PHONETIC MATCHING TOOLKIT WITH STATE-OF-THE-ART META-SOUNDEX ALGORITHM (ENGLISH AND SPANISH)

_____

A Thesis

Presented to

The Faculty of the Department of Computer Science

Sam Houston State University

_____

In Partial Fulfillment

of the Requirements for the Degree of

Master of Science

_____

by

Keerthi Koneru

December, 2016

PHONETIC MATCHING TOOLKIT WITH STATE-OF-THE-ART META-SOUNDEX ALGORITHM (ENGLISH AND SPANISH)

by

Keerthi Koneru

_____

APPROVED:

Cihan Varol, PhD
Thesis Director

Narasimha Shashidhar, PhD
Committee Member

Bing Zhou, PhD
Committee Member

John B. Pascarella, PhD
Dean, College of Science and Engineering Technology

DEDICATION

I dedicate this dissertation to all my family members who had always motivated and supported me to successfully achieve this. A special dedication to my father, Koneru Paramesh Babu, whom I regard as the most influential person in my life, who always showed me the right direction and responsible for grooming me into a better individual. Last but not the least, I would like to dedicate to my brother Sridhar Reddy and my friend Subash Kumar, who constantly motivated me to work comprehensively and effectively, which helped me a lot in successfully achieving this milestone.

ABSTRACT

Koneru, Keerthi, *Phonetic matching toolkit with state-of-the-art Meta-Soundex algorithm (English and Spanish)*. Master of Science (Computing and Information Science), December, 2016, Sam Houston State University, Huntsville, Texas.

Researchers confront major problems while searching for various kinds of data in large imprecise databases, as they are not spelled correctly or in the way they were expected to be spelled. As a result, they cannot find the word they sought. Over the years of struggle, pronunciation of words was considered to be one of the practices to solve the problem effectively. The technique used to acquire words based on sounds is known as "Phonetic Matching". Soundex was the first algorithm developed and other algorithms like Metaphone, Caverphone, DMetaphone, Phonex etc., are also used for information retrieval in different environments. This project mainly deals with the analysis and implementation of newly proposed Meta-Soundex algorithm for English and Spanish languages which retrieves suggestions for the misspelled words.

The newly developed Meta-Soundex algorithm addresses the limitations of Metaphone and Soundex algorithms. Specifically, the new algorithm has more accuracy compared to both Soundex and Metaphone algorithm. The new algorithm also has higher precision compared to Soundex, thus reducing the noise in the considered arena.

A phonetic matching toolkit is also developed enclosing the different phonetic matching algorithms along with the state-of-the-art Meta-Soundex algorithm for both Spanish and English languages

KEY WORDS: Caverphone, DMetaphone, Information retrieval, Misspelled words, Metaphone, NYSIIS, Phonetic matching, Soundex

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my advisor Dr. Cihan Varol, for the continuous support and advices for my research. His patience, motivation and knowledge helped me to articulate my project in a better shape. Through this journey, he made me strong technically and helped me to use different tools and get exposed to different work environments.

Besides my advisor, I would like to thank the committee, Dr. Narasimha Shashidhar and Dr. Bing Zhou, for their insightful comments, valuable suggestions, and encouragement for the improvement of project in different perspectives. I would also like to thank Computer Science Department for providing the facility for my research. Finally, I would like to thank all the professors, who transformed me to a responsible student by providing their valuable feedback, which made the whole process an exciting experience.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER I

Introduction

Information deterioration is an intensive problem for every organization in the present era. With the increase in the amount of information saved day by day, there is desperate need for locating the mistyped data. Organizations are facing great challenge to maintain the quality of data in information systems with various sources of data damage. Whenever the data is assimilated from multiple sources, it is a challenge to recognize the duplicate records due to the existence of misspelled data for the same record. As a result, the information of organization always ends up at risk.

Databases play a crucial role in almost all the establishments. Matching records in database is a persistent and a well-known problem for years. Data matching process mainly involves comparison of records to ascertain whether they are same entity or not. While retrieving information, the major role includes ranking the set of data that is most likely to be similar. One of the techniques to improve the data retrieval process involving variations in sound, which detect the misspelled data, is Phonetic matching.

String Matching Vs Phonetic Matching

String Matching is the technique of matching the approximate pattern of strings by diving the string into substrings. It mainly involves insertion, deletion, and substitution of letters to find the near matches (SaiKrishna et.al., 2012, Singla et.al., 2012). Phonetic comparison meticulously obtains the quantitative analysis of pronunciations between speech forms and spellings of words. It involves identification of words that are most likely to sound similar. The comparison of String Matching and Phonetic Matching is described in Table 1.

Table 1

String Matching Vs Phonetic Matching

	String Matching	Phonetic Matching
Matching	Matches data based on patterns of substrings	Matches data based on the similar pronunciations
Involves	Addition, Deletion or Substitution of Letters	Conversion of data to phonetic patterns.
Applications	Applied in Search Engines, Bio-Informatics, spell checkers, digital forensics etc.	Mainly used in name retrieval in enquiry lines, record linkage and fraud detection. Gaining its importance in spell checkers and; search engines.
Prominence	Mainly used for matching names and nouns from English Language.	Can be used in multi-lingual environment, where diversities in pronunciation or writing styles may be present.

Phonetic Matching Algorithms

The evolution of Phonetic matching has come into frame when there is a hardship in the retrieval of information (Beider et.al., 2010). Phonetic matching algorithm involves indexing of words centered on phonation. The algorithms comprise of many rules and exceptions as spellings and pronunciations in English and Spanish are complicated and include historical changes having the words borrowed from many languages (Phonetic Matching, 2009). The technique of obtaining words using sounds was used in the US census since the late 1890's, but a concrete solution to this was first proposed and

patented by Robert C. Russell in 1912 as Soundex algorithm (Shah et.al., 2014). Later, many algorithms were developed based on the different specifications and language constraints. Some of the other prominently used algorithms are Metaphone, Daitch-Mokotoff Soundex, NYSIIS, DMetaphone, Caverphone, Phonix etc.

Meta-Soundex Algorithm

In spite of many phonetic matching algorithms, there is still a need to develop a concrete algorithm to achieve higher data quality as each and every algorithm has its own disadvantages (Shah et.al., 2014). Soundex is one of the prominent algorithms having high accuracy but it has very low precision due to the large overhead. Metaphone is a well-known phonetic matching algorithm comprising of rules involving vowels and sounds of diphthongs but has less accuracy. To overcome such shortcomings, a new algorithm is proposed, where the encoding process includes both the vowel and diphthong sounds. As these sounds are reflected, the number of false positives are reduced, thus reducing the overhead occurred by them.

CHAPTER II

Related Work

Sources of Variations in Data

Information Retrieval is one of the major viewpoints of data mining application areas (Singh et.al., 2014). However, the information may not be consistent over the considered arena due to various causes. The different sources of variations can be illustrated as:

Spelling Variations. These mainly occur due to typographical errors, substituted letters or by addition or omission of letters.

Phonetic Variations. These are caused when the phonetic structure of words is modified due to mishearing.

Double names or Double first names. It occurs when the names contain more than one word and all of them are not mentioned consistently in data.

Change of Name. In the course of time, if an individual undergoes change of name, which might not be updated in all the places of existing data (Shah et.al., 2014).

Of the different criteria mentioned above, the research in phonetic variations led to the development of phonetic matching algorithms which obtains worthwhile approximate matches to misspelled words.

Evolution of Phonetic Matching Algorithms

The main goal of phonetic matching algorithms is to encode homophones to the same representation so that they can be matched despite minor differences in spelling (Stephen Haunts, 2014). The background of various phonetic matching algorithms is discussed here and the details of these algorithms are given in Chapter III.

Soundex. The earliest algorithm in the literature is Soundex developed by Odell and Robert C. Russell in 1912, which produces a four-digit code retaining its first letter. The algorithm is patented by the authors in 1918 (Odell et.al., 1918). The process mainly encodes consonants and a vowel is not encoded unless it is the first letter. Arguably, Soundex is one of the most widely known of all phonetic algorithms. It is used as a standard feature in applications like mySQL, oracle, etc. Because of the few disadvantages like dependency on the first letter, failure of detection of silent consonants, Soundex can only be used in applications where high false positives and false negatives can be tolerated (Shah et.al., 2014).

Beider-Morse Phonetic Matching. An improvement of Soundex is implemented by Beider and Morse to reduce the number of false positives and false negatives, known as Beider-Morse Phonetic Matching (BMPM). Beider et.al., has also mentioned that the algorithm is extended to languages other than English, with the application of some generic rules to obtain the phonetic codes (Beider et.al., 2010). Varol et.al., discussed BMPM as a hybrid technique with a 6-letter encoded code in which the percentage of irrelevant matches can be abated by 70% (Varol et.al., 2014). A set of tables representing the pronunciation rules for specific languages are designed for BMPM, where the language of the word can be recognized from its spelling. The design includes nearly 200 rules to specify the language in this technique. If the language cannot be determined, special kind of generic rules are used to encode the word.

NYSIIS. NYSIIS algorithm was developed in 1970 as a part of New York State Identification and Intelligence System project headed by Robert L. Taft, which produces a canonical code similar to Soundex (David Hood, 2004). Unlike Soundex, NYSIIS

retains the information regarding position of vowels in the encoded word by transforming them all to ‘A’. It generates only alphabetic code and is extensively used in record linkage systems (Balabantaray et.al., 2012, Snae, 2007).

Daitch Mokotoff Soundex. Daitch Mokotoff Soundex System is developed by Randy Daitch and Gary Mokotoff of the Jewish Genealogical Society (New York) in 1985. The algorithm is mainly used for determining the near matches with Eastern European surnames which include Russian and Jewish names. Similar to Soundex, the algorithm also encodes into digits by extending it to a complete 6-digit code. The conversion rules are much complicated and involves groups of characters for encoding (Soundex Coding, 2016).

Phonex and Phonix. Phonex is a technique in which words are pre-processed before encoding. In order to overcome defects of Phonex, Phonix has been introduced with a number of transformations in the beginning, ending, and in the middle of the word (Varol, 2011). Phonix is considered to be the variant of Soundex, where a prior mapping involves nearly 160 letter-group conversions to normalize the string. For example, X is converted to ‘ECS’, PSv is converted to Sv (where ‘v’ is any vowel) if it occurs at the start of string. Phonix also produces a four letter code like Soundex, which is highly useful when an exact index search is required but, due to the truncation of code, it is not beneficial when the complete string matching should be assessed (Zobel et.al., 1996).

Metaphone. In 1990, a new technique considering diphthongs (combination of two or more letters) of words was developed by Lawrence Philips, known as Metaphone (Lawrence, 1990). It indexes the original word based on the pronunciation rules in English. It retains more information than other variants of Soundex as the letters are not

defined into groups (Nikita, 2011). The final code of Metaphone includes 16 consonant letters but retains the vowels if at the beginning of the word.

Bhattacharjee et.al has stated that the technique is mainly used for data cleaning in the text files to remove erroneous data (Bhattacharjee et.al., 2013). Pande et.al detailed that the Metaphone has its extended usage in stemming, which improves performance in Information Retrieval (IR) (Pande et.al., 2011). David Hood cited that though the algorithm is sensitive to combination of letters like 'TH', it is not subtle enough with the vowels especially at the postvocalic L and R (David Hood, 2004).

Double Metaphone. Double Metaphone, popularly known as DMetaphone, is an enhancement to Metaphone algorithm by Lawrence Phillips in 2000. It is distinctive from other algorithms as it generates two code values – one representing the basic version and other representing the alternate version. Unlike Soundex, DMetaphone encodes groups of letters called diphthongs according to a set of rules (Varol et.al., 2011). The encoding process involves rules which consider the words from different origins such as Eastern European, Italian, Chinese and another words.

Caverphone. In pace, the specified algorithms are not suitable for a particular database, named Caversham, which is mainly used for data source linkage. The algorithm, known as Caverphone, which is analogous to Metaphone with some rules subsequently applied, is enforced by David Hood in 2002 to encode the data of Caversham database (David Hood, 2004). The algorithm is later improvised in 2004 to Caverphone 2.0, to increase its accuracy and efficiency by applying more set of rules. David Hood, (David Hood, 2004) also stated that the algorithm is efficient by giving

precise matches when compared to Soundex and Metaphone algorithms for linking data sources (Carstensen, 2005).

Phonetic Matching for Spanish

One of the major applications of phonetic matching algorithms is its appliance to different languages. The limitations of Soundex make it straightforward that the algorithm is specifically designed for English language. Also, the grouping articulation of the English letters and limit to the four characters makes it less efficient to detect common spelling errors in other languages such as Spanish (Angeles et.al., 2015).

Spanish Soundex. In 2012, Am´on et.al have proposed an improvement to Soundex algorithm by including Spanish letters making it feasible to obtain phonetic codes for Spanish words (Am´on et.al, 2012). The encoding also removes the dependency on the first letter by converting all the letters into digits. As a result, the Spanish Soundex is more accurate than the original Soundex in finding near matches for Spanish words. In 2014, Angeles et.al, have improvised the algorithm to make the encryption code resizable (Angeles et.al, 2015).

Spanish Metaphone. Alejandro Mosquera had developed Metaphone algorithm in 2012, for Spanish language by adapting the techniques from the algorithm used for English Language (Mosquera, 2012). Unlike Spanish Soundex, the Spanish Metaphone retains the information related to vowels. The encoded word results in group of characters.

CHAPTER III

Phonetic Matching Algorithms

This project involves implementation of different phonetic matching algorithms for both English and Spanish Languages. The algorithms for the Soundex, Daitch-Mokotoff Soundex, NYSIIS, Caverphone, Metaphone, Double Metaphone, Spanish Soundex, and Spanish Metaphone are illustrated in this chapter.

