# Survey of Arabic Checker Techniques

Ahmed Abdalrhman Saty[1], Karim Bouzoubaa[2], Aouragh Si Lhoussain[3]

[1] Sudan University of Science and Technology, Sudan
[2,3] Mohammed V University in Rabat, Morocco
wdsaty@sustech.edu

*Received:28/09/2019*
*Accepted:06/11/2019*

***ABSTRACT***- It is known that the importance of spell checking, which increases with the expanding of technologies, using the Internet and the local dialects, in addition to non-awareness of linguistic language. So, this importance increases with the Arabic language, which has many complexities and specificities that differ from other languages. This paper explains these specificities and presents the existing works based on techniques categories that are used, as well as explores these techniques. Besides, it gives directions for future work.

***Keywords:*** *spell checking, rule-based, morphology, n-gram, radix-search tree, levenshtein distance, jaro-winkler distance*

***المستخلص*** *– من المعلوم أهمية التدقيق الإملائي، والتي تزداد مع توسع التقنيات، استخدام الانترنت واللهجات المحلية، اضافة إلى عدم الإلمام بقواعد اللغة. وبالتالي تزداد أهميته أكثر مع اللغة العربية نسبة لأنها تحتوي على بعض التعقيدات والخصائص التي تميزها عن اللغات الاخرى. هذه الورقة تستعرض بعض خصائص اللغة العربية، كما تقوم بعرض الأعمال الموجودة في هذا المجال بناء على التقنيات المستخدمة ومن ثم شرحها. علاوة الى ذلك تعطي بعض الاتجاهات للأعمال المستقبلية.*

## *INTRODUCTION*

With the increased usage of computers and smart devices in the processing of various languages, comes the need for correcting errors introduced at different stages. Texts of any language can be generated from different sources either by humans as document typing and emailing software, or by machine such as optical character recognition (OCR) and machine translation (MT). These produced texts may have typing mistakes that need to be spell checked and corrected. Spell checking constitutes one of the major areas in the field of Natural Languages Processing (NLP) and has been the subject of different research studies since 1960 [1]. Spell checking mainly consists of verifying that some typed words are not accepted in the used language and suggests a list of close words to the erroneous word. Accordingly, numerous approaches have been explored to correct spelling errors in texts using NLP tools and resources.

Some languages, such as English, developed advanced detection and spell checking systems. For the case of Arabic, such systems are double-needed with the rapid growth of the Arabic digital content and users (it is reported that, for 2017, 43.8% of the whole Arabic populations are Internet users [37]) and because of the specificity of many linguistic phenomenon that increase the probability of user mistakes such as multiplicity of local dialects and the non-awareness of Arabic linguistic rules.

An Arabic spell checker behaves exactly the same as an English one. For example, for the text "الزلد" (alzalad), the checker detects it as an erroneous word and suggests a list of close words such as "الزاد, الولد, الصلد, الزبد, الزلط" (azzad, alwalad, assalad, azzabad, azzlad).

In the context of Arabic spell checking, many approaches and methods have been studied. Multiple systems with different designs already exist. Some of them exploit dictionaries while others use morphological analysis [2]. A lot of them use also similarity among words [3, 4], and a few use the context [5] or mix between these techniques [6, 7].

This paper surveys the existing Arabic spell checkers with broad coverage of their advantages and disadvantages and consequently sheds light on new opportunities in order to improve these existing works. The remainder of the paper is organized as follows. Section 2 explains the specificities of the Arabic language needed in the context of spell checking. Section 3 introduces the classification of errors, explains the meaning of datasets with their types alongside used techniques in existing works, and

presents our notices on these works. Finally, we conclude the paper and make propositions about new ideas for improving the existing works for future work.

