Al-Qadisiyah Journal for Administrative and Economic Sciences ISSN$_{Online}$: 2312-9883
QJAE, Volume 28, Issue 2 (2026) ISSN$_{Print}$: 1816-9171
https:/ /doi.org/ 10.33916/qjae.2026.023647

A dynamic system for detecting and correcting spelling errors in Arabic texts using Levenshtein distance and dynamic programming

Fathala ali Chachan
Dept. of Financial &Banking, College of Administration & Economics, Mustansiriyah University,Baghdad,Iraq.
Fathala@uomustansiriyah.edu.iq

Alaa Shnaishel Cheetar
Dept. of Financial &Banking, College of Administration & Economics, Mustansiriyah University,Baghdad,Iraq.
alaash1973@uomustansiriyah.edu.iq

Fiaria ali Chachan
Dept. of mathematics, College of science, Mustansiriyah University,Baghdad,Iraq.
fiaria_altaqi@Uomustansiriyah.edu.iq.

Received: 05/04/2026
Accepted: 05/06/2026
Available online: 18/06/2026

Corresponding Author Alaa Shnaishel Cheetar

Abstract*:* Arabic is tougher to spell check on computers than other languages because the letters are shaped and placed in a way that is different from other languages. The spelling and accuracy of Arabic information have a big effect on how well it is in academic research, government documents, and teaching materials. This is because spelling mistakes can make a piece of writing hard to read and understand or distort its meaning. We need clever technology that can automatically discover and rectify spelling issues in texts from multiple areas, including Word and PDF files or digital photos. The purpose of this work is to make an Arabic text processing system that employs both automatic spelling mistake detection and repair and statistical analysis to measure the quality of the text in numbers, such as the number of correct and incorrect words and the error rate. The system utilizes a significant amount of new technology. The system initially utilizes optical character recognition (OCR) to extract text from pictures and data.After that, it fixes any mistakes in the text using the ar corrector library, which has an Arabic dictionary. The whole cost falls down, and the words that were repaired are more correct. We utilize the Levenshtein distance from the original word to figure out how much each suggestion costs. We use dynamic programming to find the best arrangement of words in the whole text.
Keywords: Checking spelling, fixing text automatically, and dynamic programming.

1. Introduction:

The Arabic language is one of the most complex languages in terms of morphology, spelling, and grammar, due to its diversity in letter forms and differences in word structure depending on context and meaning. This complexity has posed a major challenge for researchers in the field of natural language processing (NLP), especially with regard to the task of detecting and correcting spelling and linguistic errors in Arabic texts. [2] A new approach to designing a multilingual spell checker and corrector has been presented. Experiments have shown that the proposed algorithm guarantees finding the intended word with an accuracy of nearly 100%, if the word is present in the dictionary.]3[One of the first multi-system approaches to correcting grammatical errors in Arabic is the Enhanced Arabic Editing Selection (ArbESC+) system. The system uses several models to collect correction suggestions, which are represented by numerical features. A classifier determines and implements the appropriate corrections based on these features. To investigate the Arabic web spell checker Checker Spell Web, it can handle a large percentage of words covering classical Arabic in general and modern classical Arabic in particular using a computer dictionary. There are digital texts all over the place, and schools, newsrooms, and scientific research are all employing more and more automated tools. We need to quickly create systems that can read, check, and repair Arabic very well. This project was done to address a need by making a dynamic system that can discover and mend spelling issues in Arabic texts using Levenshtein distance and dynamic programming.

The method's purpose is to make spell checking better by using arithmetic and language correctness together. You can also use it with text from other locations, such Word and PDF files and photographs taken with a digital camera. The approach also does a full statistical study of the text's quality by counting the number of correct and incorrect words and working out the mistake rate and distribution. This is a significant step forward for learning Arabic. It combines artificial intelligence and linguistic analysis to create a useful application that helps people write better Arabic. This technology also sets the stage for future language apps that will be able to better understand what sentences mean and how they are put together.

