A New Classical Arabic Soundex Algorithm

Nedjma Ousidhoum\textsuperscript{1}, Asma Bensalah\textsuperscript{1}, and Nacéra Bensaou\textsuperscript{1}
\textsuperscript{1}University of Science and Technology Houari Boumediene Algiers, Algeria
Email: nedjma.ousidhoum@gmail.com
Email: bensalah.asma1@gmail.com
Email: nacera_bensaou@yahoo.fr

Abstract—Search algorithms play an indispensable role in information retrieval. In this paper, we attempt to define a new useful indexing and search algorithm for Classical Arabic language using Arabic phonetics’ specificities. Our algorithm is based on the classical Soundex algorithm. In fact, we investigate Classical Arabic phonetic features toward words’ phonemes adequate representation. Thus, the algorithm proposed would create a suitable environment for a relevant information retrieval. It would be useful in creating phonetic dictionaries and in coding queries for an effective preprocessing in a spelling corrector.

Index Terms—Spelling correction, Soundex, Classical Arabic, phonetics, natural language processing, information retrieval, search algorithms, search engines

I. INTRODUCTION

Two words in a specific language could sound the same and be spelled differently, which led to the emergence of phonetic encoding algorithms.

Phonetic encoding algorithms take into consideration a word’s phonetic transcription for classification and coding purposes, such as correcting eventual spelling mistakes and classifying phonetically digital libraries, dictionaries and databases.

The most famous phonetic encoding algorithm is Soundex. First, Soundex was used to code surnames, in American census, based on the way they sound rather than the way they are spelled [3]. The algorithm keeps the surname’s first letter and uses three digits to represent the rest of the surname’s letters [2]. Besides, Soundex had been extended for several languages and numerous Soundex’s improvements had been developed, such as: Phonex [5], Metaphone [7], Double Metaphone [6], NYSIIS [12], German and Slavic Soundex [11], French Soundex variants [6], etc...

For the Arabic language, some Soundex functions had been suggested.

In 2006, as an English Soundex’s adaptation for Arabic, a Soundex function had been proposed [9]. The authors started by classifying the Arabic letters into eight categories. These categories correspond to a six human sounds’ classification, enriched with two new classes specific to Arabic. Nevertheless, the authors took into account Arabic local pronunciation’s particularities.

Another Arabic Soundex based on finding equivalent letters’ classes to those defined in the English Soundex was suggested [1]. This function, only defined for Arabic names, categorized Arabic letters into eleven classes.

In 2010, another phonetic function similar to Soundex [10], but considering different evaluated phonetic criteria, was developed in order to be used in a spelling corrector. This latter one, was designed especially to correct common errors made by non-native Arabic learners.

In the following work we present a new Arabic Soundex function that examines the Classical Arabic sounds’ classification mentioned in [4].

II. PRELIMINARY DEFINITIONS

Definition 2.1

The alphabet \( \Sigma \) is a non-empty set composed of letters. Given an alphabet \( \Sigma \). A word \( w \) defined in \( \Sigma \) is a suite of letters. Such as : \( w=c_{1}c_{2}...c_{n} \) with \( 0<i<n \) and \( c_{i} \in \Sigma \). \( \Sigma \) is the set of words over that alphabet. The empty word is noted \( \epsilon \).

Definition 2.2

A dictionary \( D \) is a set of words \( w_{1}, w_{2}, w_{3},..., w_{n} \) such as \( w_{i} \in \Sigma \), which supports the research operation.

This operation returns the query’s position in the dictionary.

III. ARABIC SOUNDS’ CLASSIFICATION

A Soundex function assembles words having similar pronunciation. It unifies letters articulated similarly by according the same code to them.

A classical Soundex function engenders a high number of false positives and false negatives. It deletes mute letters and must notice a language’s words’ particular cases. Indeed, the more a Soundex supports a language’s particular cases, the better it is.

In Arabic, mute letters don’t exist; the word is exactly read the way it is written.

Actually, Arabic letters’ sounds are essentially divided into five main categories. Every category gathers the Arabic letters that are delivered from a same area:

A. Abdominal category

which contains the Arabic long vowels ا, او, او, او, او, او.

B. Dental category

which contains ق, ك, ع, ج, ض, ح, م, ن, ر, ط, د, ب, ضت, طت.

C. Velar category

which contains ئ, ئ, ع, ح, ع, ح.
D. Labial category
  which contains “م”.

E. Nasal Category
  which contains “the ghuna” sound (a typical Arabic sound).

These phonetic categories include sub-categories.

A phonetic sub-category symbolizes the smallest area from which the letter’s sound is delivered. Therefore:
- letters whose sounds greatly resemble have the same phonetic sub-category and automatically the same phonetic category,
- letters whose sounds moderately resemble have the same phonetic category without sharing the same phonetic sub-category,
- letters whose sounds do not resemble neither share the same phonetic sub-category nor are included in the same phonetic category.

IV. NEW ARABIC SOUNDEX ALGORITHM

Dental, Velar and labial categories would be the ones used in our Arabic Soundex algorithm.