**Arabic Language Uniqueness**

Nowadays more than 400 million people in the Middle East and North Africa speak the Arabic language [38]. Arabic is also used as a religious language in the Islamic Word. Therefore, it is learned by various levels of proficiency, as a venerated, liturgical language[8] by many Muslims mainly in Asia (e.g., Pakistan, Malaysia, China) and Africa (e.g., Senegal) [39].

Arabic has its own alphabet, its own lexicon, its own morphology, and its syntactic rules. The alphabet is 28 letters and the morphology is very rich. As a Semitic language, Arabic is derivative and has a flexible syntax allowing, for example, both verbal and nominal sentences [9, 11]. However, it is the alphabet and lexicon specificities of the Arabic language that have a direct impact of producing errors while typing and that necessitate the use of spell checking systems.

The alphabet is written from right to left in a cursive style. Some letters have specific rules for writing leading their shapes depending on their position in the word. For example, the letter "ك" is written at the beginning and middle of the word different than the last. The hamza letter is written with 5 possible shapes depending on its position in the word as well as the diacritic of the previous and the next letter. These rules are not accurately known by all users causing them to make typing errors.

In addition, some letters have very close pronunciations such as ("د", "ض"). Consequently, people who are typing on computers are interchangeably using one letter instead of the other thinking that is the right way to spell the corresponding word.

On the other hand, and due to its derivative aspect, the Arabic lexicon is extremely large (about 12 million entries). However, Arabic citizens do not make use of this entire lexicon and have over the history "squeezed-minimized" the Arabic language and use a reduced lexicon that does not exceed almost 20,000 entries. Nowadays, from region to region Arabic citizens use their own dialects but with a change in the pronunciation of some letters. For instance, Egyptians replace the pronunciation of ("ق") with ("أ") while Sudanese replace ("ق") with ("غ") and replace ("ذ") with ("ز"). In these cases, users type the word using the letter with its dialect-pronunciation instead of the original letter.

According to the above Arabic language specificities (alphabet and lexicon), the origins of typing errors include:

1. Changing of shape according to the position of letters in the word.
2. Similarity of pronunciation and shape of some letters (see Figure1).
3. The use of dialects.

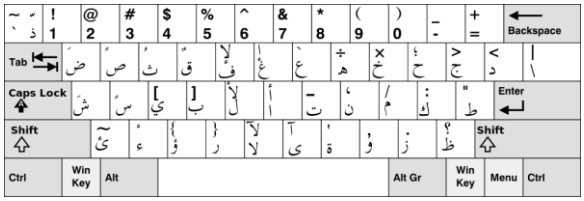


**Figure1: Arabic Keyboard**

Whatever the origins are, to assist Arabic users during their typing, many Arabic spell-checking approaches and systems have been developed. In the next section, we review these works.

**Arabic Spell-checking Issues**

In this section, we review the most important works about Arabic spell checking. Firstly, we introduce the classification of errors. Secondly, we explain the meaning of datasets used in the context of spell checking with their different kinds. Thirdly, we focus on the most used approaches and techniques which are in turn divided into five categories, which are: rule-based approach, distance similarity techniques, techniques exploiting morphological analysis, techniques relying on phonetics and finally hybrid ones, which combined more than one of the existing techniques.

**Classification of errors**

There are two main types of typing errors; the first one is isolated-words and the second one is context-sensitive [12, 13]. The first type detects that the misspelling words do not exist in the lexicon. To deal with this type, the researchers introduced many analyzing and statistical studies. Based on the user knowledge, the misspelling errors can be divided into three categories[11]: the first one is the typographic error; where the user is familiar with how to write the word but (s)he makes the typing error. The second category is the cognitive error; where the user is not familiar with how to write the words, due to the non-awareness of the Arabic rules. The third category is the phonetic error that takes place when there is a replacement of letters, due to the proximity of the sound. In general, the typing errors (approximately 80%) occur due to one of

the following reasons [14]: letter insertion, letter deletion, letter substitution and transposition of two adjacent letters.

