International Journal of Computer Applications (0975 - 8887)
Volume 172 - No.4, August 2017

Spelling Detection Errors Techniques in NLP: A Survey

Rasha Altarawneh
AlBalqaa Applied University
Prince Rahma Collage
Jordan - Alsalt

ABSTRACT

This paper reports the efforts of Arabic Spell-checker researches by providing a brief summary of proposed methods and techniques that explains how the spelling errors might be discovered in any entered text. It mainly unites two areas that are quite different in appearance, computer science and natural languages .The domain of this topic is limited because of the complex morphology it has compared with other Languages, and the variation of its stems and the similarity of the characters for this language. This poses a challenge for the researchers to concern about it.

Keywords

Spell-Checker , Natural Language, N-gram, Radix tree , Context text

1. INTRODUCTION

Arabic is the spoken language of 250 million people in the world, of which roughly 200 million are first language speakers and 50 million are second language speakers. Arabic, also, is an official language in 22 Arab countries, and it is the language of all Muslims regardless of their origin. AL-Quran ?The holy book of Islam? is written in Arabic language as well [8]. Developing systems to improve the interaction between humans and computers have always attracted the attention of researchers in many fields of computer science. Natural language-processing systems are considered to be one of the most important fields of investigation that reflects this interest. Thus, morphological analysis techniques form the basis of most natural language processing systems. These beneficial techniques are used in many applications such as information retrieval, text categorization, dictionary automation, text compression, data encryption, vowelization and spelling aids, automatic translation, and computer-aided instruction.
However, processing Semitic languages such as Arabic is not an easy task because of their non-concatenative nature. For instance, although the Arabic words seem to be formed by using concatenating morphemes, in fact, they are formed by using root pattern schemes. Morphologically speaking, Arabic language is a complicated and rich language where tens or hundreds of words can be formulated by using one root, a few patterns, and a few affixes. Moreover, Arabic shows a high degree of ambiguity and such ambiguity is clear in the following cases: the deletion of vowels and the similarity between the affixed letters and the stem or root letters. Consequently, morphological analysis usually affects other higher levels of analysis such as syntactical and semantic analyses.
Studying and evaluating Arabic Spell Checker analysis techniques is considered as a difficult task for many reasons which will be clear by the end of this article. Thus, the current article aims at providing an introduction to Arabic Spell Checker analysis techniques including basic definitions, and effectiveness measures. Furthermore, the main objective of this paper is to survey Arabic Spell checker analysis techniques including multi ideas of how errors can be found.

2. TYPE OF ERRORS

Some studies address errors as typing errors which are caused by keyboard skids they maybe:

—Ignorance of grammatical rules.
—Semantic similarity.
—Phonetic similarity.

Other researchers defined simple errors as words that differ from the original word by single character. These errors could be a result of four operations:

—Deletion: One of letters missed .
—Insertion: Type an extra letter in the word.
—Substitution: Replace any letter with another one.
—Transposition: Swapping two adjacent letters in the word.

Real word errors are further sub classified in the literature. There are classified into these subclasses:

—Structural errors .
—Pragmatic errors .
—Syntactic errors .
—Semantic (Real word) errors

3. CORPUS CREATION

Mostly Spellchecker researches call for creating an Arabic Corpus which may include words from a specific field or more than one field. To achieve this stage many attempts to collect a large number of words will be presented in this section.

Muaidi and Altarawneh corpus (hereafter Muaidi Corpus) is implemented and compiled by the first author at De Montfort

University in UK . The corpus consists of 101,987 word types. It is implemented by a visual basic tool that browse daily newspapers and articles via web site [6].

Al-Jefri and Mahmoud A large corpus was collected from Al-Riyadh newspaper articles on three topics, namely health, economics and sports with a total number of (4,136,833), (24,440,419) and (12,593,426) words each, taken from (7,462), (49,108), (50,075) articles, respectively. This sums up into a general corpus which is composed of (41,170,678) words. In their model a sample of confusion set is organized from non-native Arabic speakers and an Arabic OCR system[1].

Shaalan et al created an AraComplex Extended corpus that is filtered from accepted words which are normalized by removing diacritics, numbers, symbols, punctuation marks and English Letters then are passed through a Microsoft spell checker to make a list of 9,306,138 words. Hence, it is considered as the largest corpus for Arabic spelling detection and correction[9].

Another modelling proposed by Zamora associates each word in the dictionary with its alpha-code (consonants of the word). Hence, it raised the need for having two dictionaries:

—One for the words.

—The other for their alpha-codes.

Therefore the correction will be done by comparing the alpha-code with the misspelled word. This method is efficient for permutation errors cases[7].

