# Metaphone-pt_BR: The Phonetic Importance on Search and Correction of Textual Information

Carlos C. Jordão[1] and João Luís G. Rosa[2]

[1] São Carlos City Hall
Department of Information Technology
São Carlos, SP, Brazil
carlosjordao@gmail.com
[2] University of São Paulo
Computer Science Department
São Carlos, SP, Brazil
joaoluis@icmc.usp.br

**Abstract.** The increasing automation in the communication among systems produces a volume of information beyond human administrative capacity to deal with on time. Mechanisms to find out the inconsistent information and facilitate the decision-making are required. The use of a phonetic algorithm (Metaphone) adapted to Brazilian Portuguese proved to be a valuable tool in searching for name and address fields for automatic decisions, increasing substantially the performance regular database queries could obtain in information retrieval.

## 1 Introduction

Today, each sector of society is constantly searching for improvements to its registration forms, in order to make them increasingly accurate. One of the tools used to minimize problems of inconsistency is the use of closed form questions, which allows the user to choose only one of a predefined set of options, such as "select box" and "radio buttons" used in HTML forms. This mechanism makes indexing and information exchange among systems easier.

However, there are fields and systems that need to work with textual data, such as names and addresses, regardless of the reason. This makes the cross-checking of information among different systems difficult, because the most common solutions to determine matching between two different registers are not suitable for these cases. The command SQL *like*, for example, cannot guarantee a good match of words, except for very simple variations. Even so, names and addresses have complex variations of spelling without being semantically different. So, manual programming becomes unfeasible, because of the phonetic variation that may occur, which needs to be taken into account when comparing two words.

## 2 Objectives

Since simple methods of textual comparison are not efficient, it is essential to study other mechanisms that could deal with the challenge of the phonetic

A. Gelbukh (Ed.): CICLing 2012, Part II, LNCS 7182, pp. 297–305, 2012.

variation. Algorithms of this category do at least one of the two options: create an index of similarity between two words or supply a simplified representation of the word.

Those algorithms that compare words, such as Levenshtein [3], demand a lot of processing because it is necessary to completely scan the database [1], comparing a word with every other word stored, for example, to find which words are closer to the one given for comparison. Thus, it is preferable to use simplified representation of words once the scan can be done at a low cost, for example, by comparing the simplified phonetic representation of the words. The individual phonetic conversion is normally a very simple process, so that algorithms such as Metaphone [5] and Soundex [4] do not need more than one *loop* to scan the word in order to create its representation.

A sample of 2,591,562 proper names from the database of beneficiaries of Brazilian government's social programs, available in its website, resulted in 8,799,513 words and 226,686 exclusive words, which represents an average of 3.39 words per name. Those names were taken in September 2008 with the purpose of building a real base to evaluate the impact of phonetic algorithms for searching names. Figures 1 and 2 show the name length distribution and the word length distribution, respectively, in the sample.

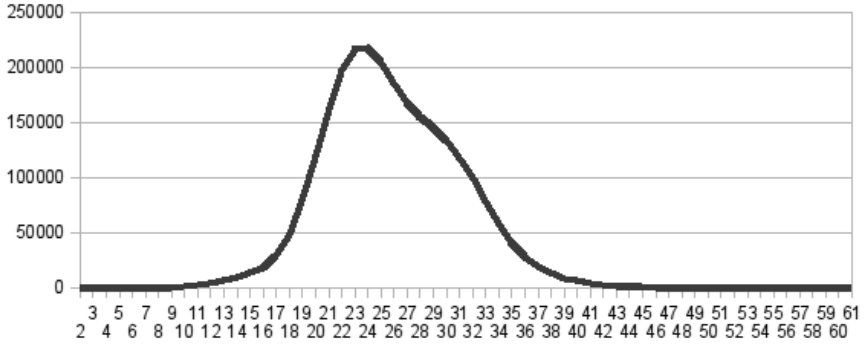


**Fig. 1.** Distribution of amount of names (y-axis) by word length (number of characters) (x-axis)

Considering that the complexity of the algorithm Levenshtein is $\mathcal{O}(n * m)$, where $n$ and $m$ are the lengths of the words to be compared, it is possible to calculate the necessary effort to scan through this mechanism by the average length of the stored words. In this case, other types of phonetic algorithms are very helpful.