The second type is context-sensitive and is also called real-words error. In this type, the written word is correct but its position is incorrect and leads to a wrong meaning [5, 15, 16]. Few Arabic spell-checking researchers addressed real-word errors. Among these works, the researchers of [5] proposed a spell checker with a large corpus collected from three topics (sport, health, and economics), as well as 28 confusion sets, that were collected from commonly confused words. Later on, the authors of [16] proposed a system that deals with context errors by applying n-gram and machine learning instead of predefined confusion sets approach. Furthermore, context-sensitive also can be used with the first type at the correction stage to get the proper suggestions for a non-word based on its position in the sentence such as the work of [17].

**Datasets**

The datasets, which are word lists, are an indispensable component of any spell checker. They mostly contain correct words and are used as a reference in order to detect wrong words at the detection phase. On the other hand, the correction phase uses them to candidate the suggestion words. Hence, when the dataset is large, the result is better. The dataset has many faces of using. It can be used as a dictionary (lexicon of language) such as "Alwassit Arabic Dictionary".

Also, the dataset can be used as a corpus containing a set of words in a particular field to support a specific checker. For instance, authors of [5] made a large corpus composed of (41,170,678) words collected from Al-Riyadh newspaper articles about health, economics and sports. A standard corpus can also be used such as QALB corpus (Qatar Arabic Language Bank) which is a large manually corrected corpus of errors collected from native and non-native speaker articles and machine-translation output [18].

**Arabic Spell-checking Techniques**

The Arabic Spell checking Techniques are divided to five categories of techniques as shown below:

**Rule-Based Techniques**

Rule-based is a set of rules containing a lot of instructions to perform a particular task. Its results are often taken as suggested words [12]. It is a very useful way to do something and arrange works. In spell checking, the rule-based approach is considerably used to handle common spelling and typographic errors. For example, authors of [19] proposed a system that has a mechanism for automatic correction of common errors in Arabic based on rules such as the dealing of hamza errors since there is a confusion between the dah "د"and zah "ذ", taa marbuta "ة" and yaa "ي". The mentioned errors are treated by applying regular expressions and word replacement list. Moreover, the works of [7, 20, 21] captured also various kinds of common errors. It is also noted that the use of rule-based techniques gives more satisfactory corrections.

In addition, the rule-based approach can also be used to rank the candidate words by aggregating the probabilities of applied rules [12]. The work of [22] applied A* lattice search and n-gram probability estimation for this purpose. As well, other rule-based approaches were used to deal with common errors. Besides, the authors of [23] used knowledge-based rules to get scores to the suggested words, then choose the best word regardless of the context. In general, the use of rule-based approaches makes it possible to develop spell checkers with good characteristics.

**Similarity Distance Techniques**

Similarity techniques are used to suggest close right words for erroneous words. There are multiple similarity techniques such as edit distance (Levenshtein distance), Jaro-winkler distance, Jaccard distance, TF-IDF, radix search tree, and n-gram distances. Most spell-checking studies mainly use the Levenshtein distance either by developing or integrating it with other distances in order to get an appropriate result. In the next paragraphs, we present some similarity techniques used in Arabic spell checking.

The first one is the Levenshtein distance, also called edit distance [40], considered as a simple technique. It is defined as the minimal number of editing operations (insertion, deletions, and substitutions) required to change the non-word to the right words existing in the dataset. See Algorithm 1. Levenshtein distance is suitable for correcting errors resulting from keyboard input but not for correcting phonetic errors [1].

As well, a spell checker using this distance alone has a limitation in the order of suggested words that have the same edit distance. Some works addressed this issue such as authors of [1] who introduced a new measurement of Levenshtein distance using the matrices frequency of the editing errors (insertion, deletion, and permutation). These matrices were created from a set of Arabic documents typed by four

experienced users. Moreover, authors of [17] added a weighting into the Levenshtein distance based on the n-gram language models.