Soundex

Russell had categorized letters of alphabet based on their phonetic description. The steps for generating phonetic code using Soundex algorithm are as below:

1. Retain the first letter of the word.
2. For the remaining letters, numbers are to be assigned based on the phonetic description as shown in the below table 2:

Table 2

Soundex Transformation

Phonetic Description	*Letters to encode*	*Encoding Digit*
Oral Resonant	A, E, H, I, O, U, W, Y	0
Labials and labio-dentals	B, F, P, V	1
Gutterals and sibilants	C, G, J, K, Q, S, X, Z	2
Dental-mutes	D, T	3
Palatal-fricative	L	4
Labio-nasal and Lingua-nasal	M, N	5
Dental fricative	R	6

3. From the string obtained by the above manipulations, remove all pairs of same digits that occur beside each other.
4. All zeroes, obtained from the above step, are removed from the string.
5. The first four characters are considered to be Soundex code, and are right padded with zeroes if the string is deficit of four characters (Odell et.al., 1918, Carstensen 2005).

Daitch-Mokotoff Soundex

The Daitch-Mokotoff Soundex Algorithm is used to reduce false positives with the number of complex rules enforced in the algorithm. The transformation of string to Daitch-Mokotoff code uses the following table. The order of transformations is in the same order of letter groupings in the table 3.

Table 3

Daitch-Mokotoff Transformation

Letter combinations to encode	*At the Start*	*After a Vowel*	*Other*
AI, AJ, AY, EI, EY, EJ, OI, OJ, OY, UI, UJ, UY	0	1	
AU	0	7	
IA, IE, IO, IU	1		
EU	1	1	
A, UE, E, I, O, U, Y	0		
J	1	1	1

(continued)

Letter combinations to encode	*At the Start*	*After a Vowel*	*Other*
SCHTSCH, SCHTSH, SCHTCH, SHTCH, SHCH, SHTSH, STCH, STSCH, STRZ, STRS, STSH, SZCZ, SZCS	2	4	4
SHT, SCHT, SCHD, ST, SZT, SHD, SZD, SD	2	43	43
CSZ, CZS, CS, CZ, DRZ, DRS, DSH, DS, DZH, DZS, DZ, TRZ, TRS, TRCH, TSH, TTSZ, TTZ, TZS, TSZ, SZ, TTCH, TCH, TTSCH, ZSCH, ZHSH, SCH, SH, TTS, TC, TS, TZ, ZH, ZS	4	4	4
SC	2	4	4
DT, D, TH, T	3	3	3
CHS, KS, X	5	54	54
S, Z	4	4	4
CH, CK, C, G, KH, K, Q	5	5	5
MN, NM		66	66
M, N	6	6	6
FB, B, PH, PF, F, P, V, W	7	7	7
H	5	5	
L	8	8	8
R	9	9	9

The above algorithm generates a Daitch-Mokotoff code of 6 digits (Nikita, 2011).

NYSIIS

The transformations for generating NYSIIS code is as shown below:

1. If the first character of the name is a vowel, remember it.
2. Remove all 'S' and 'Z' chars from the end of the name.
3. Transcode first characters of name as follows,

   MAC → MC, PF → F
4. Transcode trailing strings as follows,

   IX→ IC

   EX → EC

   YE, EE, IE → Y

   DT, RT, RD, NT, ND → D
5. Repeat this last step as necessary.
6. Transcode 'EV' to 'EF' if not at start of name.
7. Use first character of name as first character of key
8. Remove any 'W' that follows a vowel
9. Replace all vowels with 'A' and collapse all strings of repeated 'A' to one
10. Transcode 'GHT' to 'GT'
11. Transcode 'DG' to 'G'
12. Transcode 'PH' to 'F'
13. If not first character, eliminate all 'H' preceded or followed by a vowel
14. Change 'KN' to 'N', else 'K' to 'C'
15. If not first character, change 'M' to 'N'
16. If not first character, change 'Q' to 'G'

17. Transcode 'SH' to 'S'
18. Transcode 'SCH' to 'S'
19. Transcode 'YW' to 'Y'
20. If not first or last character, change 'Y' to 'A'
21. Transcode 'WR' to 'R'
22. If not first character, change 'Z' to 'S'
23. Transcode terminal 'AY' to 'Y'
24. Remove trailing vowels
25. Collapse all strings of repeated characters
26. If first character of original name is a vowel, prepend to code (or replace first transcoded 'A') (Steve Hobbs, 2006)

Metaphone

The step by step procedure of Metaphone encoding is as described below:

1. Drop duplicate adjacent letters, except for C.
2. Transform the word using following table 4:

Table 4

Metaphone Transformation I

KN	GN	PN	AE	WR
N	N	N	E	R

3. MB → B only if MB at the end of word.
4. Replace the diphthongs using the table 5:

Table 5

Metaphone Transformation II

G	CIA, CH	SCH, C	CI, CE, CY	DGE, DGI, DGY	D
K	X	K	S	J	T

5. Drop 'G' if followed by 'H' and 'H' is not at the end or before a vowel. Later following transformations are carried out on the words as shown in table 6:

Table 6

Metaphone Transformation III

GH	GN	GNED	GI, GE, GY, ^GG
H	N	NED	J

6. Drop 'H' if after vowel and not before a vowel.
7. It is followed by the replacements from the below table 7:

Table 7

Metaphone Transformation IV

CK	PH	Q	SH, SIO, SIA	TIAO	TH	TCH	V
K	F	K	X	IAO	O	CH	F

8. 'WH' transforms to 'W' if at the beginning. Drop 'W' if not followed by a vowel.
9. Drop 'Y' if not followed by a vowel.
10. Transform ‘Z’ to ‘S’ and drop all the vowels unless it is in beginning.

The above algorithm generates a Metaphone code up to 12-letter (Lawrence, 1990).

Double Metaphone (DMetaphone)

Unlike other phonetic matching algorithms, Double Metaphone, commonly known as DMetaphone, produces two code values – one considered as primary representation and other as alternative version. It comprises of large number of rules by considering words from various origins such as Eastern European, Italian, Chinese etc. As the transformation rules are numerous, the algorithm can be easily referred from the mentioned reference (Dobbs, 2000).

Caverphone

The algorithm for Caverphone 2.0 follows the steps as below (David, 2004):

1. Convert all letters to lower case.
2. Remove the letter ‘e’ at the end.
3. Transform the word using following tables 8, 9, 10:

Table 8

Caverphone Transformation I

cough	rough	tough	enough	gn	mb
cou2f	rou2f	tou2f	enou2f	2n	m2

Table 9

Caverphone Transformation II

cq	ci	ce	cy	tch	c	q	x	v	dg
2q	si	se	sy	2ch	k	K	k	f	2g

Table 10

Caverphone Transformation III

tio	tia	d	ph	b	sh	z
sio	sia	t	fh	P	s2	s

4. Replace all vowels at the word beginning with ‘A’; in other cases, replace them with ‘3’. At the next step, it is necessary to replace using the following tables 11 and 12 (the legend: s+ - group of consecutive letters, ^h - letter at the start, w$ - letter at the end):

Table 11

Caverphone Transformation IV

j	^y3	^y	y	3gh3	gh	G	s+	t+	p+
y	Y3	A	3	3kh3	22	K	S	T	P

Table 12

Caverphone Transformation V

k+	f+	m+	n+	w3	wh3	w	^h	r3	r$
K	F	M	N	W3	Wh3	2	A	R3	3

5. Remove all digits ‘2’. If there is a digit 3 at the end, replace it with A. After that all the digits ‘3’ are removed.
6. Truncate the word to 10 letters or fill it to 10 letters with digit 1 (David Hood, 2004).

Spanish Soundex

To overcome the limitations of Soundex algorithm for Spanish words, letters from Spanish language are incorporated into the algorithm. The steps of obtaining Spanish Soundex code is as follows:

1. The string is converted to uppercase by ignoring all the punctuations.
2. Eliminate the following letters: 'A’, E ‘, ' I', ' O’, 'U ', ' H ', ' W '.
3. Change the letters of the following obtained string by the corresponding digits as shown in the table 13:

Table 13

Spanish Soundex Transformation

Letters to encode	*Encoding Digit*
P	0
B, V	1
F, H	2
T, D	3
S, Z, C, X	4
Y, LL, L	5
N, Ñ, M	6
Q, K	7
G, J	8
R, RR	9

The resultant code will only comprise of digits and hence the dependency on first letter does not exist in Spanish Soundex (Am´on et.al, 2012).

Spanish Metaphone

The step by step procedure for obtaining Spanish Metaphone code is as follows:

1. Convert all letters to lowercase.
2. Make the transformations as shown in table 14:

Table 14

Spanish Metaphone Transformation

á	ch	Ç	é	í	ó	ú	ñ	gü	ü	b	ll
A	X	S	E	I	O	U	NY	W	U	V	Y

3. Convert all letters in string to uppercase.
4. If the first letter is a vowel, retain the first letter.
5. Drop duplicate adjacent letters, except for C.
6. Transform the string as follows

   CC → X,

   CE, CI → Z,

   C → K.
7. GE, GI → J, or G is retained.
8. ‘Hv’ is transformed to ‘v’, where v is a vowel. Otherwise, ‘H’ is preserved.
9. Q → K, if not followed by U. Else ‘QU’ is removed.
10. W → U.
11. S → ES, if it is present at the start of string and is followed by an vowel,

    Otherwise, S is retained.
12. X → EX, if it is present at the start of string and is followed by an vowel,

    Otherwise, X is retained (Mosquera, 2012).

CHAPTER IV

Data Pre-Processing

Need for Data Preparation

Data pre-processing is considered to be an important phase in data mining because the data that is collected from various sources lacks consistency, which makes it unsuitable to directly apply data processing algorithms (Zhang, Zhang, & Yang, 2003). The raw data can also be incomplete with missing values of some attributes. In some cases, we can encounter noisy data with some unwanted values to a given attribute. As a result, we preprocess the data into a suitable format to apply different algorithms.

Until now, various experiments were conducted on finding phonetic matches for misspelled words of personal names (Shah, 2014). But there is only little exploration in finding the phonetic matches for dictionary words using these algorithms. Hence, in this project we mainly concentrated on obtaining the phonetic matches for misspelled words of English and Spanish diction.

Reference Dataset Preparation

The reference datasets for the experiment are prepared as follows. For the English dictionary dataset, all the words are extracted from the reference (Lawler, 1999) and a list is formed. This list comprises of 267,750 correct, non-duplicate words. Phonetic codes are generated for each of these words, by applying the algorithms illustrated in the previous chapter. A dataset is created with these English words and their corresponding phonetic codes. This dataset is used as a reference dataset for obtaining the suggestions for misspelled English words.

Similarly, Spanish wordlist is extracted from (Diccionario). The list consists of 95,487 correct words. Phonetic codes are generated using Spanish phonetic matching algorithms. Another dataset, having these Spanish words and their corresponding phonetic codes are created to use as reference for retrieving suggestions to misspelled words.

Synthetic Dataset Preparation

According to Kukich (Kukich, 1992), nearly 80% of problems of misspelled words can be addressed either by addition of single letter, or replacement of single letter or swapping of letters. Therefore, synthetic datasets are generated by executing addition, deletion, swapping, and replacement of letters.

From the above mentioned correct word list of English language, different pairs of datasets are generated by randomly selecting the words. Each pair consists of correct words as reference data and their corresponding manipulated words as misspelled data. The generation of synthetic datasets is shown in Figure 1.

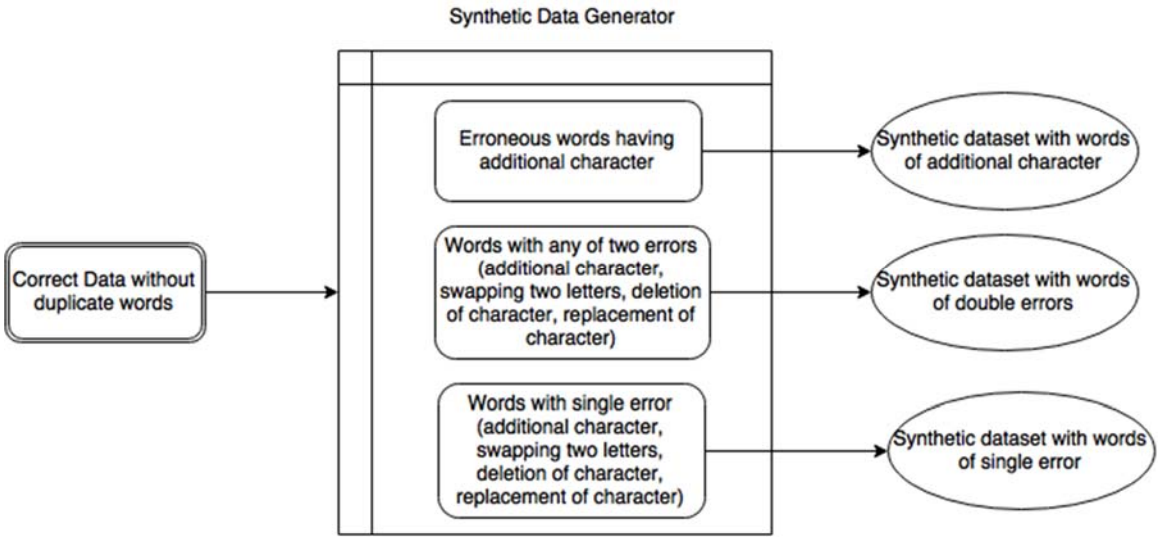

Figure 1. Synthetic datasets generation for analysis of various algorithms.

While creating the manipulated data, words with three types of errors are generated, namely, words with additional character, words having single error (replacement or substitution of character or swapping of two characters), and words having double errors (two single errors). The generated words are accumulated into datasets of different sizes 200, 500, and 800. Four datasets are generated for each size and each type of error. Hence, a total of thirty-six pairs of correct and manipulated datasets are generated.

By the same token, thirty-six pairs of correct and manipulated datasets are generated with data sizes 200, 500, and 800 for the Spanish language.