4. ERROR DETECTION

In this section, a review of the proposed Classifications of Arabic Spellcheckers techniques found in the literature is given by providing a description of each technique. Also, a comparison and a brief judgement for each Method are given at the end of this section.

4.1 Proposed Classifications

Khaled Shaalan tried to develop a tool for Arabic spellchecker. This tool is built using SICStus Prolog in IBM pc and the interface is built using Microsoft Visual Basic. The presented spellchecker consists of Arabic morphological analyser, lexicon spellchecker and corrector. The morphological analyser is very simple therefore the lexicon spellchecker is built to store and look up words during running time. The lexicon spellchecker consist of two types of lexicon: the base lexicon which includes the words that have no root, and the Stem lexicon which includes the words that have stem. Both types of lexicons were built through providing a letter tree for each word which is represented as Prolog term. The spellchecker corrector correct each word by adding missing characters, replace incorrect characters, remove excessive characters or adding space to split words [5].

Haddad and Yaseen proposed a hybrid model for non-word detection and correction. The ultimate goal of this work is to develop a context-dependent syntax and semantic checker. Therefore, the proposed model is primarily based on morphological and morpho-syntactical knowledge. Actually, it proceeds from the view that there are certain consistent pattern that correspond to each Arabic root. Also, there are certain prefixes and suffixes that frequently occur with derivative words, non-derivative words, and Arabized words. The authors classified Arabic words into three categories: Derivative Arabic Words (i.e., words which have valid roots); Non-Derivative Arabic Words or Particles (i.e., non-inflectional word types such as pronouns, adverbs, relative nouns, and particles); and Arabized Basic Words. Hence, a word that cannot be classified under one of these categories is regarded as a non-word. The paper also proposed certain measures to fulfil the task of correction through locating errors and ranking the optimal correction candidates in Arabic derivative words. Two measures have been proposed which are mainly based on frequency of occurrence. They are Root-Pattern Predictive Value (RPV); and Pattern-Root Predictive Value (PPV). In addition, keyboard impact, phonetic similarity, and certain lexical features have been taken into consideration to improve the process of error detection and correction. However, since the root pattern relationship does not play a role in the non-derivative Arabic words and the Arabized basic words, such words are considered as stems and collected in the Knowledge base. Errors in non-derivative words are then corrected based on certain lexical features like morphographemic, N-gram, and morpho-syntactical rules[3].
Taha Zerrouki modified Aspell and Hunspell by adding some features to those two open sources spellchecker to work properly with the Arabic language. The modification includes: ignoring Arabic diacritics, internal change with use of infixes, and depending on suffixes and prefixes with the use of circumffixes. The spellchecker searches for the word by the first character in the prefix rule. If it finds the prefix, it will remove it. Then it searches for the stem. If it finds the stem, it searches for affix in the affix rule if there is an affix for the word[10].

[9] proposed two modellings:

—Direct Detection :
It checks an input text against a list of correct words. If the word exists within the list then it will be considered as a valid word, otherwise it will be considered as a wrong word.

—Detection through language modelling:
A language model is built to support the classification and validation of Arabic words through checking the existence of a given word in the word list.

[1] proposed two Approaches; Context word and N-gram.
These approaches can easily handle errors caused by widely used confusing words. But such approaches can only detect particular errors that are predefined in advanced in a form of confusion sets. In addition, though the experimental results of the techniques applied in this study show promising correction accuracy, it is not possible to compare the results of this study with those of previous works, since there is no benchmarking dataset for real-word errors correction for Arabic text. Obviously, not all errors are covered by the stated confusion sets in this work. Hence, an extension is recommended to increase the number of confusion sets to cover most of the detected errors.
[6] the authors developed an Arabic spellchecker which is depends on N-gram score. To achieve this purpose they built 11 matrixes to show the connection between the letters of each word. They used Muaidi corpus which is adapted from Muaidi PhD thesis. This corpus consists of 101,987 words. The reason for building only 11 matrix is that the longest word in Muaidi corpus consists of 12 letters therefore there are 11 connections between the letters of the intended word. Each matrix consists of 28 row and 28 column according to the number of Arabic alphabet. Each cell in the matrix has a value of (0, 1, or 2) depending on the connection between letters in a given word. If the two letters have connection and they are the last two letters in the word ; the value of cell will be 2. If

the two letters have connection but they are not the last two letters in the word ; the value of cell will be 1. Otherwise, the value of the cell will be 0. To search for word, the spellchecker divided the word into pairs of letters and search for each pairs in the matrix. The first pair will search into first matrix if the cell value is 0, this indicates that the word is wrong. If the cell value is 1, this shows that there is a connection and the word is not end. It, then, continues searching until the cell value is 2 which in turn mean that the word is found and it is right.