**Algorithm 1: Dynamic programming algorithm for computing the edit distance [43] between strings $s_1$ and $s_2$, Edit Distance ($s_1$,$s_2$)**

```
1.  int m[i, j] = 0
2.  for i ← 1 to |s1|
3.  do m[i, 0] = i
4.  for j ← 1 to |s2|
5.  do m[0, j] = j
6.  for i ← 1 to |s1|
7.  do for j ← 1 to |s2|
8.  do m[i, j] = min { m[i - 1, j - 1] + if (s1[i] = s2 [j]) then 0 else 1fi,
9.                     m[i - 1, j] + 1,
10. m[i, j - 1] +1}
11. return m[|s1|,|s2|]
```

However, these proposed measures require huge corpus containing the largest number of words to give satisfactory suggestions. As well, the similarity and proximity between Arabic characters was considered in the work of [24]. Although it deals with the permutation errors, it also needs to be added later in order to deal with insertion and deletion errors.

The second one is the Jaro-Winkler distance considered as a development of the Jaro distance. It gives a better measurement between two strings because it accounts the similarity characters and the transposition letters in the two compared strings. It also uses a prefix scale that gives more favorable ratings to strings that match from the beginning for a set prefix length [41]. Furthermore, the output value of this algorithm is a real number belonging to the interval (0,1). Therefore, whenever the output tends to 1, this means there is a high similarity between the two compared strings. This distance is specifically used in the field of record linkage [25]. In Arabic spell checking, this distance is used only in the work of [3] combining it with the Levenshtein one to output a better order for candidate suggestions.

On the other hand, the radix-search tree is one of the search techniques where each letter of a word is represented by a node, in addition to labeling the last letter of any word to indicate the end of it. This method reduces the time of searching but needs more memory in order to represent a large dataset. The authors of [26] applied the radix-search tree approach to detect misspelling words in the detection phase without explaining what was used at the correction phase.

Finally, the n-gram technique is also used in spell checking. N-gram means n-letter subsequences of n-adjacent letters in a word (n = 1 refers to unigram, 2 to bigram, and 3 to trigram). The spell checker of [4] is based on bi-gram scores and uses a matrix approach (eleven matrices are built for the longest Arabic word that has 12 letters). Although the test results of this spell checker were good, it requires a large memory capacity to deal with the huge data. Also, the authors of [27] proposed a speech recognition system that corrects the erroneous words (specifically clear Arabic language and Iraqi dialect) using n-gram. On the other hand, this technique can be used as a language model (n-word subsequences of n-adjacent words in a document). This use is beneficial in spell checking either to detect a real-word error as the work of [5] or to arrange appropriate suggestion words as the works of [17, 28].

**Morphology Techniques**

Morphological analyzing is also used to improve the spell checking. In general, morphology studies the generation and analysis of words with their roots and stems alongside affixation. Using morphology helps in having quicker and more intelligent spell checkers such as [29]. Also, the authors of [30] introduced a lightweight system that uses derived words by surface pattern. Furthermore, the works of [23, 28, 31, 32] used a finite-state morphological transducer in their spell checker.

**Techniques relying on phonetics**

The spell checking needs to include the phonetic errors resulting from proximity and changing of some sound letters due to the expanding Internet, spread local dialects and moving people from countries to others. A few Arabic works that deal with this kind were found. Among them, the work of [33] captured the error mistakes for Egyptian dialects and the work of [5] considered the dictionary of Iraqi. Although they all used phonetic confusion matrices (dataset), they limit it on the mentioned dialects. On the other hand,

the works of [34, 35] help the non-native learners to learn unfamiliar words and correct their mistakes, although these works are more educational programs and correct common errors made by non-native, they do not detect and correct the entire text (sentences). However, Arabic spell checking requires a lot of studies to handle the phonetic errors by applying the Soundex algorithm [42] which is designed specifically to deal with this type of error or apply other techniques.