2. Research objective:

The purpose of this project is to build a system that can read Arabic texts and discover and fix spelling issues in both documents and digital photos. The system uses the Levenshtein distance method to figure out how bad words are and dynamic programming to get the best correction sequence that costs the least and is the most correct linguistically. The technology also gives you analytical and statistical information on the quality of the writing, such as how many words are correct and how many are wrong. It also shows you mistakes in documents and pictures so that people may readily judge them.

3. Importance of the research:

This study is significant as it employs both linguistic and statistical approaches to rectify Arabic texts, hence enhancing the quality of academic content. You can also utilize the proposed method to assist you look at academic research and official papers. In the future, this will lead to the development of more advanced language systems that can understand context and correct grammar and meaning errors.

4. Research problem:

It's challenging for computers to read Arabic language since the letters seem different depending on where they are in a word. It's tougher to detect and rectify spelling issues in papers and digital photos that aren't all the same or come from different places. Also, dictionary-based solutions don't always provide you the best or most appropriate changes for the situation. This could lead to wrong results and make your Arabic writing worse.

5. Theoretical aspect:

5-1 Linguistic errors in digital and image documents[4, 8]

Recent years have seen a significant increase in the reliance of academic institutions on digital and image documents, which has increased the need for intelligent tools capable of detecting and correcting linguistic errors with high accuracy. Spelling errors are among the most common problems in documents resulting from manual entry or text extraction from images (OCR), as these errors negatively affect the quality and accuracy of scientific content. The most important of these errors are as follows:

5.2 Types of linguistic errors in digital and image documents [11]

5.2.1 Optical character recognition errors (OCR-specific errors)

These errors occur during the process of converting images into digital text, the most prominent of which are:

- Substitution errors: This is the process of replacing a character with another that is similar in appearance.
- Insertion errors: adding letters that are not present in the original word.
- Deletion errors: the loss of one of the letters of the word .
- Segmentation/spacing errors: the process of merging two words or separating one word into two words.
- Hyphenation errors: Errors resulting from the end of a line or the formatting of the text on the page.

5.2.2 Linguistic errors (language of the content itself)

Even after successfully converting text from an image to digital text, there may still be traditional linguistic errors such as:

- Spelling errors
- Grammatical errors
- Morphological errors
- Syntactic errors Sometimes.

contextual errors ("real-word" errors) also appear, where the word is spelled correctly but is not appropriate for the overall context of the sentence, which is difficult to detect using traditional methods.

5.2.3 Errors due to the quality of the original image

OCR accuracy is greatly affected by the quality of the original image, as noise and distortion of letters lead to an increase in errors in the resulting text.

5.2.4 Impact of these errors on academic documents .

1. Decreased research quality.
2. Distortion of OCR results, causing incorrect statistical analysis.
3. Misunderstanding of content by the reader.
4. Rejection of research by some journals due to numerous errors

5-3 Methods of detecting spelling errors [3]

5-3-1 Dictionary-Based Detection (4 and 2)

This method relies on comparing each word in the text with a list of correct words in the dictionary. If no entry in the dictionary matches, the word is considered incorrect. This method is the most commonly used in detection systems because it is straightforward and highly accurate.

$$\left.\begin{matrix} W \in Dictionary & 1 \\ otherwise & 0 \end{matrix}\right\} = Error(w) \dots\dots.. (1)$$

5-3-2 Detection Using Language Models (N-gram / Language Models).

This method relies on calculating the probability of a word appearing in a given context using statistical models such as N-gram. If the probability of the word is abnormally low, it is classified as a spelling error even if it resembles a correct word.

$$\frac{COunt\left(w_{i-n+1}^{i}\right)}{COunt\left(w_{i-n+1}^{i-1}\right)} = p\left(w_i \backslash w_{wi-n++1}^{i-1}\right) \dots\dots\dots\dots. (2)$$

5-3-3 Statistical Error Detection

This method is based on comparing the distribution of words in the text with the natural distributions of correct words in the language. Words that show a significant deviation from this distribution are considered incorrect.

