓ | LV | ✓ |
| | Leave | L100 | ✓ | LF | ✓ | LV | ✓ |
| 13. | Sign | S250 | × | SN | ✓ | SN | ✓ |
| | Sine | S500 | × | SN | ✓ | SN | ✓ |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 14. | Sin | S250 | × | SN | ✓ | SN | ✓ |
| | Seen | S500 | × | SN | ✓ | SN | ✓ |
| 15. | By | B000 | ✓ | B | × | B | ✓ |
| | Bye | B000 | ✓ | BY | × | B | ✓ |
| 16. | Reach | R200 | ✓ | RX | ✓ | RCH | ✓ |
| | Rich | R200 | ✓ | RX | ✓ | RCH | ✓ |
| 17. | Sort | S630 | ✓ | SRT | × | SRT | ✓ |
| | Short | S630 | ✓ | XRT | × | SRT | ✓ |
| 18. | Center | C536 | ✓ | SNTR | ✓ | SNTR | ✓ |
| | Centre | C536 | ✓ | SNTR | ✓ | SNTR | ✓ |
| 19. | Full | F400 | ✓ | FL | ✓ | FL | ✓ |
| | Fool | F400 | ✓ | FL | ✓ | FL | ✓ |
| 20. | Then | T500 | ✓ | ON | × | THN | ✓ |
| | Than | T500 | ✓ | XN | × | THN | ✓ |
| 21. | Fill | F400 | ✓ | FL | ✓ | FL | ✓ |
| | Feel | F400 | ✓ | FL | ✓ | FL | ✓ |
| 22. | Two | T000 | ✓ | TW | × | TW | × |
| | To | T000 | ✓ | T | × | T | × |
| | Too | T000 | ✓ | T | × | T | × |
| 23. | Four | F600 | ✓ | FR | ✓ | FR | ✓ |
| | For | F600 | ✓ | FR | ✓ | FR | ✓ |
| 24. | Mat | M300 | ✓ | MT | ✓ | MT | ✓ |
| | Met | M300 | ✓ | MT | ✓ | MT | ✓ |
| 25. | Merry | M600 | ✓ | MR | ✓ | MR | ✓ |
| | Marry | M600 | ✓ | MR | ✓ | MR | ✓ |

## 6. FUTURE ENHANCEMENT

It is possible to combine more than two algorithms to derive the hybrid algorithm for better result enhancement. It is also possible using artificial intelligent that can be incorporated in such algorithms to obtain more powerful systems and as human being is able to identify and recognize such homophone spelling, a machine with artificial neural network and fuzzy logic can also be able to identify and recognize such spelling.

## 7. CONCLUSION

Table – 3 represents the total number of success and fail cases with percentage of success for all three algorithms bases on the sample data set of 25 words experimented as in given Table – 1. From table – 3 it is concluded that soundex algorithm proves 88% of success, metaphone algorithm proves 72% of success where as proposed algorithm proves 96% of success.

Table – 4 represents the comparison of all the three algorithms with each other including comparison with itself using the success ratio. Success ratio for each algorithm is calculated by taking division of success percent of one algorithm by success percent of another. Ratio obtained greater than one indicates good performance over another where as less than one indicates inefficiency over another and success ratio equal to one indicates equal performance.

It is concluded that soundex compared with itself yields success ratio of 1.000 that means proves equal performance which is obvious but when compared with metaphone yields ratio 1.222 that implies how much it performs better than metaphone. When soundex is compared with proposed new algorithm results in ratio 0.917 and that implies the deficiency over new algorithm.

Similarly success ratio of metaphone when compared with soundex yields 0.818 that proves the deficiency over soundex. Metaphone when compared with itself results in equal efficiency with success ratio 1.000 but when compared with the new algorithm proves deficient with success ratio of 0.750.

The newly proposed algorithm when compared with both soundex and metaphone algorithms it proves itself efficient with the success ratio of 1.091 and 1.333 respectively. Similar to soundex and metaphone algorithms when the new algorithm is compared with itself results in the success ratio of 1.000.