**Hybrid Techniques**

Whenever the objectives of a spell checker increase, the used techniques to design this checker will increase to meet these objectives. It is known that each technique deals with certain errors and it has limitations with others. Thus, combining approaches are helpful to overcome the deficiencies of each one of them taken alone. A spell checker may combine two similarity distances to take out a new measurement to be more suitable in particular cases, such as [1, 3], or it may hybrid with the rule-based approaches, such as [35]. Furthermore, the author of [36] proposed a hybrid system based on the confusion matrix extracted from QALP corpus and the noisy channel spelling correction model.

It initially treats the missing space errors depending on a set of predefined common prefixes (rule-based), then the word with space is added to the suggestion's list. Otherwise, it applies character-based operations (with similarity techniques) to generate candidate words. Moreover, the work of [31] proposed a system based on a hybrid pipelines that combines rule-based linguistic techniques with statistical methods using language model and machine translation, in addition to an error-tolerant finite-state automata method. Generally, a hybrid approach is used to strengthen the outputs and achieve the goals more flexibly and fastly.

The summary in Table 1 and Table 2 present main studies of Arabic spell checking with their dataset and used techniques. According to the tables, most works focus on the isolated-word error more than context-sensitive error. Therefore, the last one needs more studies. On the other hand, most works combined techniques to overcome the limitations of the use of one method (algorithm) to provide better results.

## CONCLUSION

The paper surveys Arabic spell checking systems. We started explaining the specificities of the Arabic language. Then the paper presents the existing works according to the used approaches and techniques. The analysis of the existing systems showed that some of them use one particular technique, while others combine many of them. It is also noted that most spell-checking works mainly use the Levenshtein distance either by developing or integrating it with other distances.

On the other hand, our survey showed that even if every particular existing system has advantages and overcomes specific spelling problems to deal with certain types of errors, all systems still have shortcomings in other aspects. Therefore, there is still space to improve these systems and contribute in the development of enhanced Arabic spell checkers.

## REFERENCES

[1] G. Hicham, Y. Abdallah, and B. Mostapha, (2012), "Introduction of the weight edition errors in the Levenshtein distance," International Journal of Advanced Research in Artificial Intelligence, vol. 1, no. 5, pp. 30–32.

[2] B. Hamza, Y. Abdellah, G. Hicham, and B. Mostafa, (2014), "For an Independent Spell-Checking System from the Arabic Language Vocabulary," International Journal of Advanced Computer Science and Applications, vol. 5, no. 1, pp. 113–116.

[3] H. Gueddah, A. Yousfi, and M. Belkasmi, (2016), "The filtered combination of the weighted edit distance and the Jaro-Winkler distance to improve spell checking Arabic texts," IEEE/ACS International Conference on Computer Systems and Applications, AICCSA, vol. 2016-July, pp. 1–6.

[4] H. Muaidi and R. Al-Tarawneh, (2012), "Towards Arabic Spell-Checker Based on N-Grams Scores," International Journal of Computer Applications, vol. 53, no. 3, pp. 975–8887.

[5] M. M. Al-Jefri and S. A. Mahmoud, (2015), "Context-Sensitive Arabic Spell Checker Using Context Words and N-Gram Language Models," in Proceedings of Taibah University International Conference on Advances in Information Technology for the Holy Quran and Its Sciences, NOORIC, pp. 258–263.

[6] K. Bacha and M. Zrigui, (2016), "Contribution to the Achievement of a Spell checker for Arabic," Research in Computing Science, vol. 117, no. January, pp. 161–172.

[7] Y. Hassan, M. Aly, and A. Atiya, (2014), "Arabic Spelling Correction using Supervised Learning," the EMNLP 2014 Workshop on Arabic Natural Language Processing (ANLP), pp. 121–126.

[8] A. O. Al-Shbail and M. A. B. Diab, (2018), "Arabic Writing, Spelling Errors and Methods of Treatment," Journal of Language Teaching and

Research, vol. 9, no. 5, p. 1026.