Phonetic algorithms seek to build simplified representation of words, which can be seen as indexes for a database. Their objective is to find words that two people consider as equal or equivalent even if they were spelled differently due to the fact of phonetic context, by allowing these words to be clustered in several

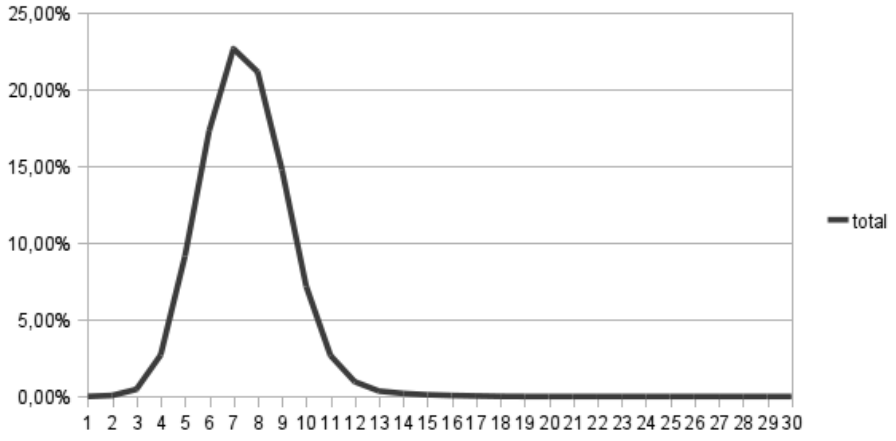


**Fig. 2.** Distribution of word percentage found (y-axis) by word size (number of characters) (x-axis)

clusters, according to their representation. Each algorithm, such as Soundex, Metaphone and Double Metaphone [6], produces different clustering results, in order to find a solution to several situations proposed.

In any case, each algorithm processes the word individually, so that its complexity for the worst case is of order $\mathcal{O}(n)$. This allows the representation to be stored in the database in advance, resulting in an index, which improves substantially the search time, minimizing the total computational effort necessary to come up with a smaller set of word records close to the one searched. Thus, the application of a second algorithm of comparison, even Levenshtein, becomes much more feasible. Sanae [8] shows that the best results come from hybrid methods, normally by using a phonetic algorithm with any other type.

Soundex, created by Rober C. Russell and Margaert K. Odell, patented in 1918, is the most famous phonetic algorithm, which inspired many other variations, with adaptation to foreign languages (it was created based on English phonetic rules). It was used for a retrospective analysis of the US censuses from 1880 to 1920 by the United States Census Bureau. Metaphone was created in 1990 by Lawrence Philips [5] as an alternative to resolve the deficiencies of Soundex. Later, the author released another version, called Double Metaphone, which returns two sequences of representations. The first one is called primary, the second, secondary. The primary is closer to the first Metaphone and to the English phonetic rules. The secondary works with a larger set of phonetic rules, such as Chinese and German.

But this alternative is not able to cluster efficiently words from the same language. This paper shows that the Metaphone version for Brazilian Portuguese produces better clustering results when used with the words of the main language than in the multilingual variant. The difference is the search for a method to compare the qualities of two algorithms, so that it is possible to measure the impact of the subsequent changes of the algorithm or variations.

## 3 A Brazilian Metaphone

The understanding of phonetic algorithms as searching tools and the intention of finding a way of searching names and addresses emphasized the need for implementing an algorithm for Portuguese phonetic rules during the project REDECA[1], which is a software aimed to help entities that take care of children and adolescents, exchanging information among themselves. Most of these entities keep a very poor register, consisting mainly of textual information with no possibility of associating one with each other automatically. Therefore, in order to help them to work as a whole, it was important to regulate the mechanism of registration and guarantee the non-existence of duplicated names in the database. This task soon proved to be a challenge. Since there are several different types of documents, each entity uses the one that best suits it. Also, not all of them have all kinds of documents, for many reasons, which makes the indexing of a name to a particular document not reliable. So that, it is necessary to check the entries by the name, in order to avoid duplicity in the database.