So, from the table – 4 it is concluded that the newly proposed algorithm proves as successful algorithm with comparing to soundex and metaphone algorithms.

The working model of the proposed phonetic algorithm with each necessary operation is given in Figure – 1.

**Table - 3 : Conclusion of comparing three algorithms based on sample data set of Table - 1**

| No. of Words | Soundex Algorithm | | | Metaphone Algorithm | | | New Phonetic Algorithm | | |
|---|---|---|---|---|---|---|---|---|---|
| | Success | Fail | Percent Success | Success | Fail | Percent Success | Success | Fail | Percent Success |
| 25 | 22 | 3 | 88% | 18 | 7 | 72% | 24 | 1 | 96% |

**Table - 4 : Success ratio of algorithms**

| Comparison of algorithms based on sample data set | With Compared to Algorithm | | |
|---|---|---|---|
| **Success Ratio of Algorithms** | Soundex | Metaphone | New Algorithm |
| Soundex Algorithm | 1.000 | 1.222 | 0.917 |
| Metaphone Algorithm | 0.818 | 1.000 | 0.750 |
| New Algorithm | 1.091 | 1.333 | 1.000 |

Figure 1 represents the design schematic of working model of developed new algorithm

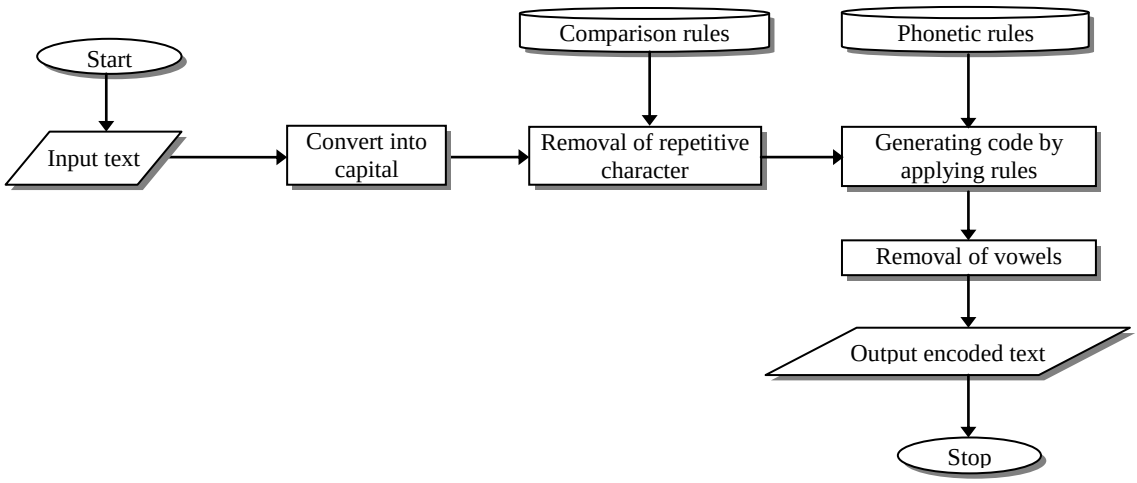


**Figure - 1: Working model of new derived phonetic algorithm**

## 8. REFERENCES

[1] Phonetic algorithm meaning and description, http://en.wikipedia.org/wiki/Phonetic_algorithm

[2] Soundex algorithm description and working mechanism, http://en.wikipedia.org/wiki/Soundex

[3] Metaphone algorithm description and working mechanism, http://en.wikipedia.org/wiki/Metaphone

[4] Chakkrit Snae, A Comparison and Analysis of Name Matching Algorithms, World Academy of Engineering and Technology 1, 2007.

[5] Phonetic Comparison Algorithms By BRETT KESSLER Washington University in St. Louis Volume 103:2 (2005) 243–260

[6] Analysis and Comparative Study on Phonetic Matching Techniques, Rima Shah, Dheeraj Kumar Singh, IJCA Volume 87 – No.9, February 2014

[7] Name and address matching strategy – White Paper December 2010