[9] A. Martinench, (2014), "The Formation of Nominal Derivatives in the Arabic Language With a View to Computational Linguistics," University of Salford.

[10] T. Zerrouki and A. Balla, (2009), "Implementation of infixes and circumfixes in the spell checkers," in the Second International Conference on Arabic Language Resources and Tools, pp. 61–65.

[11] B. Haddad and M. Yaseen, (2007), "Detection and Correction of Non-Words in Arabic: A Hybrid Approach," International Journal of Computer Processing Of Languages, vol. 20, no. 04, p. 237.

[12] N. Gupta and P. Mathur, (2012), "Spell Checking Techniques in NLP: A Survey," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 12, pp. 2277–128.

[13] A. Protopapas, A. Fakou, S. Drakopoulou, C. Skaloumbakas, and A. Mouzaki, (2013), "What do spelling errors tell us? Classification and analysis of errors made by Greek schoolchildren with and without dyslexia," Reading and Writing, vol. 26, no. 5, pp. 615–646.

[14] V. V Bhaire, A. A. Jadhav, and P. A. Pashte, (2015), "Spell checker," International Journal of Scientific and Research Publications, vol. 5, no. 4, pp. 5–7.

[15] A. A. M. Mahdi, (2012), "Spell Checking and Correction for Arabic Text Recognition," King fahd university of petroleum & minerals.

[16] A. M. Azmi, M. N. Almutery, and H. A. Aboalsamh, (2019), "Real-Word Errors in Arabic Texts: A Better Algorithm for Detection and Correction," IEEE/ACM Transactions on Audio Speech and Language Processing, vol. 27, no. 8, pp. 1308–1320.

[17] A. S. Lhoussain, G. Hicham, and Y. Abdellah, (2015), "Adaptating the levenshtein distance to contextual spelling correction," International Journal of Computer Science and Applications, vol. 12, no. 1, pp. 127–133.

[18] B. Mohit, A. Rozovskaya, N. Habash, W. Zaghouani, and O. Obeid, (2015), "The First QALB Shared Task on Automatic Text Correction for Arabic," the EMNLP 2014 Workshop on Arabic Natural Langauge Processing (ANLP), pp. 39–47.

[19] T. Zerrouki, K. Alhawaity, and A. Balla, (2014), "Autocorrection Of Arabic Common Errors For Large Text Corpus QALB-2014 Shared Task," EMNLP 2014 Workshop on Arabic Natural Language Processing, no. 2005, pp. 127–131.

[20] N. AlShenaifi, R. AlNefie, M. Al-Yahya, and H. Al-Khalifa, (2015), "Arib @ QALB-2015 Shared Task: A Hybrid Cascade Model for Arabic Spelling Error Detection and Correction," Proceedings of the Second Workshop on Arabic Natural Language Processing, pp. 127–132.

[21] M. Attia, M. Al-badrashiny, and M. Diab, (2015), "Priming Spelling Candidates with Probability," in Proceedings of the Second Workshop on Arabic Natural Language Processing, vol. 10.18, no. January.

[22] M. I. Alkanhal, M. A. Al-Badrashiny, M. M. Alghamdi, and A. O. Al-Qabbany, (2012), "Automatic stochastic arabic spelling correction with emphasis on space insertions and deletions," IEEE Transactions on Audio, Speech and Language Processing, vol. 20, no. 7, pp. 2111–2122.

[23] K. Shaalan, M. Attia, P. Pecina, Y. Samih, and J. van Genabith, (2012), "Arabic Word Generation and Modelling for Spell Checking," the Eight International Conference on Language Resources and Evaluation, pp. 719–725.

[24] H. Gueddah and A. Yousfi, ( 2013), "The impact of arabic inter-character proximity and similarity on spell-checking," 8th International Conference on Intelligent Systems: Theories and Applications, Rabat, Morocco.