[2]proposed an Arabic spellchecker using Radix search tree. They used Muaidi corpus to build the spellchecker by taking each word in corpus and dividing it into letters. The first letter of each word represents the root of the tree and other letters connected to the root as children until the word letters finish. The leave node which represents the end of a given word has attribute (*) to mark as the end of word. The tree may has more than one branch that might share the same letters. To search for a given word, the spellchecker searches for the first letter of the word in the root of all trees then finds the second letter in the same tree. It, then, continues searching for other letters of the word until it finishes the whole word and finds the attribute that indicates that the word is right. If a given word is not found, it will be considered wrong.

[4] proposed an independent spellchecker corpus based on morphological analysis in the process and utilizing a stems dictionary to reduce the size of the huge amount of all Arabic words. Table 1 summarize the surveyed methods.

Table 1. : Summary Of Surveyed Methods

Author(s) and date	Required lists	Corpus	Models	Advantages	Disadvantages	Accuracy [%]
[5]	Two type of lexicon	Not required	-Morphology analyzer -Lexicon spellchecker corrector	No need for a huge corpus	Worked just with standard Arabic	Did not mention
[3]	-Sets of NA stems, common particales and feature structures	-Arabic root corpus -Arabized Basic words	A hybrid approach	Taking dialects in the spell-checker	Need to save all Arabic root	High accuracy and better than MS-Word Spell-Checker
[10]	Suffix and infix files are used	Need corpus of words verbs prepositions nouns	Depends on rules	-Develop open source program -reduce the size of Aspell dictionary	-Use large space for storage the rules	-Better than Huspell -test is in progress with Aspell
[9]	Not required	Ara ComLex Extended	- Direct Detection - Detection through language modelling	can integrated in text authoring tools	Need long time through Look up	Gives a precision of 98.2% at a recall of 100%.
[6]	Not required	Muaidi Corpus	N-Grams Scores	Reduced Searching time and storing space	Need a huge corpus contains all Arabic words	Reached to 98.99%
[1]	28 Confusion sets	(41,170,678) words a Largest corpus of Arabic	-Context words - n-gram language models	Can handle errors caused by common confused words in an easy way	Can only detect specific errors that are predefined in a form of confusion sets	Average accuracy rate of 95.9%
[2]	Not required	Muaidi Corpus	Radix Search Tree	Reduced Searching time and storing space	Need a huge corpus contains all Arabic words	It provides a high accuracy; reached to 100%
[4]	Need list of words	Stems dictionary	AraMorph	Reduced size	Did not discover all types of errors	Reached to 84%

5. CONCLUSION

This paper addresses the researches which attempt to solve the spell-checking problem for Arabic language. Each suggested method has an advantages and disadvantages and all of them solve the problem. There is may be effected to applied over other languages, and may be merged with each other to implement an optimal spell-checker program for arabic Language.

6. REFERENCES

[1] Majed Al-Jefri and Sabri Mahmoud. Context-sensitive arabic spell checker using context words and n-gram language models. 2013.

[2] Rasha AL-Tarawneh, Hatem S. A. Hamatta, and Hasan Muiadi. Novel approach for arabic spell-checker: Based on radix search tree. *IJCA*, 95:5, 2014.

[3] Bassam Haddad and Mustafa Yaseen. Detection and correction of non-words in arabic :a hybrid approach. *International Journal of Computer Processing of Oriental Languages*, 20, 2007.

[4] Bakkali Hamza, Gueddah Hicham, Yousfi Abdellah, and Belkasmi Mostafa. For an independent spell-checking system from the arabic language vocabulary. *(IJACSA) International Journal of Advanced Computer Science and Applications*, 5:4, 2014.

[5] AbdAllah Gomah Khaled Shaalan, Amin Allam. Towards automatic spell checking for arabic. 2003.

[6] Hasan Muaidi and Rasha Al-Tarawneh. Towards arabic spell-checker based on n-grams scores. *IJCA*, 53:5, 2012.

[7] JOSEPH J. POLLOCK and ANTONIO ZAMORA. Automatic spelling correction in scientific and scholarly text. *Communications of the ACM*, 27:11, 1984.

[8] Zeina Azzam Seikaly. The arabic language: The glue that binds the arab world. AMIDEAST.

[9] Khaled Shaalan, Younes Samih, Mohammed Attia, Pavel Pecina, and Josef van Genabith. Improved spelling error detection and correction for arabic. *Council for Science Engineering and Technology*, page 7, 2011.

[10] Amar Balla Taha Zerrouki. Implementation of infixes and circumfixes in the spellcheckers. 2009.