Real World Misspelled Data

Apart from the synthetic data, the performance of the algorithms is also analyzed on real-world data. For English, the misspelled data is referred from (Hempel, 2014) having nearly 4,200 misspelled words along with the correct words. In the same way, the Spanish data is retrieved from (Planeta Curioso, 2008). As there is only little research in the field of misspelled words in Spanish language, the data size of misspelled words is only about 100.

CHAPTER V

Implementation of Phonetic Algorithms for Performance Evaluation

Complication in the recovery of data is the result of type errors, misspelled words, inconsistent expression habit, and different formats. Matching of words can be defined as the process of determining whether both the words are similar or not. With typographical errors, often there would be interchanging of letters or misspelling of words.

Though Soundex and Metaphone are naïve algorithms being used in different applications as embedded tools, each of them have their own disadvantages. Soundex mainly depends on the first letter of the word. It has a high overhead in retrieving the near matches and it does not consider the phonetic sounds of vowels. In spite of addressing the above problems with Metaphone algorithm, Metaphone only has less accuracy in obtaining the proper matches to the misspelled word.

State-of-the-Art Meta-Soundex Algorithm

To overcome the limitations in both algorithms, a new algorithm is developed, namely, Meta-Soundex. The schematic design of Meta-Soundex algorithm is shown in Figure 2.

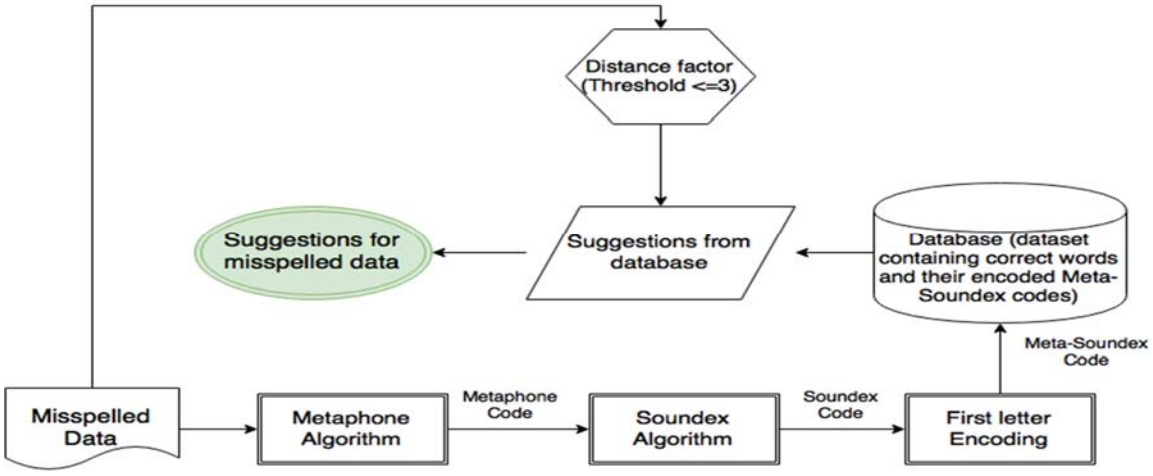

Figure 2. Schematic design of suggestions retrieval for Meta-Soundex Algorithm.

Meta-Soundex Algorithm - English

1. Convert all the letters to uppercase.
2. Encode using Metaphone algorithm to retain vowel sounds and diphthong combinations.
3. Encode the obtained string using Soundex algorithm.

Encode the first the letter using the following table 15 (Soundex Coding, 2016).

Table 15

Meta-Soundex Transformation

Letters to encode	*Encoding Digit*
A, E, I, O, U	0
J, Y	1
D, T	3
S, Z, C	4
X, G, H, K, Q	5
N, M	6
B, F, P, V, W	7
L	8
R	9

Meta-Soundex Algorithm - Spanish

1. Convert all the letters to uppercase.
2. Encode using Metaphone algorithm to retain vowel sounds and diphthong combinations.

3. Encode the obtained string using Soundex algorithm.

The above algorithm generates a Meta-Soundex code of variable length for Spanish language.

Distance Factor on the retrieved approximate matches – Meta-Soundex

The Meta-Soundex code is sent to the database to obtain approximate matches for the input data. After the approximate matches are retrieved, the distance factor between the misspelled word and the retrieved matches is calculated using Levenshtein Edit Distance (LED) method (Diman et.al, 2014). The threshold of the distance is set to 3, as the maximum number of errors in the synthetic data is less than 3, whereas for real-world data the distance factor is mostly observed to be 3. If LED is less than or equal to 3, then the word is considered to be nearest match for the misspelled word.

The proposed Meta-Soundex algorithm improves its precision over Soundex as it includes encoding of vowel sounds and combinational phonetic sounds before grouping individual letters. The accuracy of Meta-Soundex is higher than Metaphone transforming it to be more efficient than other algorithms as shown in Figure 3.

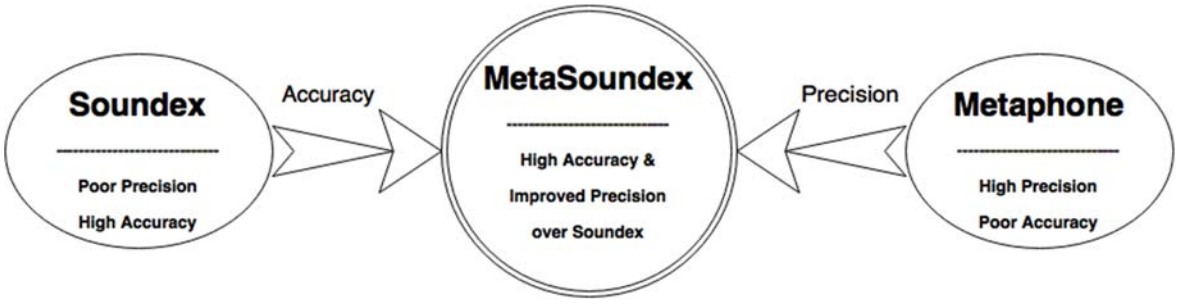

Figure 3. Design of Meta-Soundex algorithm.

Architectural design of comparison of phonetic matching algorithms

The design of the experimental analysis supports two languages, English and Spanish. It comprises of a language selector and file uploader to upload two input files -

one with correct data indicated as “reference data file” and other with some amount of data crooked represented as “incorrect data file”. The data source of the design encloses two different schemas referred as Spanish dictionary and English dictionary for both Spanish and English languages, respectively. Each schema encompasses the dictionary words and their corresponding phonetic codes to give approximate matches for the misspelled data.

Primarily, the language is selected to redirect the process to the corresponding simulator. Both the input files are uploaded to the design, which are correlated to elicit the mismatched words from the crooked data as shown in Figure 4. This errant data list is fed to the pairing simulator.

The simulator of the Spanish language contains the functionality of three phonetic matching algorithms - Soundex, Metaphone and the proposed Meta-Soundex, whereas, the simulator of English language contains the functionality of six algorithms – Soundex, Metaphone, Caverphone, DMetaphone, NYSIIS and the proposed Meta-Soundex.

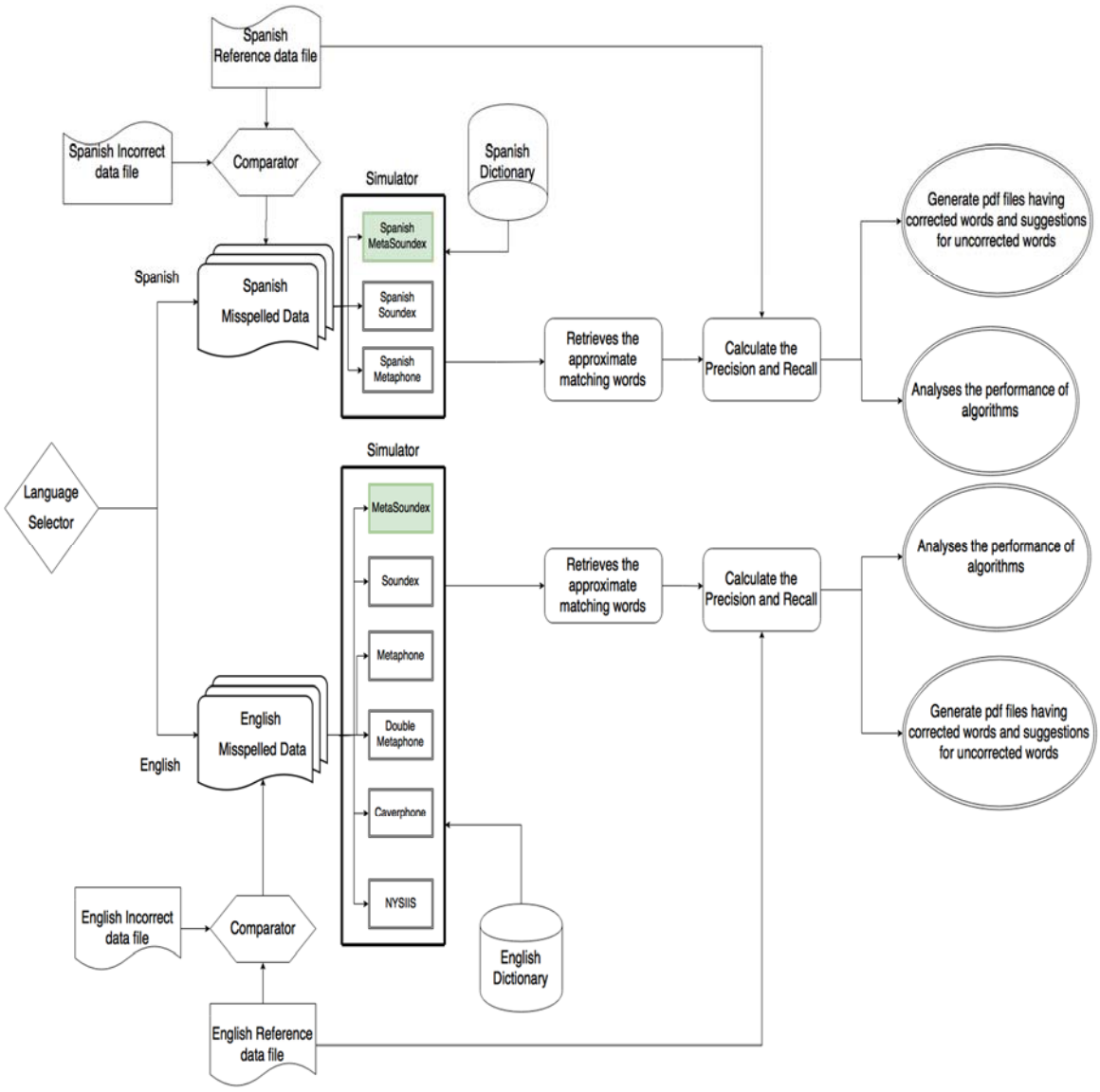

Figure 4. Architectural design of comparison of phonetic matching algorithms.

The simulator generates phonetic codes by executing phonetic matching algorithms of the corresponding language, for the errant data. These codes are compared to the phonetic codes present in a data source and the matched word lists are retrieved as the approximate suggestions. These matched words are evaluated by comparing with the reference file to calculate precision and recall, which would symbolize the better algorithm. Along with the analysis, two pdf files are generated for depicting the results.

One of them contains the corrected words for misspelled words while the other contains suggestions for the misspelled words from each algorithm.

Evaluation Metrics

The performance of phonetic matching algorithms used for information retrieval is evaluated by calculating precision, recall and F - measure.

Precision. Precision gives the total number of true positives obtained over the total number of suggestions for the obtained true positives.

$$P = \frac{\sum p}{\sum Number\ of\ suggested\ words\ for\ each\ corrected\ word}$$

$$\text{, where } p = \begin{cases} 1, if\ the\ word\ is\ corrected \\ 0, if\ the\ word\ is\ not\ corrected \end{cases}$$

$$P = cumulative\ precision\ of\ an\ algorithm$$

Recall. Recall provides the total number of relevant words over the total number of suggestions (Kelkar, 2012).

$$R = \frac{Number\ of\ corrected\ words}{Total\ number\ of\ misspelled\ words}$$

$$\text{, where } R = recall\ or\ accuracy\ of\ an\ algorithm.$$

F-measure. The F – measure is calculated based on precision and recall and is defined as the harmonic mean of precision and recall. It is given by,

$$F = \frac{2 \times P \times R}{P + R}$$

$$\text{, where } F = F - measure\ of\ the\ algorithm.$$

For the analysis, the maximum F - Measure for different datasets are considered, which vary in size and features.

CHAPTER VI

Tools and Software

Following tools and software are used to implement phonetic matching algorithms and for the development of phonetic matching toolkit.

Programming Languages

Java. Java is used as a programming language because of several reasons. It is a high level object oriented language which is simple to write. Also, its platform independence makes it more suitable for this research.

HTML5, JQuery, JavaScript, JSP, CSS3, Bootstrap. The front-end web technologies are highly useful to develop a versatile and user friendly toolkit that provides easy access to the users, when opened from web browser.

Software Tools

Eclipse Software Development Kit. Eclipse is an excellent platform to run java code that has a very good integration with the open source Apache Tomcat. It has a nice and easy to handle user interface.

Apache-Tomcat Server 8.0.36. Apache Tomcat Server is one of the best platforms to run web applications. It helps the users to easily load the web applications on the localhost without any external server.

Microsoft SQL Server 2012 R2 (MS SQL). MS SQL server is a remarkable database to store and retrieve data for Java applications. It has a nice and easy to handle user interface, where user can create his/her own schema as per the requirement.

CHAPTER VII

Analysis and Results

The project illustrates the performance of different algorithms on datasets of particular size having various types of errors. From the results, it can be stated that the variations in performance is also dependent on the type of error.

Analysis on Synthetic Data English

The experimental results show that Meta-Soundex excels in retrieving more accurate words compared to other techniques for all types of errors, which is followed by Soundex and Double Metaphone. The test results are obtained from four different datasets for various sizes of data ranging from 200 to 800.

Recall. Recall for different techniques on the datasets having synthesized data of English dictionary words is shown in the Figures 5, 6, and 7 for different data sizes of 200, 500, and 800 respectively.