So, the Brazilian Metaphone was created to fill this need, since the existing solutions were not sufficient. Other authors have also chosen a similar approach [10] to use in their native languages, once they have found the same difficulties when applying algorithms to their daily activities. Since such algorithms behave as clustering algorithms, a poor classification, i.e. similar words in different clusters, would yield a bad search result [7,9]. Then, since this approach produced positive results, it was expected that it would be equally useful when applied to Portuguese language.

Metaphone has to be understood as an algorithm that excerpts the phonetic information from the consonants of the words. That is, when the vowels are taken out from the word, it is still possible for a person to recognize the essence of the original word in most of western languages. Consequently, the simplification method reduces to the maximum the number of characters that the final representation may have.

It is possible to correlate most phonemes used in the original Metaphone by following the Portuguese spelling rules with the addition of three new symbols for sounds not present in English pronunciation. For the representation of the rules on table 1, a mnemonic notation was used since many rules depend on the analyses of up to three prior or subsequent rules for a final decision of which phoneme will be used.

It is important to analyze how the choice of rules affects the word clustering, when converting phonetic rules. In Portuguese, for example, the consonant "L" sounds as the vowel "U" when preceded by any vowel. So, words like MAL (bad = not well) and MAU (bad = not good) have the same sound. Of 226,686 words, there are 33,682 that fit in this rule, which affect 1,364,712 (52,66%) of the analyzed names. In this case, it is better to consider the vowel as a consonant too, or unnecessary distinctions between words could be created.

[1] `http://www.softwarepublico.gov.br/ver-comunidade?community_id=18016032`

**Table 1.** Symbols used on rule phonetic mapping table (Table 2) of Brazilian Metaphone algorithm

| *symbol* | *meaning* |
|---|---|
| ^ | word beginning. |
| $ | word end. |
| [] | any characters inside the brackets. |
| v | any vowel (lower case letter 'v'). |
| c | any consonant (lower case letter 'c'). |
| . | any letter. |
| 0 | empty. It means that the character found was ignored (not mapped). |
| capital letters | specific vowel or consonant found. |

The sounds not present in English were mapped to the symbols "1", "2", "3", to represent the sounds of the (combinations of) letters "LH", "R" (voiced uvular fricative) and "NH", respectively. As a result of this work, tables 3 and 4 illustrate two word clustering that share the same phonetic representations, showing the similarity between them.

Finally, the main challenge in a language-dependent phonetic algorithm is to work with foreign names and its adaptation to the language usage, by adjusting the spelling of a foreign name to a local spelling. Therefore, being limited to lexical rules is also a problem because the algorithm would be helpful only for the dictionary words, but not for names and addresses, which suffer variations due to surnames and foreignness, which have consonant clusters unusual for the local language. This reinforces how far this complexity is from being solved by simple approach of comparison between name and database.

Thus, it is crucial to be able to exchange information among several systems through registration cross-checking of names and addresses in environments where the accurate correlation through numerical keys is not possible, as well as the correlation of information through address, for georeferencing.

The implementation of the algorithm can be found at SourceForge[2] under the GPL license.

### 3.1 Comparison between the Brazilian Metaphone and the Double Metaphone

The first experimental results with the algorithm were quite satisfactory, but it is necessary to measure how accurate it could be, having the secondary string of Double Metaphone as reference.

Since each phonetic algorithm has its own rules to create a phonetic representation of a word, it is not possible to directly compare the rules or the output of each word individually.