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

[26] R. Al-Tarawneh, H. S. A. Hamatta, H. Muiadi, P. Abdullah, and B. Ghazi, (2014), "Novel Approach for Arabic Spell-Checker: Based on Radix Search Tree," International Journal of Computer Applications, vol. 95, no. 7, pp. 975–8887.

[27] H. F. Alshahad, (2018), "Arabic Spelling Checker Algorithm for Speech Recognition," International Journal of Computer Science and Information Security (IJCSIS), vol. 15, no. 12, pp. 228–235.

[28] M. Attia, P. Pecina, Y. Samih, K. Shaalan, and J. Van Genabith, (2012), "Improved Spelling Error Detection and Correction for Arabic," Natural Language Engineering, vol. 22, no. 5, pp. 103–112.

[29] N. Mohammed and Y. Abdellah, (2018), "The vocabulary and the morphology in spell checker," in The First International Conference on Intelligent Computing in Data Sciences, vol. 127, pp. 76–81.

[30] A. El Oualkadi, F. Choubani, and A. El Moussati, (2016), "A lightweight system for correction of Arabic derived words," in Mediterranean Conference on Information & Communication Technologies, vol. 380, pp. 131–138.

[31] H. Bouamor, H. Sajjad, N. Durrani, and K. Oflazer, (2015), "Shared Task: Combining Character level MT and Error-tolerant Finite-State Recognition for Arabic Spelling Correction," the Second Workshop on Arabic Natural Language Processing, pp. 144–149.

[32] M. Attia, P. Pecina, and A. Toral, (2011), "An open-source finite state morphological

transducer for modern standard Arabic," in Proceedings of the 9th International Workshop on Finite State Methods and Natural Language Processing, pp. 125–133.

[33] K. Shaalan, A. Allam, and A. Gomah, (2003), "Towards automatic spell checking for Arabic," in Proceedings of the Fourth Conference on Language Engineering, Egyptian Society of Language Engineering (ELSE), Egypt, no. May, pp. 240–247.

[34] S. C. Wayland et al. , (2010), "Finding Entries in an On-line Arabic Dictionary," Human-Computer Interaction Lab 27th Annual Symposium, pp. 1–2.

[35] K. Shaalan, R. Aref, anda. Fahmy , (2010), "An approach for analyzing and correcting spelling errors for non-native Arabic learners," The 7th International Conference of Informatics and Systems.

[36] H. M. Noaman, S. S. Sarhan, and M. A. A. Rashwan, (2016), "Automatic Arabic Spelling Errors Detection and Correction Based on Confusion Matrix- Noisy Channel Hybrid System," Journal of Theoretical and Applied Information Technology, vol. 40, no. 2, pp. 54–64.

[37] Miniwatts Marketing Group, Arabic Speaking Internet Users Statistics, Internet World State usage Population Statistcs, June 30, (2017). Accessed on: Mar 3, 2019. [Online]. Available: https:/https://www.internetworldstats.com/stats19.htm

[38] Miniwatts Marketing Group, Arabic Speaking Internet Users Statistics, Internet World State usage Population Statistcs, June 30, (2017). Accessed on: Mar 3, 2019.

[39] Satu Limaye, ISLAM in ASIA, Asia-Pacific Center for Security Studies, April 16, (1999).

[40] "Levenshtein distance,", (2019). Accessed on: May 30, 2019.

[41] "Jaro–Winkler distance," May 30, (2019). Accessed on: May 30, 2019.

[42] "Soundex," June 14, (2019). Accessed on: June 14, 2019.

[43] Christopher D. Manning Prabhakar Raghavan Hinrich Schütze, (2009), "An introduction to Information retrieval", Cambridge University Press Cambridge, England.