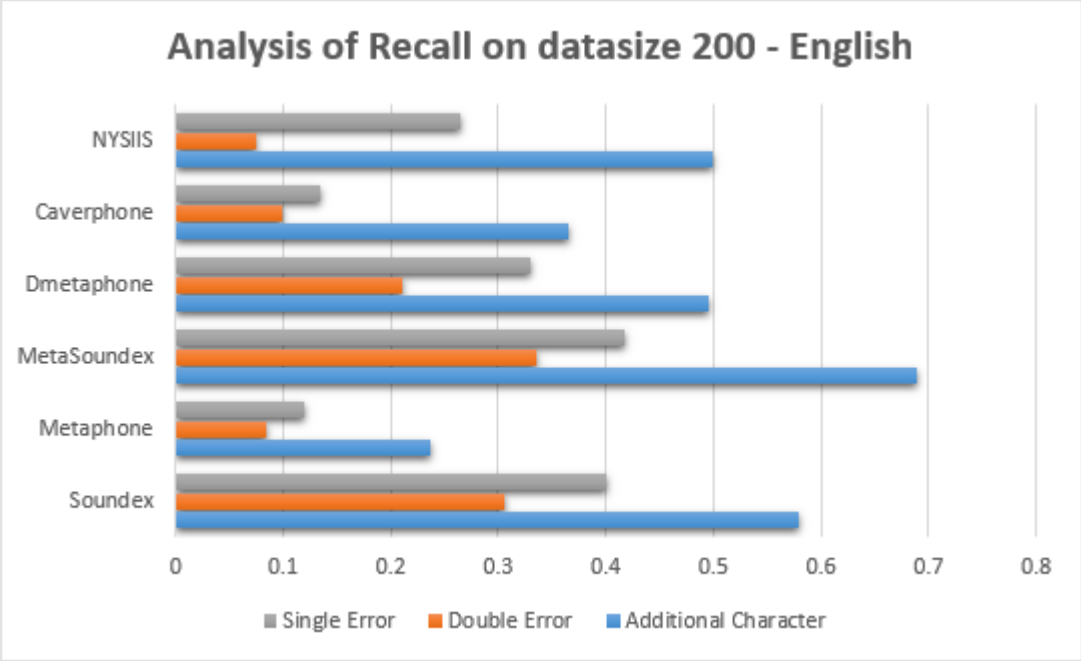

Figure 5. Recall for different techniques on synthetic English dataset of size 200.

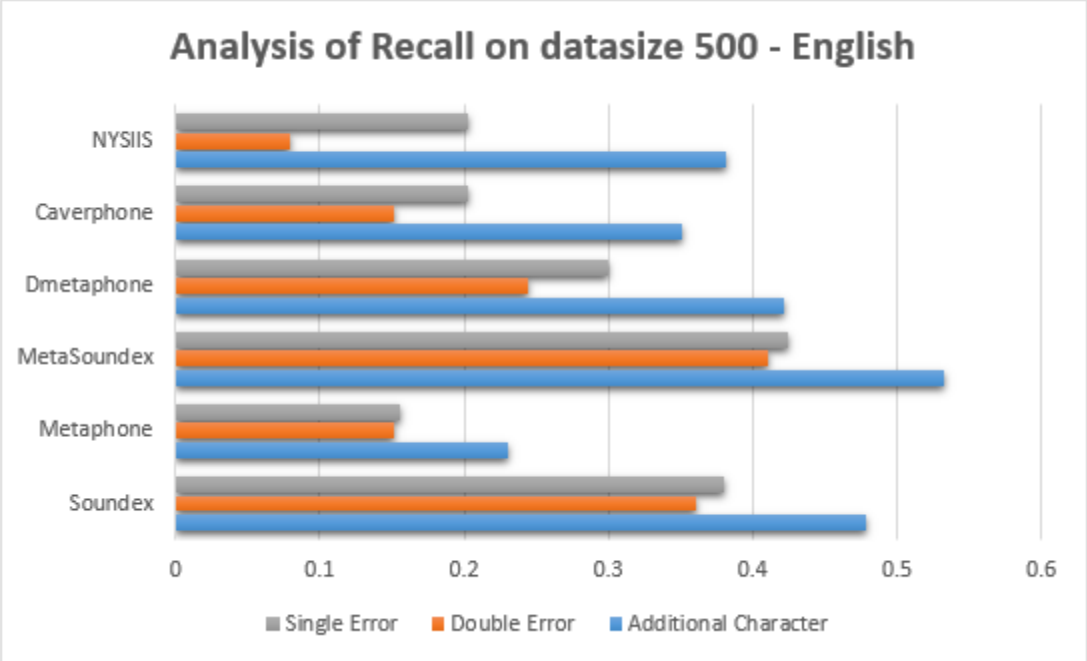

Figure 6. Recall for different techniques on synthetic English dataset of size 500.

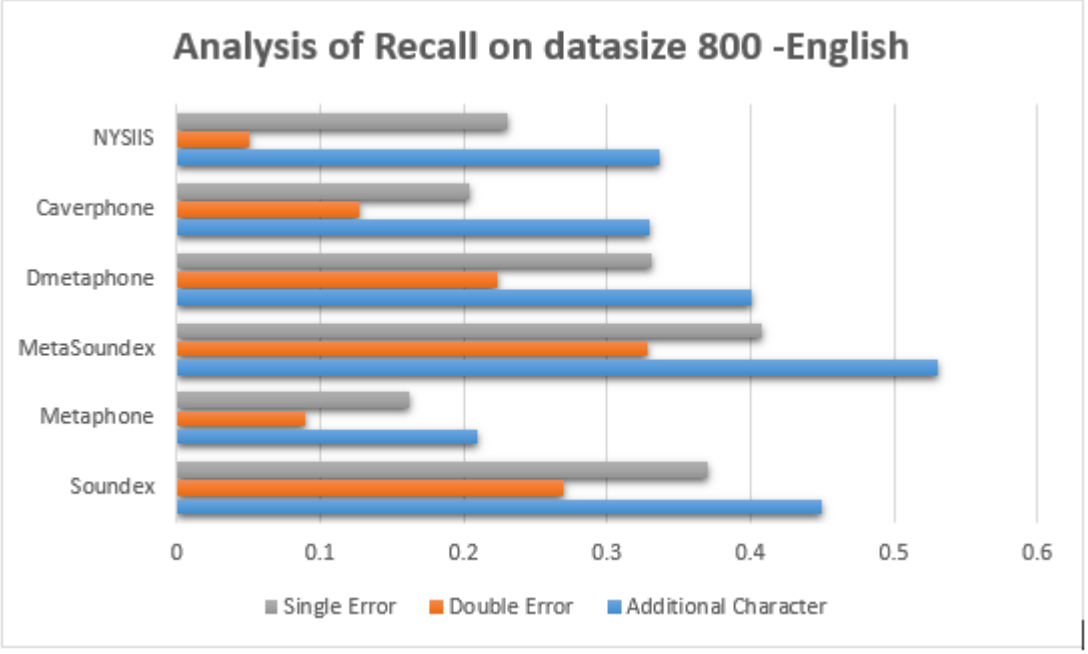

Figure 7. Recall for different techniques on synthetic English dataset of size 800.

From the above figures it can be clearly observed that the state-of-the-art Meta-Soundex algorithm has highest accuracy, whereas, Metaphone has the lowest accuracy of all the algorithms. It can also be observed that the recall value is highly dependent on the type of error. The recall value is high for the erroneous wordlist having additional character, while, it is low for the wordlist having two errors in each word. Apart from Meta-Soundex, the Soundex shows its high recall value in the second place, followed by DMetaphone, NYSIIS and Caverphone in succession. It is completely arbitrary that the recall values are either increased or decreased by the change of datasize.

F-measure. The F-measure represents the overall performance and efficiency of the algorithm, which is calculated using precision and recall. F-measure of various techniques for data sizes ranging from 200 to 800 are indicated in the Figures 8, 9, and 10.

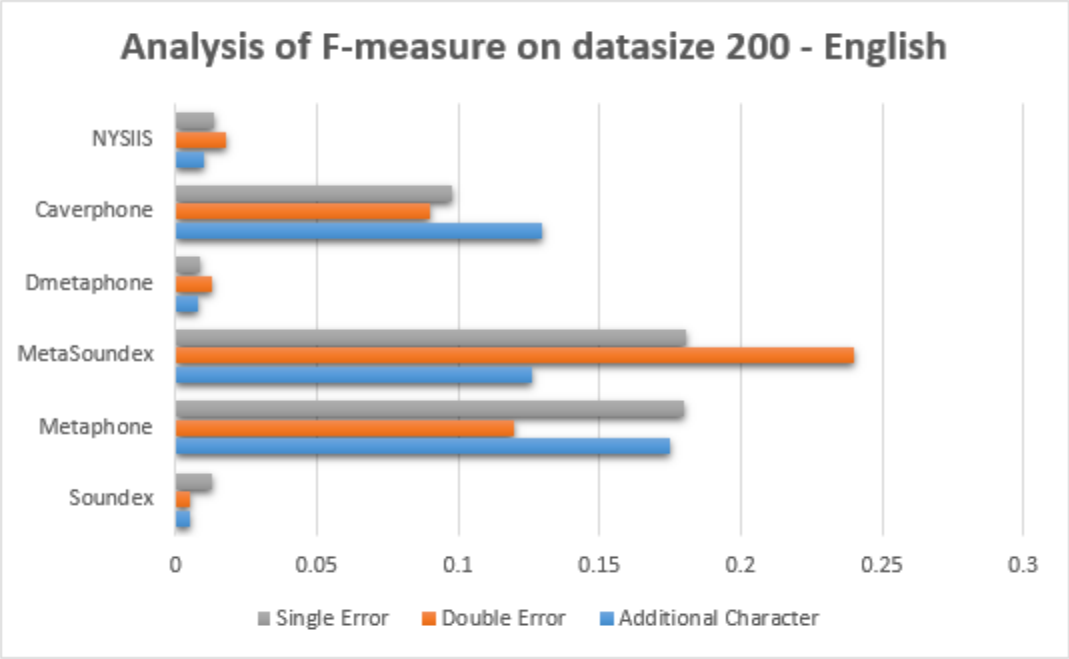

Figure 8. F-measure for different techniques on synthetic English dataset of size 200.

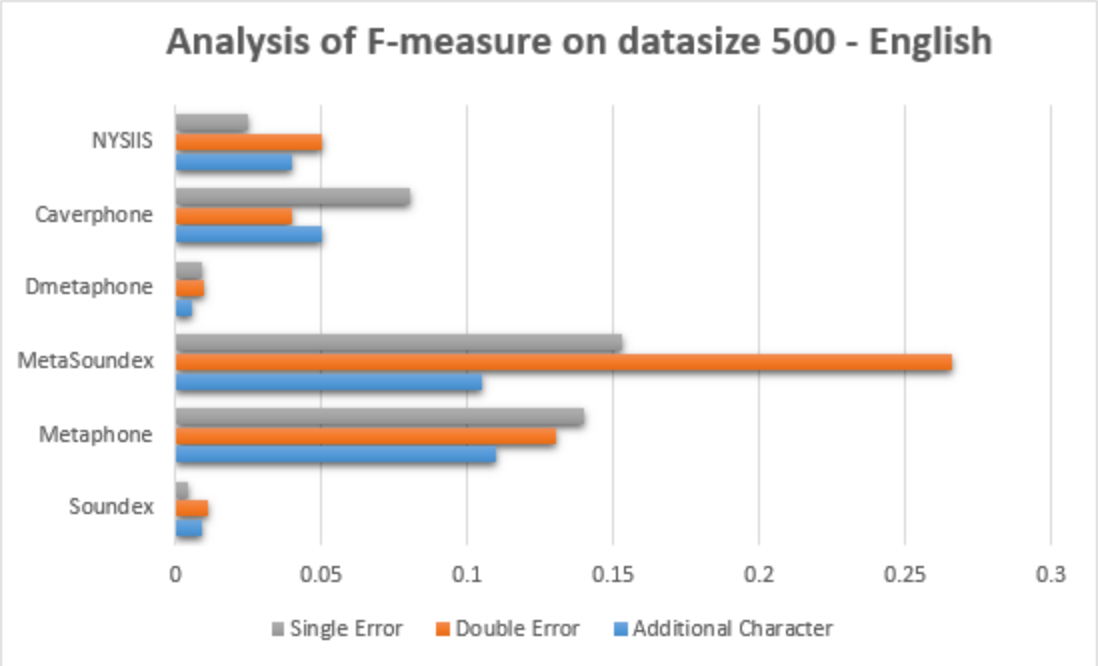

Figure 9. F-measure for different techniques on synthetic English dataset of size 500.

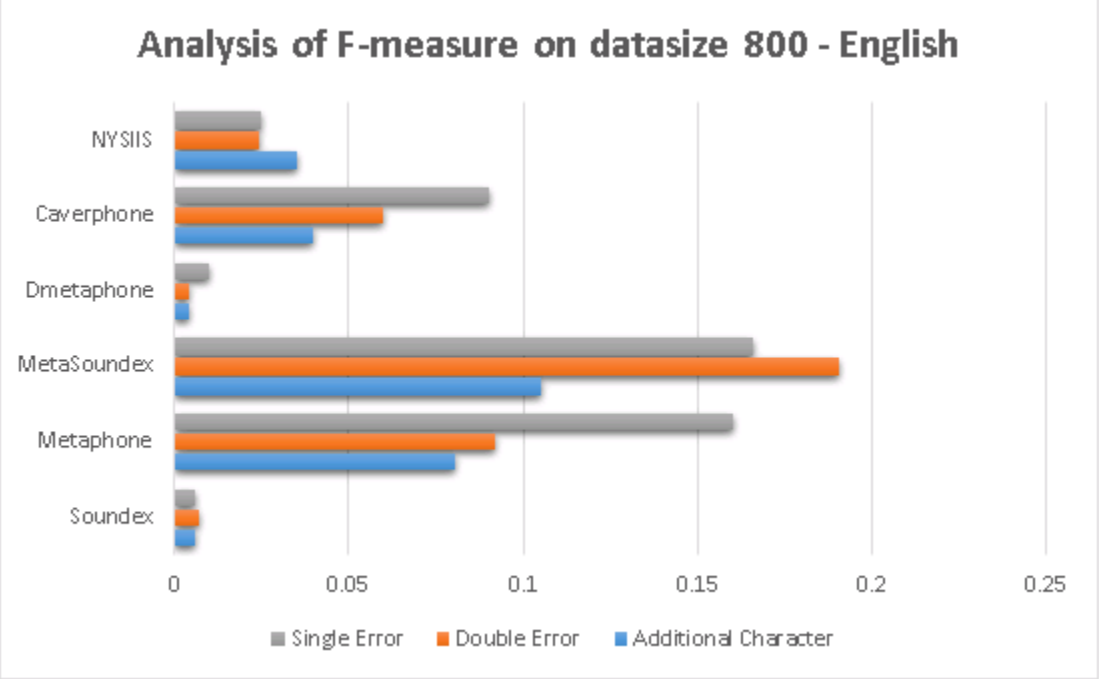

Figure 10. F-measure for different techniques on synthetic English dataset of size 800.