[2] `http://sourceforge.net/projects/metaphoneptbr/`

**Table 2.** Rules for the Brazilian metaphone algorithm

| *Letters* | *Phonetic representation (comments)* |
|---|---|
| ^v | v (repeats the vowel.) |
| B | B |
| $C[AOU]$ | K |
| Cc | K |
| C$ | K |
| $C[EI]$ | S |
| CHR | K |
| CH | X (this rule applies if the most specific does not match first.) |
| Ç | S |
| D | D |
| F | F |
| $G[AOU]$ | G |
| $G[EI]$ | J |
| $GH[EI]$ | J |
| GHc | G |
| ^Hv | v |
| H | 0 |
| J | J |
| K | K |
| LH | 1 (new sound) |
| ^L | L |
| Lv | L |
| vLc | c (keep last consonant) |
| M | M |
| N$ | M |
| NH | 3 |
| P | P |
| PH | F |
| Q | Kv |
| ^R | 2 |
| R$ | 2 |
| RR | 2 |
| vRv | R |
| .Rc | R |
| cRv | R |
| SS | S |
| SH | X |
| $SC[EI]$ | S |
| $SC[AUO]$ | SK |
| SCH | X |
| Sc | S |
| S | S (again, specific rules come first.) |
| T | T |
| TH | T |
| V | V |
| Wv | V |
| Wc | 0 |
| ^EXv | Z |
| $[MV]EX$ | X |
| $.EX[EI]$ | X |
| $.EX[AOU]$ | KS |
| $EX[PTC]$ | S |
| EX. | KS |
| $[vCKGLRX][AIOU]X$ | X |
| $[DFMNPQSTVZ][AIOU]X$ | KS |
| X | X |
| Y | I |
| Z$ | S |
| Z | Z |

**Table 3.** Words sharing the phonetic key $2BK$

| | | | | |
|---|---|---|---|---|
| ROBECA | REBCA | REBELC | REBOUCO | REBOLCA |
| RHEBECA | REBBEKA | RBECA | RABACO | REBOUCA |
| REBHECA | RABEKHA | REBEKAH | RABECO | REBEQUE |
| REBEKKA | REBBECA | REBELK | RHEBEKA | REBEKA |
| REBEECA | RUBICA | REBEK | REBEKHA | REBECA |
| REBECAH | REBOLCO | RABECA | REBEQUI | |

**Table 4.** Words sharing the phonetic key $JLRD$

| | | | | | |
|---|---|---|---|---|---|
| GELIARD | GILIARDO | GILYARD | GELIARDE | GILIARDY | JILIARD |
| GILARD | GILLARDE | JILLIARD | GILEARDE | GILLIARD | JOLYARDE |
| GILEARDY | GILLIARDE | JULIARD | GILIARD | GILLIARDI | JULIARDE |
| GILIARDE | GILLIARDY | JULIARDI | GILIARDI | GILLYARD | JULIARDO |

On the other hand, the words grouped by each Metaphone word are expected to be as homogeneous as possible, even if the quantity and the length of each one vary according to the algorithm used.

Therefore, to measure this uniformity, an algorithm of similarity was applied between the words of each cluster, calculating homogeneity by the average of the figures within that cluster.

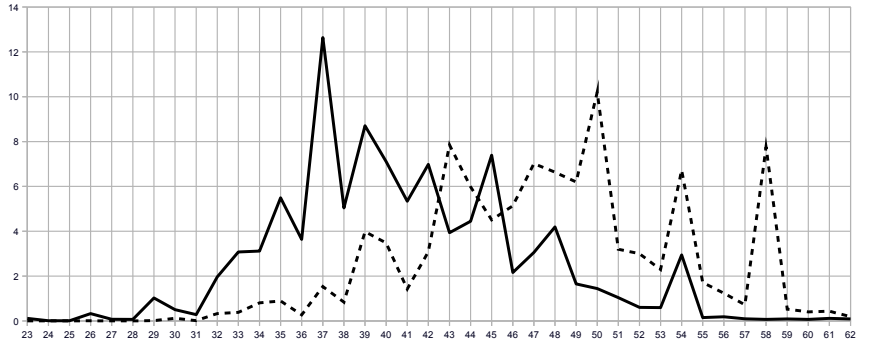


**Fig. 3.** Comparison between Double Metaphone (filled line) and Brazilian Metaphone (dotted line). The x-axis is the homogeneity percentage and the y-axis is the cluster counting percentage.