5-3-4 Detection using OCR Confidence

When extracting text from images, the OCR system provides a "confidence" value for each word. Words with confidence below a certain threshold (e.g., 70%) are often the result of OCR errors and are treated as suspicious.

$$if\ w \Longrightarrow \Theta > Confidence(w) \text{ Suspicious}$$

5-3-5 Proposed hybrid detection for spelling errors.

This research proposes a hybrid detection mechanism for spelling errors in Arabic texts extracted from all types of digital and image documents, including Word files, PDFs, and digital images. This mechanism relies on a systematic

integration between detection based on the Arabic dictionary and the confidence level resulting from the optical character recognition system. The aim is to improve the accuracy of identifying incorrect words before moving on to the correction stage. This is the first stage in the proposed system for processing Arabic texts.

5-4-Detailed steps for the detection mechanism [6,7]

Step (1) Text input :

The detection process begins with entering text from various sources, including Digital text documents such as Word and PDF files that can be copied, where the text is extracted directly Image documents such as images and non-copyable PDF files, where they are converted to digital text using optical character recognition, The extracted text is represented as follows:

$$\{W_1, W_2, \dots\dots W_n\} = T \dots\dots (3)$$

with a confidence score for each word if it was extracted from OCR:

$$[1,0] \ni C(W_i) \dots\dots\dots (4)$$

Values close to 1 indicate high and accurate optical recognition, while low values indicate the possibility of errors resulting from image distortion, punctuation marks, or similar Arabic letters (ب, ت, ث, ي, ه, ة).

Step 2: Dictionary check of words

Each word in the text is checked against an Arabic dictionary to verify that it is linguistically correct. This is expressed mathematically by the function.

$$\left.\begin{matrix} w_i \in D & ,1 \\ w_i \notin D & ,\ 0 \end{matrix}\right\} = D(W_I) \dots\dots\dots (5)$$

where ($D(W_I) = 0$) , which indicates that the word is not found in the dictionary.

Step 3: Confidence rating (for texts extracted from OCR)

For texts generated by OCR, the confidence level associated with each word is analyzed using a specific threshold Ø to identify visual ambiguity as follows:

$$\left.\begin{matrix} C(w_i) \prec D & ,1 \\ C(w_i) \geq D & ,\ 0 \end{matrix}\right\} = O(W_i) \dots\dots\dots (6)$$

Words extracted from other documents are considered $O(W_I) = 0$

Step (4): Apply the hybrid decision function .

The results of the dictionary check are combined with the confidence level using a logical decision function that aims to identify words that are incorrect.

$$\left.\begin{matrix} D(w_i) = 0 \curlyvee O(w_i) & ,1 \\ otherwise\ , & 0 \end{matrix}\right\} = H(W_i) \dots\dots (7)$$

where a value of 1 indicates that the word is misspelled, while a value of 0 indicates that the word is spelled correctly.

Step (5): Extract the set of suspected words

The output of the hybrid detection stage is a set of words that have been classified as suspicious. This set is defined as follows.

$$\{1 = H(W_i)\backslash W\} = E \dots\dots (8)$$

5.5 Proposed spell checking stage

After completing the hybrid detection stage and extracting the set of words suspected of containing spelling errors, we move on to the spelling correction stage, which aims to suggest the correct alternative for each incorrect word and select the optimal correction based on mathematical criteria through the following steps.

1. Inputs for the correction stage Let the set of words detected in the detection stage be

$$\{1 = H(W_i)\backslash W\} = E$$

where W_i represents the word in question.

2. Candidate Generation

W_i a set of candidate words for correction is generated from the Arabic dictionary D.

$$D \supset \{c_1, c_2, \dots\dots, c_k\} = S(W_i) \dots\dots\dots (9)$$

so that each candidate word is within a specified editing distance

$$\epsilon > Lev(w_i, c_j) \dots\dots (10)$$

3. Calculating Levenshtein Distance

The correction process relies on the Levenshtein distance to measure the degree of difference between the incorrect word and the candidate word. The distance is defined as follows.

$$\left.\begin{array}{cc} delet & Lev(a_{i-1..1}, b_{j...1}) + 1 \\ add & Lev(a_{i..1}, b_{j-1...1}) + 1 \\ replace & Lev(a_{i-1..1}, b_{j...1}) + 1(a_i \neq b_j) \end{array}\right\} min = Lev(a, b) \dots\dots (11)$$

where 1(.) index function.