From the experimental analysis, it can be clearly stated that Meta-Soundex has better performance than all other algorithms for any data size and type of error, reducing

the number of false positives and noise in the retrieved suggestions. It is followed by Metaphone and Caverphone. Soundex and DMetaphone shows the lowest performance in all the considered arenas. Though DMetaphone has noticeable recall values, it has low precision similar to Soundex due to retrieval of suggestions for both the primary and secondary codes.

Based on the type of error, Meta-Soundex shows high performance for the erroneous list having two errors, while it reflects lower performance for the words having additional character. From the figures, it can also be inferred that all other algorithms show average performance for double errors irrespective of size of dataset for English words. The results also state that the performance is not highly dependent on the size of the dataset for all the algorithms.

Analysis on real-world data - English

In addition to the analysis on synthetic dataset, the experimental analysis is also conducted on the real-world misspelled data to check the performance of the algorithms.

Recall. The recall of different techniques obtained from the analysis are shown in Figure 11. The analysis is performed on a real-word dataset of size, nearly 4,200.

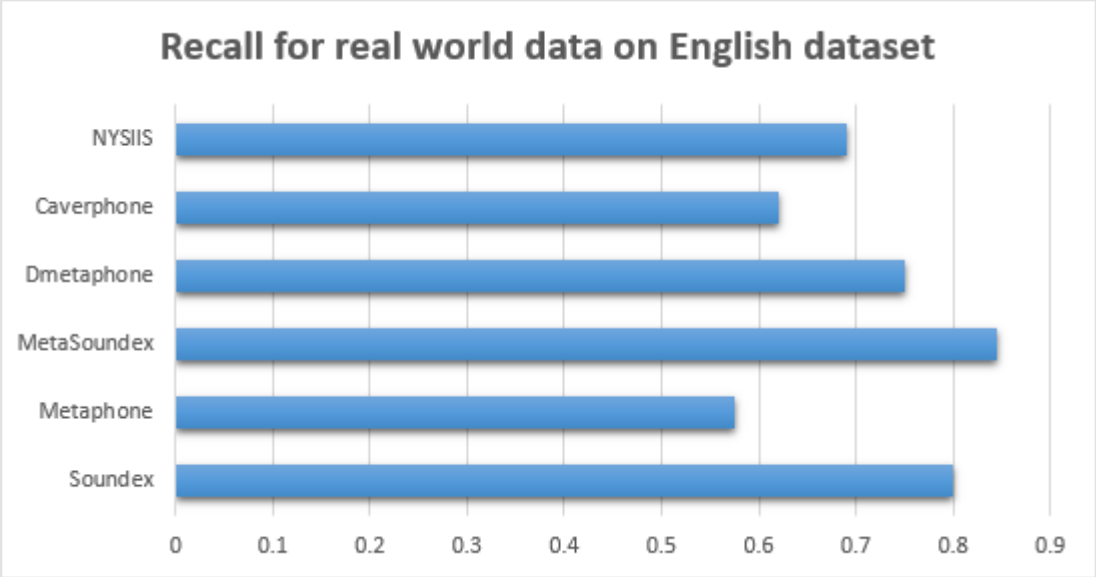

Figure 11. Recall for different techniques on real-world English dataset. Size of the dataset is nearly 4,200.

From the above, it can be stated that the Meta-Soundex has the exceptional recall value showing its high accuracy on the real world-data, which is followed by Soundex and DMetaphone, while Metaphone has the lowest accuracy.

F-measure. The performance evaluation for different techniques on the real-world dataset of English words is shown in Figure 12.

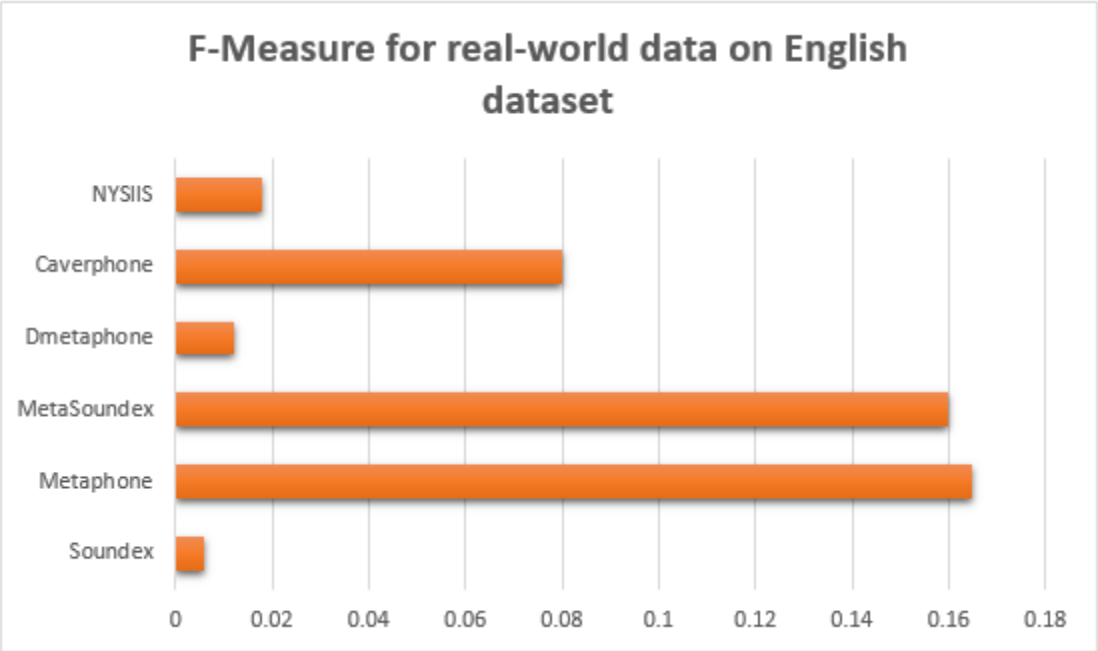

Figure 12. F-measure for different techniques on real-world English dataset. Size of dataset is nearly 4,200.

As shown above, on the real-world data, Metaphone shows highest performance with a miniature difference to the Meta-Soundex algorithm. Despite of its low recall value, Caverphone also shows better performance due to its decent precision value. The performance of Meta-Soundex has an exceptional increase over Soundex, showing that the state-of-the-art Meta-Soundex has achieved high precision over Soundex and high accuracy over Metaphone, making it more balanced than other algorithms.

Analysis on synthetic data - Spanish

Analogous to English, Meta-Soundex has high accuracy than Soundex and Metaphone for Spanish misspelled words. The results of the experimental analysis of Spanish Soundex, Spanish Metaphone, and Spanish Meta-Soundex on the synthetic datasets for different errors (additional character, double error and single error) of varying data sizes from 200 to 800 are shown below:

Recall. Recall for different techniques on the datasets having synthesized data of Spanish dictionary words is shown in the Figures 13, 14, and 15 for different data sizes of 200, 500, and 800 respectively.

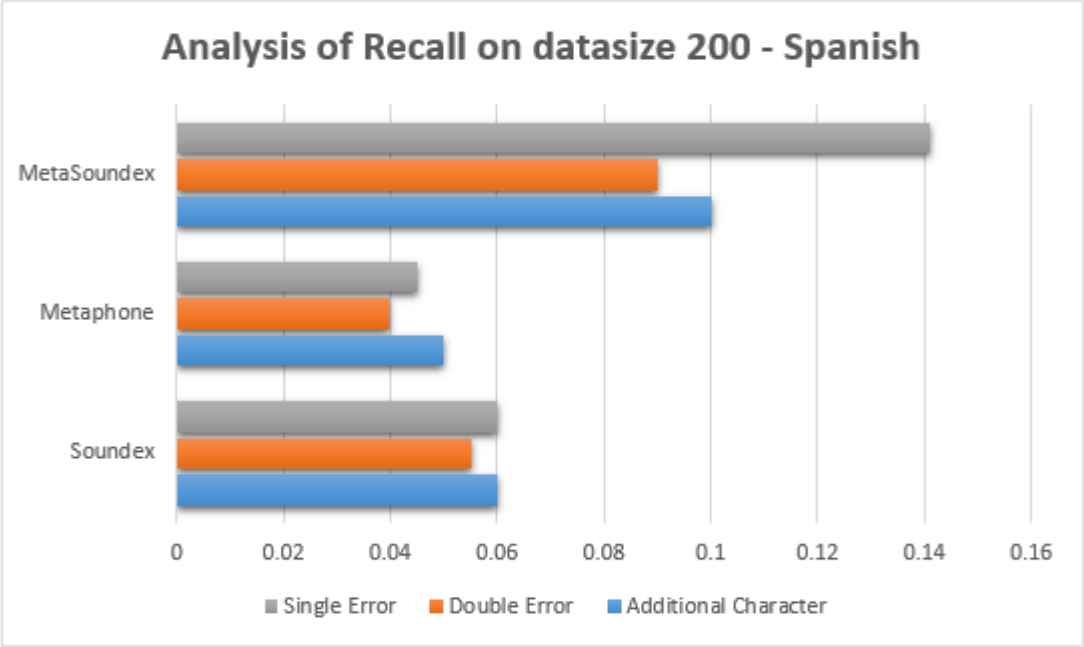

Figure 13. Recall for different techniques on synthetic Spanish dataset of size 200.

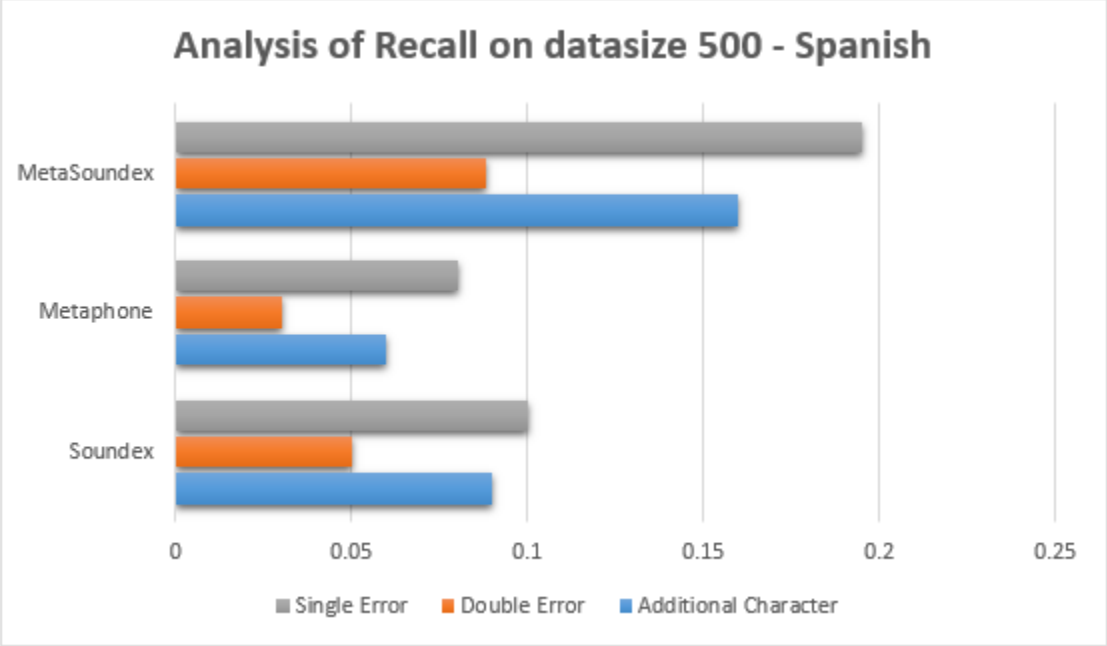

Figure 14. Recall for different techniques on synthetic Spanish dataset of size 500.