This process was made for the following algorithms of similarity: Levenshtein, Q-gram [12], Jaro [2], Jaro-Winkler [11], Similarity (implementation of trigraphs for PostgreSQL). Each of these algorithms has its own particularities, so a comparison using several of them would minimize those differences, as each of them returns a number between 0 and 1 representing the percentage of similarity. After being applied to all, the average for the cluster was calculated from the averages obtained with each algorithm.

The final result of the amount of clusters per similarity index is shown in Figure 3. The weighted average of each distribution is 40.8% and 47.9% for Double Metaphone and Brazilian Metaphone, respectively. That demonstrates that the expectation of the Brazilian algorithm resulting in a more homogeneous cluster is around 17% better than the Double Metaphone. Naturally, this figure is just a reference since it varies very much according to the similarity algorithm used. In the present experiment, the largest variation obtained was with Jaro-Winkler.

## 4 Conclusion

Considering that there is no previous formal specification for Metaphone to Brazilian Portuguese language, this work not only provides a new one, but also shows how this specification is better then the main Metaphone algorithm, through the analyses of more than 2 million names.

The qualitative comparison is important to lead the experiments with rule variations in the algorithm itself, as well as in other similar ones, to verify how it affects the way the words are clustered, impacting in the textual searches, specially for its more immediate application, that is, to find similar names and addresses, albeit spelled differently.

**Acknowledgments.** João Luís G. Rosa thanks Fapesp - Fundação de Amparo à Pesquisa do Estado de São Paulo, Brazil, for the research support under project number 2008/08245-4, with which this paper is associated. Also, the authors would like to thank the anonymous reviewers for their constructive criticism and useful suggestions, and their families for the unconditional support.

## References

1. Freeman, A.T., Condon, S.L., Ackerman, C.M.: Cross linguistic name matching in english and arabic: a "one to many mapping" extension of the levenshtein edit distance algorithm. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, NAACL 2006, pp. 471–478. Association for Computational Linguistics, Stroudsburg (2006), http://dx.doi.org/10.3115/1220835.1220895
2. Jaro, M.A.: Advances in Record-Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American Statistical Association 84(406), 414–420 (1989), http://dx.doi.org/10.2307/2289924

3. Levenshtein, V.: Binary Codes Capable of Correcting Deletions, Insertions and Reversals. Soviet Physics Doklady 10, 707 (1966)
4. Odell, M.K., Russell, R.C.: U.S. Patents 1261167 (1918), 1435663 (1922)† (1918/1922), cited in Knuth (1973)
5. Philips, L.: Hanging on the metaphone. Computer Language 7(12) (1990)
6. Philips, L.: The double metaphone search algorithm. C/C++ Users Journal 18(5) (June 2000)
7. Piltcher, G.: Correção de palavras em chats: Avaliação de bases para dicionários de referência. In: Anais do XXV Congresso da Sociedade Brasileira de Computação, pp. 2228–2237 (2005)
8. Sanae, C.: A comparison and analysis of name matching algorithms. Proceedings of World Academy of Science, Engineering and Technology 21, 252–257 (2007)
9. Snae, C.: A comparison and analysis of name matching algorithms. International Journal of Applied Science. Engineering and Technology 21, 252–257 (2007)
10. UzZaman, N., Khan, M.: A double metaphone encoding for approximate name searching and matching in bangla. Computational Intelligence, 108–113 (2005)
11. Winkler, W.E.: String comparator metrics and enhanced decision rules in the fellegi-sunter model of record linkage. In: Proceedings of the Section on Survey Research, pp. 354–359 (1990)
12. Zobel, J., Dart, P.W.: Phonetic String Matching: Lessons from Information Retrieval. In: Frei, H.P., Harman, D., Schäble, P., Wilkinson, R. (eds.) Proceedings of the 19th International Conference on Research and Development in Information Retrieval, pp. 166–172. ACM Press, Zurich (1996), http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.18.2138