These three main categories include into a total of fifteen phonetic sub-categories as presented in table I.

<table>
<thead>
<tr>
<th>Category</th>
<th>Code</th>
<th>Subcategory’s Code</th>
<th>Subcategory’s elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dental</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>ف،، و، ي غير المهنة</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>ص، ج، خ غير المهنة</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>ل، جه، ش، شن، شن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>م</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>ت، ب، ح، ن، ن悲剧</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>ر، ت، ن، ن، ن، ن، ن، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>م، ن، ن، ن، ن، ن، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>م، ن، ن، ن، ن، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>م، ن، ن، ن، ن</td>
<td></td>
</tr>
<tr>
<td>Velar</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>ج، ت، ن، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>م</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>م</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>م</td>
<td></td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>م، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>م، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>م، ن</td>
<td></td>
</tr>
<tr>
<td>Labial</td>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>م، ن، ن، ن، ن، ن</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>م، ن، ن، ن، ن</td>
<td></td>
</tr>
</tbody>
</table>

Arabic language possesses complementary diacritical marks that we do not consider in this algorithm.

Our Classical Arabic Soundex function $f$ receives as input a word and returns an integer representing its phonetic code.

A word’s coding steps are the following:
- blanks and spaces are deleted,
- long vowels are deleted,
- if two adjacent letters are identical, only one of the two is kept,
- to each word’s letter are associated two numbers:
  1. the first one corresponds to the letter’s main category’s code. It is represented by an integer $N$ of two bits, such as $N \in E = \{0, 1, 2\}$.
  2. the second one corresponds to the letter’s sub-category’s code. It is represented by an integer $n$ of four bits, such as $n \in S = \{0, \ldots, 10\}$.

Thus:

Given a word $w = w_{1}, \ldots, w_{n}$, $w = w_{1}, \ldots, w_{n}$.

The phonetic code generated $X$ can be used as a hash key for classifying and indexing purposes.

Example 4.1

Given a word $w$ (to incline). Our algorithm’s application on $w$ respects the following steps:

Example 4.2

Given a word $w$ (to write). Our algorithm’s application on $w$ respects the following steps:

Example 4.3

Given a word $w$ (to promise). Our algorithm’s application on $w$ respects the following steps:

V. EVALUATION

Previous Soundex functions proposed for Arabic would cause a large number of false positives. Eventually, they wouldn’t be considered as good classification methods because they are based on Arabic local pronunciation’s specificities.

However, our Classical Arabic Soundex function deals with the phonetic features of Classical Arabic as they are stated in academic works [4].

An Arabic word’s codification time via our algorithm is negligible and is of the order of 0.0000007s.

We implemented a spelling corrector and created phonetic Classical Arabic dictionaries by indexing more than 400000 Arabic words via our algorithm. This latter one, working as a
hash function by using an integer for representing each Arabic word, allowed us a direct access to the dictionary and, accordingly, reduced significantly the research time.

The algorithm’s application on an Arabic dictionary permits the gathering of phonetically similar words. For example:

\[ \text{حلاف، حلال، نان، نان} \] whose sounds are alike, are grouped in order to facilitate the correction in the case where the user would misspell a similar word, due to a confusion in its phonemes.

If the user misspells a word, the system calculates its phonetic code through our Classical Arabic Soundex Algorithm. Afterwards, the system searches for the words having the same query’s phonetic code and for those which have approximate phonetic codes in the database. These results are the eventual query’s corrections.

**Example 5.1**

The word كتب contains a mistake. After applying our algorithm on كتب, we find that \( f(كتب) \) is equal to 131457.

The system searches for words sharing the code 131457 with the word كتب. It finds out the results shown in the table below and suggests them to the user.

<table>
<thead>
<tr>
<th>Words with the same phonetic code</th>
<th>Number of different characters = 1</th>
<th>كتب, كتب</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of different characters = 2</td>
<td>كتب, كتب</td>
</tr>
<tr>
<td>Words with a different phonetic code</td>
<td>Number of different characters = 1</td>
<td>كتب, كتب</td>
</tr>
<tr>
<td></td>
<td></td>
<td>كتب, كتب</td>
</tr>
</tbody>
</table>

In the table II, the spelling correction’s results are sorted out in order to clarify the different corrections proposed by our system. By dint of our algorithm’s application, there exist two main kinds of possible corrections, which are:
- The words sharing the same phonetic code with the query: they can differ of one or two characters from it,
- The words whose phonetic codes differ from the query’s one: they differ of one character from the query.

**VI. Conclusion**

The typical phonetic characteristics of the Classical Arabic language, described in this article, make any attempt of extending the English Soundex without taking them into consideration an incomplete one.

Our research contributes to the extension of the Soundex phonetic encoding algorithm for Arabic. To our knowledge, this is the first Arabic Soundex algorithm that takes fully into account the typical phonetic features of the Classical Arabic language.

This work could be augmented to solve dialectal Arabic spelling problems.

**References**