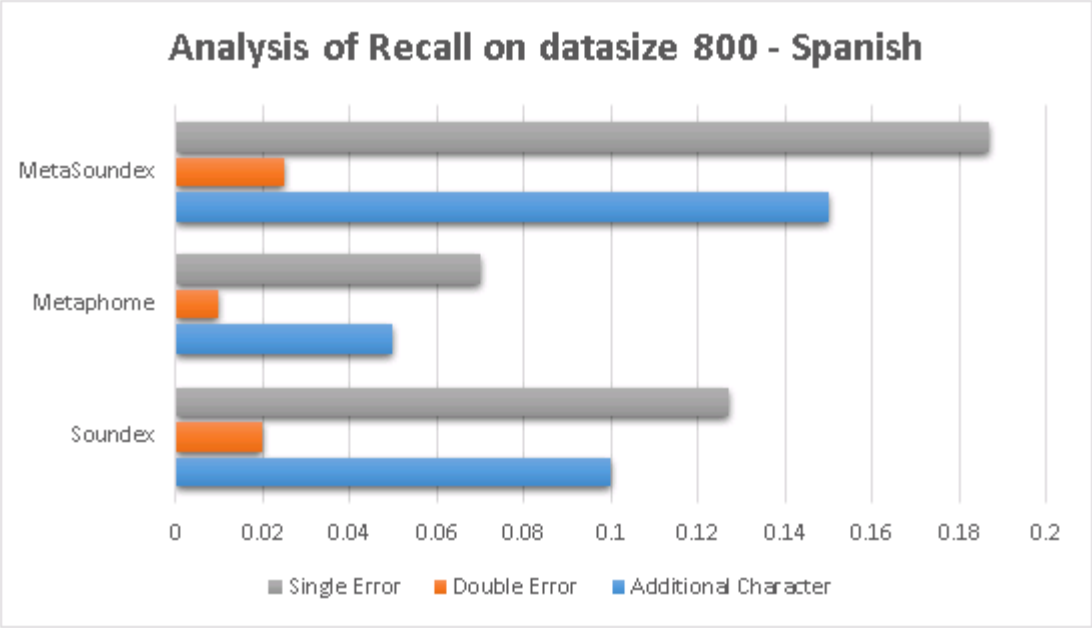

Figure 15. Recall for different techniques on synthetic Spanish dataset of size 800.

From the above figures it can be clearly observed that the state-of-the-art Meta-Soundex algorithm has highest accuracy, whereas, Metaphone has the lowest accuracy of all the algorithms. The figures also noticeably depict that the recall value is highly dependent on the type of error. The recall value is high for the erroneous wordlist having single error, while, it is low for the wordlist having two errors in each word. Apart from Meta-Soundex, the Soundex shows its high recall value in the second place, followed by Metaphone.

F-measure. Figures 16, 17, and 18 represent the F-measure of various techniques for data sizes ranging from 200 to 800 for Spanish language.

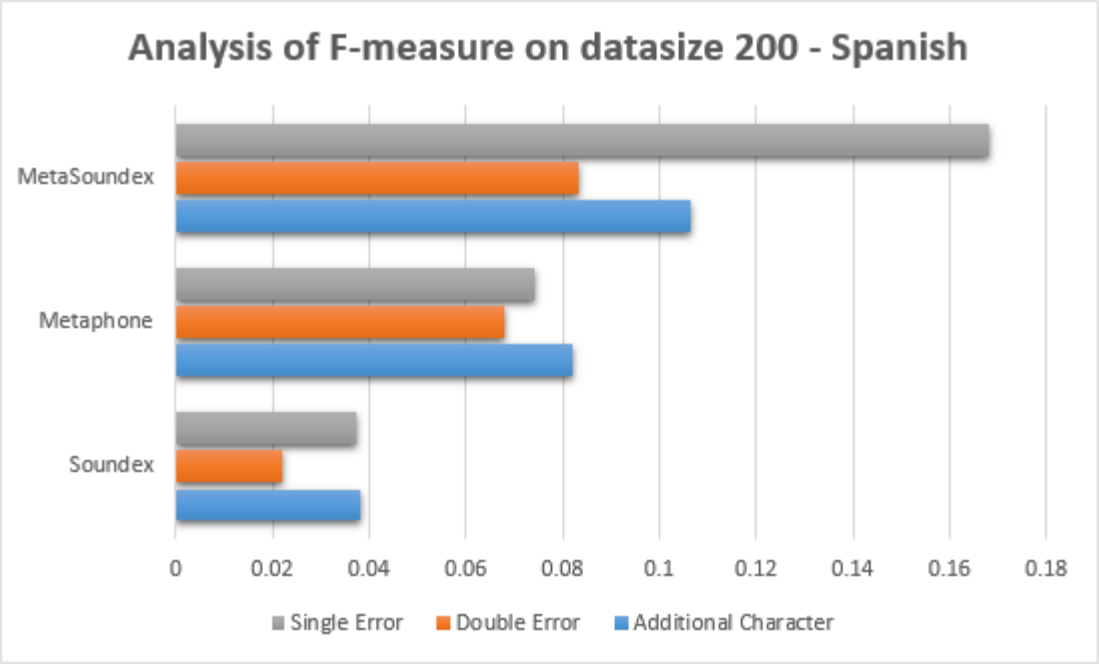

Figure 16. F-measure for different techniques on synthetic Spanish dataset of size 200.

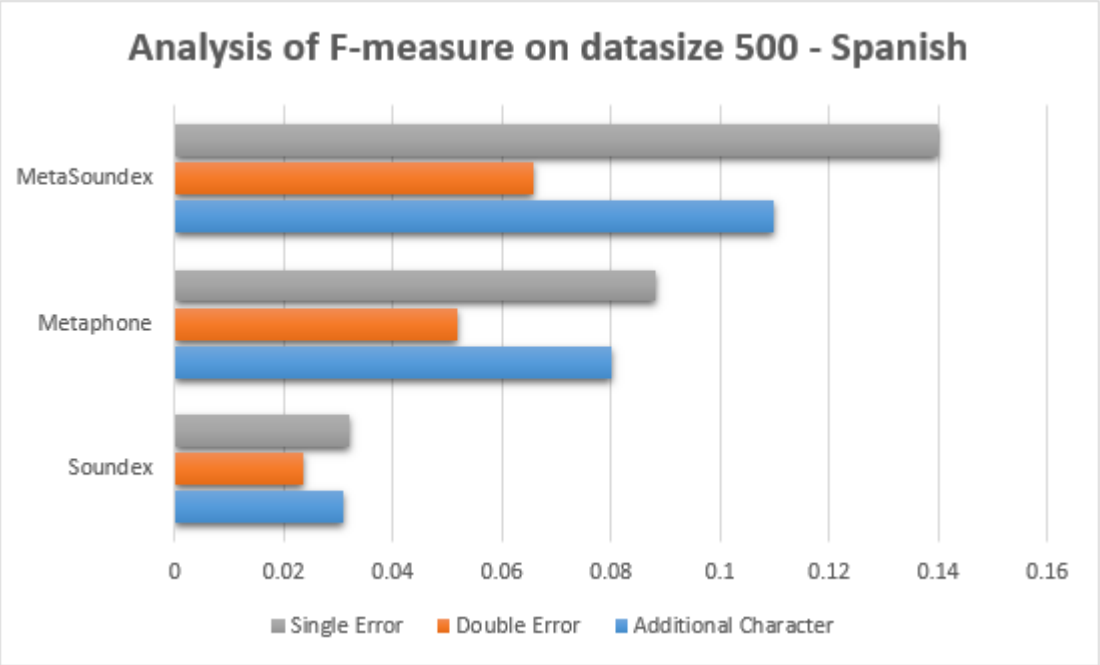

Figure 17. F-measure for different techniques on synthetic Spanish dataset of size 500.

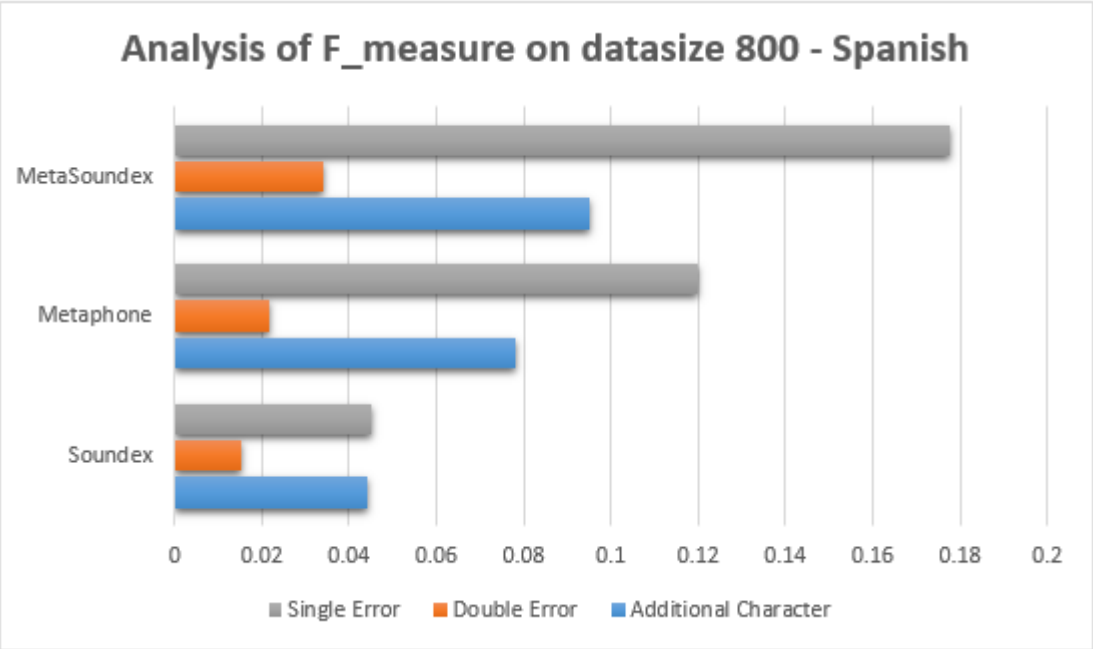

Figure 18. F-measure for different techniques on synthetic Spanish dataset of size 800. The figures indicate a substantial increase in the performance of Meta-Soundex algorithm over Soundex and Metaphone. Though Metaphone has high precision, due to its low recall, it has less performance than Meta-Soundex. Meta-Soundex has good performance along with the high recall value ensuring that the algorithm reduces noise and can be used in various applications where count of false positives play major role. All the three algorithms show least performance for the words with double errors irrespective of size of the dataset.

Analysis on real-world data - Spanish

As there is no intuitive research in the area of correction of misspelled words in Spanish language, very less real world misspelled data is available. The existing algorithms along with the state-of-the art Meta-Soundex algorithm, were executed on the real world data, which produced the following results as shown below.

Recall. The recall of different techniques obtained from the analysis are shown in Figure 19. The size of the dataset is nearly 100.

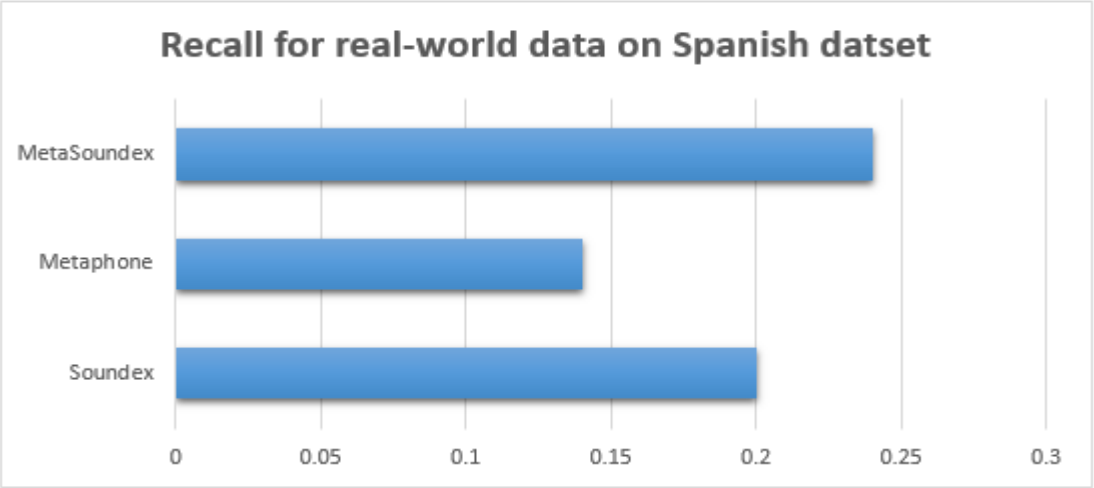

Figure 19. Recall for different techniques on real-world Spanish dataset. Size of dataset is 100.

From the above, it can be observed that the Meta-Soundex has the highest recall value showing its high accuracy on the real world-data, which is followed by Soundex, while Metaphone has the lowest accuracy rate in correcting the misspelled words.

F-measure. Figure 20 shows the F-measure for different techniques on a Spanish real-world dataset.

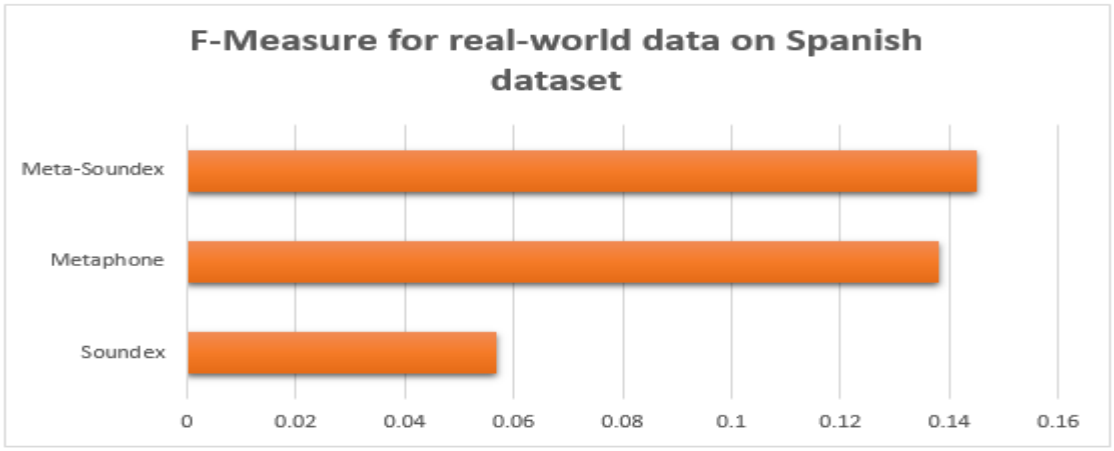

Figure 20. F-measure for different techniques on real-world Spanish dataset. Size of dataset is 100.

On real-world data for Spanish language, Meta-Soundex has the highest performance compared to other algorithms by reducing the unnecessary suggestions. In spite of its high precision, Metaphone has less performance due to least recall value. Soundex has the least performance as the precision is very less compared to other algorithms on the real world data.