**TABLE1: ISOLATED-WORD STUDIES OF ARABIC SPELL CHECKING**

| Work | Used dataset | Used techniques |
|---|---|---|
| Alshahad, 2018 [27] | | similarity techniques |
| Nejja and Yousfi , 2018 [29] | Sub-dictionaries | Morphology and similarity techniques |
| Hicham Gueddah et al., 2016 [3] | Learning corpus | Similarity techniques |
| Mohammed Attia et al., 2012 [28] | Arabic Gigaword Corpus, and news articles crawled from the Al-Jazeera website. | Hybrid techniques |
| Noaman et al., 2016 [36] | QALP corpus, and confusion matrix | Hybrid techniques |
| Nejja Mohammeda and Yousfi Abdellah, 2016 [30] | A corpus (containing 10000 word) constituted of surface patterns and roots characterized | Morphology and similarity techniques |
| Mohammed Attia et al., 2012 [28] | A dictionary of 9.3 million fully inflected Arabic words | Similarity, and Rule-based techniques |
| Bouamor et al., 2015 [31] | QALB corpus, AraComLex, and MADAMIRA | Hybrid techniques |
| Mohammed Attia et al., 2015 [21] | QALB corpus, Conditional Random Field (CRF), MADAMIRA morphological, and AraComLex Extended | Rule-based techniques |
| AlShenaifi et al., 2015 [20] | QALB corpus, KSU corpus, Arabic Corpora (OSAC),Al-Sulaiti Corpus, KACST Arabic Corpus, and MADAMIRA | Rule-based, and similarity techniques |
| Mohammed Attia et al., 2015 [21] | Arabic Gigaword Corpus, and a corpus crawled from Al-Jazeera | Rule-based techniques |
| Aouragh Si Lhoussain et al., 2015 [17] | | Similarity techniques |
| Youssef Hassan et al., 2014 [7] | QALB corpus,AraComLex2,MADAMIRA3, and Confusion matrix. | Rule-based, morphology, and similarity techniques |
| Al-Tarawneh et al., 2014 [26] | Muaidi Corpus | Similarity techniques |
| Zerrouki et al., 2014 [19] | QALB-2014 corpus and replacement list | Rule-based techniques |
| Gueddah Hicham et al., 2013 [1] | Set of Arabic documents typed by four expert users. | Similarity techniques |
| Hicham Gueddah and Abdallah Yousfi, 2013 [24] | Typing test of a training corpus | Similarity techniques |

| Work | Used dataset | Used techniques |
|---|---|---|
| Muaidi & Al-Tarawneh, 2012 [4] | Muaidi Corpus | Similarity techniques |
| Mohamed Alkanhalet al., 2012 [22] | A standard Arabic text corpus and test data (cover all types of spelling errors) | Rule-based techniques |
| Khaled Shaalan et al., 2012 [23] | | Hybrid techniques |
| Mohammed Attia et al., 2011 [32] | AraComLex, and a corpus of 1,089,111,204 words | Morphology techniques |
| Wayland et al., 2010 [34] | Arabic electronic dictionaries and confusion matrices | Similarity techniques |
| Khaled Shaalan et al., 2010 [35] | | Rule-based, and similarity technique |
| Khaled Shaalan et al., 2003 [33] | | Rule-based techniques |

**TABLE 2: CONTEXT-SENSITIVE STUDIES OF ARABIC SPELL CHECKING**

| Work | Used dataset | Used techniques |
|---|---|---|
| Azmi et al., 2019 [16] | KSU, ANC-KACST, and JM corpus. | Morphology and similarity techniques |
| Majed Al-Jefri and Sabri Mahmoud, 2015 [5] | Corpus from Al-Riyadh newspaper articles on three topics, in addition confusion sets (OCR) misrecognized words | Similarity, and relying on phonetics techniques |
| Mohammed Attia et al., 2015 [21] | Arabic Gigaword Corpus, and a corpus crawled from Al-Jazeera | Rule-based techniques |