CHAPTER VIII

Phonetic Matching Toolkit

Though different phonetic matching algorithms exist over decades, as per the research, there is no substantial phonetic matching toolkit available. This project is primarily intended to support researchers to have an integrated toolkit for various algorithms of English and Spanish languages. It also includes implementation for systematic evaluation of performance on test data. It is not intended for use on very large data sets.

Architectural design of phonetic matching toolkit

The design of phonetic matching tool (PMT) consists of a language selector and spell checker. Input is provided by text box. The architectural design of the toolkit is as shown in the Figure 21.

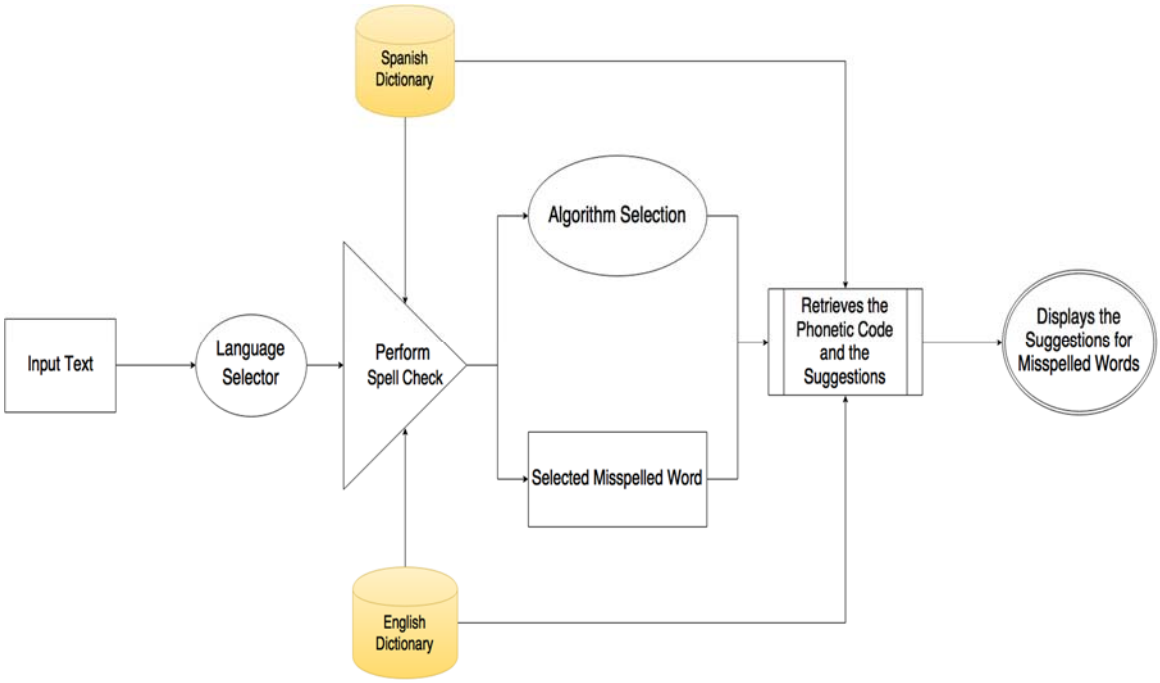

Figure 21. Architectural design of phonetic matching tool kit.

When the input data is entered, spell check is performed by considering the reference dictionaries as per the selected language. The misspelled words are populated into a drop-down list. Based on the language selection, algorithms are stacked into algorithm selector. For English language, six algorithms namely, Metaphone, Caverphone, DMetaphone, NYSIIS, Soundex, and Meta-Soundex are implemented, whereas for Spanish language three algorithms, namely, Soundex, Metaphone, and Meta-Soundex are implemented.

When a misspelled word, language, and the required algorithm is selected, the near matches for the misspelled word are generated and shown on the screen.

Experimental Design

The experimental design of the phonetic matching toolkit is shown in Figure 22. The web toolkit comprises of links for retrieving suggestions to misspelled words and performance evaluation of algorithms. Apart from that, the tool also includes the links to the dictionary words used in this application.

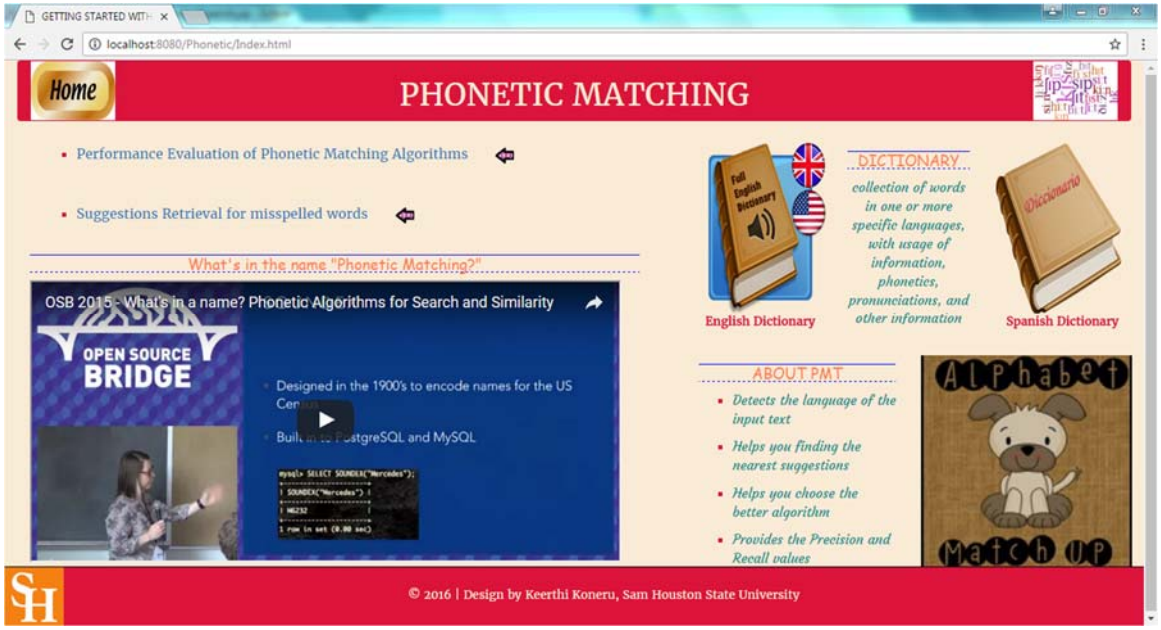

Figure 22. Phonetic matching toolkit.

Performance Evaluation. When the user clicks on the performance evaluation link, he/she will be redirected to a web page, which asks to select the language and upload the reference file and incorrect file to evaluate the accuracy and efficiency of each algorithm as shown in Figure 23. The reference file and the incorrect file should have the extension of either .csv or .txt as shown on the screen.

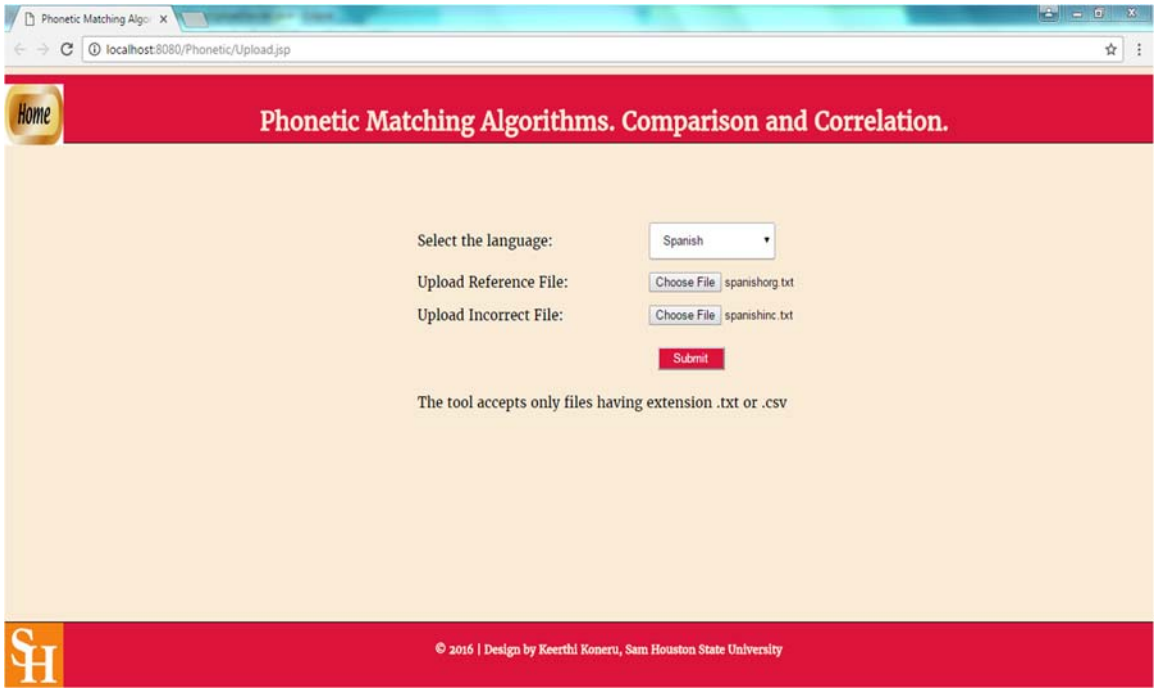

Figure 23. Webpage for uploading input files.

After the input files are submitted, different algorithms are executed on the misspelled words in the incorrect file for the corresponding selected language. The result analysis displays the recall, precision and F-measure of all the executed algorithms as shown in the Figure 24. After the analysis, two pdf files are generated, one with the corrected words and other with the suggestions for misspelled words for all the algorithms. The path of the generated pdf files is displayed on the screen as shown in the figure below.

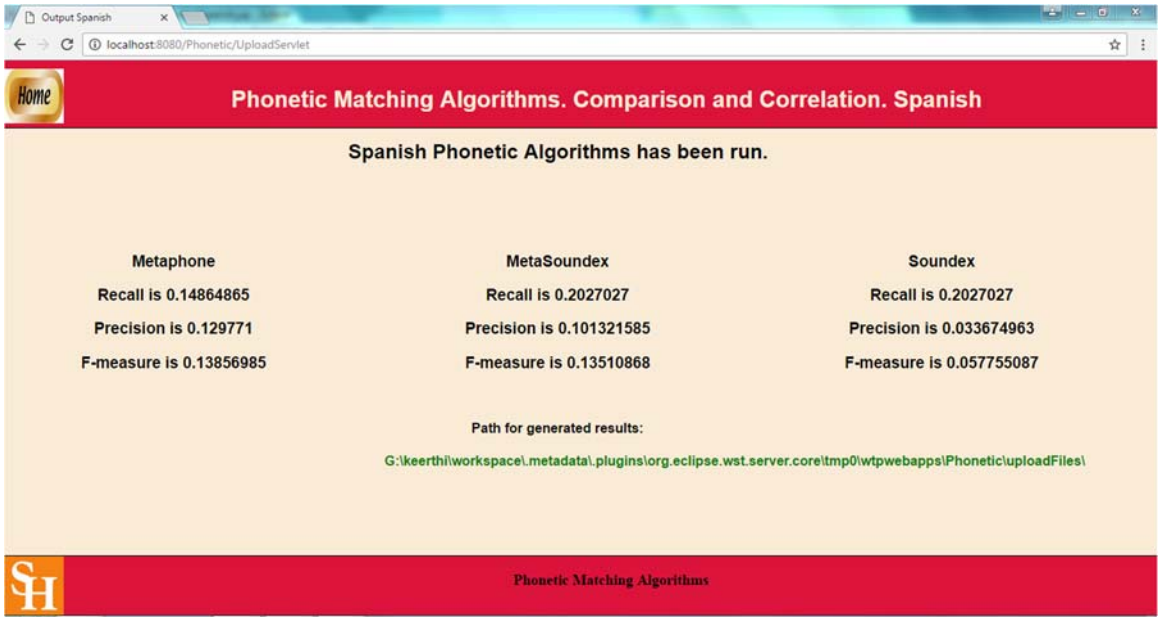

Figure 24. Comparison of precision, recall and f-measure of Spanish phonetic algorithms.

The user can be redirected to the home page by clicking on the home button present at the left top corner of the web page.

Suggestions Retrieval. On the home page, when the user clicks on suggestions retrieval link, he/she will be redirected to a page as shown in Figure 25.

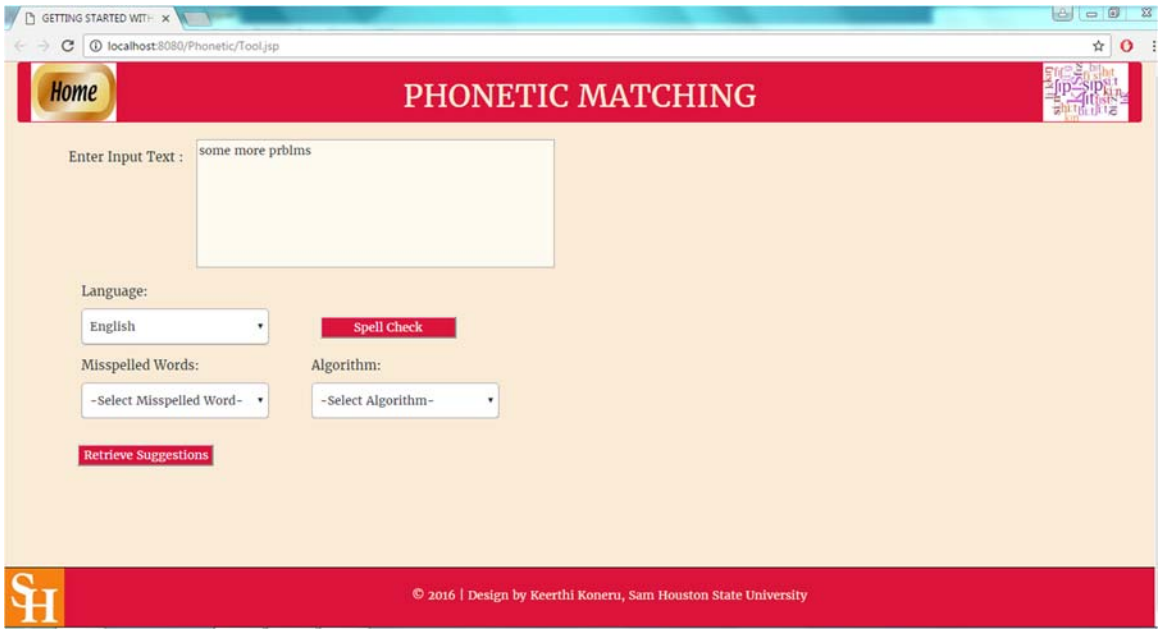

Figure 25. Suggestion retrieval webpage of phonetic matching toolkit.

The user can enter the input data in the provided text box. When the spell check button is clicked the misspelled words are loaded into the drop-down. After the selection of desired algorithm and the misspelled word, the resultant suggestions along with the generated code for the specified algorithm are shown on the web page. Figure 26 shows the resultant output.

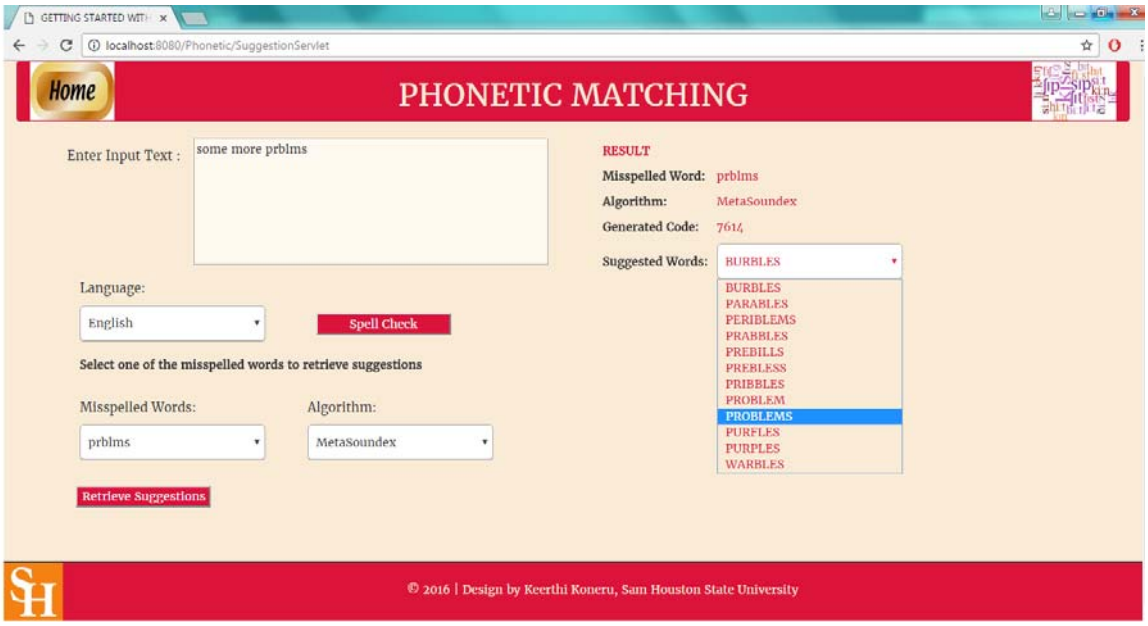

Figure 26. Screenshot showing suggestions for selected misspelled word.

CHAPTER IX

Summary and Remark

In this chapter, we discuss the summary and findings of this research. In addition, the limitation and future directions of this research will also be discussed.

Summary

In this project, we presented an overview of various phonetic matching algorithms in English and Spanish languages. We explained how newly developed Meta-Soundex algorithm is different from the existing phonetic matching algorithms. Then the functionality of different phonetic matching algorithms for both English and Spanish language are illustrated. Then, we justified the need to implement the state-of-the-art Meta-Soundex algorithm. The main purpose of the proposed approach is to improve the recall and precision over other algorithms, thus increasing accuracy and reduce the noise in retrieved suggestions for misspelled words from various sources.

The implementation of the Meta-Soundex algorithm is mentioned in detail. The performance of the proposed algorithm is evaluated on the datasets having three types of errors, namely, additional character, single error (substituted letter, missing of a letter) and words with double errors (more than one single error) along with the real-world data sets.

Apart from the development of new algorithm, a toolkit is also developed which incorporates all the algorithms into a single unit for both English and Spanish languages to retrieve suggestions for misspelled words.

Limitations

We specified a fixed data file type for the input files to evaluate the performance of algorithms. This can be considered as a limitation which requires some data processing to store data in these formats. Also, the processing time of the performance evaluation is very high for the large datasets.

Future Work

It is known that the analysis is performed only on the English and Spanish languages as both of them are most widely spoken languages across the globe (Most Widely Spoken Languages in the World, 2014). The development of phonetic matching algorithms can be extended to other languages based on the requirement, which can be considered as future work as it would provide more observance.

Meta-Soundex algorithm involves implementation of distance factor to improve the precision over other phonetic matching algorithms. The distance factor can also be applied on the other phonetic matching algorithms to improve the overall processing time. As a result, large datasets can also be evaluated which can improve their data quality.

This thesis can also be extended to take data in any format as well. This extension of the toolkit will give more flexibility in terms of time. Also, the extension of web tool kit into a standalone application can help the users to access the performance and obtain the data more efficiently without any need for online server.

Also, to improve the processing time of the performance evaluation, using Apache Spark would be a better choice rather than using a dedicated database (Justin, 2015). The Spark can store the retrieved results in Spark cache and can be used for

further analysis and comparison with reference file instead of retrieving the data from database more than once.

REFERENCES

Amón, I., Moreno, F., Echeverri, J., 2012. Algoritmo Fonético Para Detección De Cadenas De Texto Duplicadas En El Idioma Español. *Revista Ingenierías Universidad de Medellín*, Vol. 11, No. 20 pp. 127 – 138

Angeles, P. M., Gamez, A. E., Moncada, G.J., May 2015. Comparison of a Modified Spanish phonetic, Soundex, and Phonex coding functions during data matching process. *International Conference on Informatics, Electronics & Vision (ICIEV)*, At Kitakyushu, Fukuoka, Japan

Balabantaray, R.C., Sahoo, B., Lenka, S.K., Sahoo, D.K., and Swain, M., May 2012. An Automatic Approximate Matching Technique Based on Phonetic Encoding for Odia Query. *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 3, No 3.

Beider, A., Morse, S.P., March, 2010. Phonetic Matching: A Better Soundex. Retrieved from: http://stevemorse.org/phonetics/bmpm2.htm

Bhattacharjee, A.K., Mallick, A., Dey, A., Bandypoadhay, S., September 2013. Enhanced Technique for Data cleaning in text files. *International Journal of Computer Science Issues*, Vol. 10, Issue 5, No 2.

Carstensen, A., September 2005. An Introduction to Double Metaphone and the Principles behind Soundex. Retrieved from: http://www.b-eye-network.com/view/1596

Chaware, S., and Rao, S., April 2012. Analysis of Phonetic Matching Approaches for Indic Languages. *International Journal of Advanced Research in Computer and Communication Engineering*, Vol. 1, Issue 2, April 2012.

Christen, P., December 2006. A Comparison of Personal Name Matching: Techniques and Practical Issues. *Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)*, pp. 290-294, December 2006.

Diccionario. Retrieved from: http://www.deperu.com/diccionario/

Hassan, D., Aickelin, U., Wagner, C., 2014. Comparison of Distance metrics for hierarchical data in medical databases. *International Joint Conference on Neural Networks (IJCNN)*, July, 2014

Haunts, S., 2014. Phonetic String Matching: Soundex. Retrieved from https://stephenhaunts.com/2014/01/17/phonetic-string-matching-soundex/

Hempel, B., Fuzzy_tools. 2014. Retrieved from: https://github.com/brianhempel/fuzzy_tools/blob/master/accuracy/test_data/sources/misspellings/misspellings.txt

Hobbs, S., 2006. New York State Identification and Intelligence System (NYSIIS) Phonetic Encoder. Retrieved from: http://www.dropby.com/NYSIIS.html

Hood, D., December, 2004. Caversham Project Occasional Technical Paper.

Kelkar, B.A., Manwade, K.B., June 2012. Identifying Nearly Duplicate Records in Relational Database. *IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS)*, Vol. 2, No.3

Kestelyn, J., 2015. Working with Apache Spark, Retrieved from: http://blog.cloudera.com/blog/2015/05/working-with-apache-spark-or-how-i-learned-to-stop-worrying-and-love-the-shuffle

Kukich, K., December 1992. Techniques for automatically correcting words in text. ACM Computing Surveys, Vol. 24, No.4

Lawler, J., March 1999, An English Words List, Retrieved from: http://www-personal.umich.edu/

Lawrence P., December 1990, Hanging on the Metaphone. Computer Language, Vol. 7, No. 12.

Most Widely Spoken Languages in the World, 2014. Retrieved from: http://www.infoplease.com/ipa/A0775272.html

Mosquera, A., February 2012. Phonetic Indexing with the Spanish Metaphone Algorithm. Retrieved from: http://www.amsqr.com/2012/02/phonetic-indexing-with-spanish.html

Odell, M. K., and Russell, R.C. Patent nos. 1,261,167 (1918) and 1,435,683 (1922)

Pande, B.P, and Dhami, H.S., August 2011. Application of Natural Language Processing Tools in Stemming. *International Journal of Computer Applications* (0975 – 8887) Volume 27– No.6

Philips, L., June 2000. The Double Metaphone Search Algorithm. Retrieved from: http://www.drdobbs.com/the-double-metaphone-search-algorithm.

Planeta Curioso, 2008. Las 20 palabras peor pronunciadas en español. Retrieved from: http://www.planetacurioso.com/2008/10/30/las-20-palabras-pero-pronunciadas-en-espanol/

SaiKrishna, V., Rasool, A., Khare, N., January 2012. String Matching and its Applications in Diversified Fields. *IJCSI International Journal of Computer Science Issues*, Vol. 9, Issue 1, No 1.

Shah, R., and Singh, D.K., February, 2014. Analysis and Comparative Study on Phonetic Matching Techniques. *International Journal of Computer Applications,* Volume 87 – No.9.

Singla, N., Garg, D., January 2012. String Matching Algorithms and their Applicability in various Applications. *International Journal of Soft Computing and Engineering (IJSCE),* Volume-I, Issue-6, January 2012.

Singh, V., Saini, B., December 2014. An Effective Pre-Processing Algorithm for Information Retrieval Systems. *International Journal of Database Management Systems (IJDMS)* Vol.6, No.6.

Smetanin, N., March 2011. Phonetic Algorithms. Retrieved from: http://ntz-develop.blogspot.com/2011/03/phonetic-algorithms.html

Snae, C., 2007. A Comparison and Analysis of Name Matching Algorithms. *World Academy of Science, Engineering and Technology. International Journal of Computer, Electrical, Automation, Control and Information Engineering,* Vol:1, No:1.

Soundex Coding. 2016. Retrieved from: http://www.jewishgen.org/InfoFiles/soundex.html

Varol, C., and Talburt, J.R., 2011. Pattern and Phonetic Based Street Name Misspelling Correction. *Eighth International Conference on Information Technology: New Generations.*

Zhang, S., Zhang, C., & Yang, Q. (2003). Towards databases mining: Pre-processing collected data. Applied Artificial Intelligence, 17(5–6), 545–561. DOI: 10.1080/713827180.

Zobel, J., and Dart, P., 1996. Phonetic String Matching: Lessons from Information Retrieval. *Nineteenth Annual International ACM SIGIR conference on Research and development in Information Retrieval*

APPENDIX

The working code of the project is available in the git hub of the author. The link to the code is:

https://github.com/keerthikoneru/Phonetic-Matching-Tool-Kit-with-State-of-the-Art-Meta-Soundex-Algorithm.

VITA

Keerthi Koneru

Education

Master of Science student in *Computing and Information Science* at Sam Houston State University, January 2015 – present. Thesis title: “Phonetic Matching Tool Kit with State-of-the-Art Meta-Soundex Algorithm.”

Bachelor of Technology (April 2012) in *Electronics and Communications Engineering*, V R Siddhartha Engineering College, Vijayawada, Andhra Pradesh, India.

Academic Employment

Graduate Teaching Assistant, Department of *Computer Science,* Sam Houston State University, August 2015 – present. Responsibilities include: assisting professors with the preparation and presentation of undergraduate lectures, grading, and tutoring.

Graduate Research Assistant, Department of Computer Science, Sam Houston State University, June 2015 – August 2015. Research activities include requirement analysis, designing, and coding software for data cleaning.

Publications

KONERU, K., PULLA, V., VAROL, C., “Performance Evaluation of Phonetic Matching Algorithms on English Words and Street Name: Comparison and Correlation”, 5th International Conference on Data Management Technologies and Applications (DATA 2016), July 24-26, 2016, Lisbon, Portugal.

Academic Awards

Outstanding Graduate Student Scholarship, College of Sciences Engineering and Technology, Sam Houston State University, Spring/Fall 2015 – 2016.

Office of Graduate Studies Scholarship, Office of Graduate Studies, Sam Houston State University, Spring/Fall 2015 – 2016.

Summer Stipend for Thesis Research, College of Sciences, Sam Houston State University, Summer 2016.

Excellence in Writing, Academic Success Center, Sam Houston State University, April 2016.

Work Experience

Java Developer Intern, Computerized Assessments and Learning, June 2016 – August 2016, Lawrence, Kansas. Responsibilities include application development, unit testing, requirement analysis, deployment.

Software Engineer, Tata Consultancy Services, Jun 2012 - Dec 2015, Mumbai, Maharashtra, India**.** Responsibilities include web applications development, requirement analysis, coding, deployment in production server.