4. Correction cost function

For each candidate c_j a cost function is calculated based on the edit distance and the confidence level of the OCR.

$$\langle C(w_i - 1\rangle . B + Lev(w_i, c_j).\alpha = Cost(w_i, c_j) \dots\dots (12)$$

Where β, α Weight coefficients

$C(w_i)$ OCR confidence level for the original word.

Selecting the best candidate (correction)

In the case of single correction, the lowest cost candidate is selected:

$$\arg \min_{c_j \in S(w_i)} Cost(w_i, c_j) = c^* \dots\dots (13)$$

5.6 Correction using dynamic programming [10,12,13]

Dynamic programming is one of the effective algorithmic methods for solving sequential problems that depend on sequential decision-making. This approach is based on the principle of finding the optimal solution to the overall

problem based on optimal solutions to its sub problems. The importance of this method lies in its ability to reduce computational costs by storing and reusing partial results, making it suitable for processing texts that consist of a sequence of interrelated words. Since the process of correcting spelling errors in Arabic texts cannot be handled efficiently by correcting each word independently, due to the contextual errors or inconsistent linguistic choices that this may cause, dynamic programming has been adopted as a mathematical framework for selecting the best corrective sequence for the entire text. . In this way, the text can be represented as a sequence of words:

$$\{w_1, w_2, \dots, w_k\} = T$$

where representsw_i each word (a correct word or a suspected word resulting from the hybrid detection stage) so that this sequence reduces the total cost, which represents the objective function:

$$min \sum_{i=1}^{n} Cost(w_i, c_i) \dots\dots\dots (14)$$

with the context condition: $S(w_i) \ni c_i$

where at each step, the candidate that achieves the lowest cumulative cost is selected, taking into account the previous word, which ensures consistency of correction at the level of the text as a whole. In this way, correction is not viewed as a local process, but as a comprehensive contextual process aimed at improving the quality and spelling accuracy of the final text.

5.7.Outputting the corrected text

The final output is the corrected text.

$$c_1^*, c_2^*, \dots, c_n^* = T^- \dots\dots\dots (15)$$

while retaining the correct words unchanged.

5.8. Merging the correction results with the documents

After correction, the corrected words are highlighted in red within the Word documents, and then a statistical report is generated that includes.

$$\frac{\text{Number of incorrect words}}{\text{Number of incorrect words}} = Error\ Rate \dots\dots\dots (16)$$

6-Practical aspect:

This aspect of the research aims to apply the proposed theoretical framework in practice by designing and implementing an integrated system for detecting and correcting spelling errors in Arabic texts extracted from digital and image documents in a sequential manner, starting with several stages, namely

1. **Work environment:** The proposed system was implemented using the Python programming language because of its extensive support for natural language processing and Arabic texts, in addition to the ease of integrating statistical and mathematical algorithms. The implementation environment relied on Google Colab interactive development platforms because of the flexibility they offer in testing and re-implementation.

2. **The data used :**in this research was tested on a text stored in three formats (Word, PDF, scanned image). The text below will be tested in three formats.

First: Language and writing verification in Word format

The oldest known written language in Mesopotamia was Sumerian, an isolated agglutinative language. Semitic dialects were spoken in Mesopotamia early on alongside Sumerian. Later, the Semitic language, Akkadian , became the dominant language, although Sumerian was retained for administrative, religious, literary, and scientific purposes. Various varieties of Akkadian were used until the end of the Second Babylonian period. Then Aramaic, which had become common in Mesopotamia, became the official regional administrative language of the Persian Empire. The use of Akkadian ceased, but both Sumerian and Akkadian continued to be used in temples for several centuries.

Assuming that the text is entered from a Word file, the text will be extracted directly without going through the OCR stage and will be represented as a vector of words.

$$\boldsymbol{T} = \{\text{"اقدم "}, \text{"لغة" }, \text{"مكتوبة"}, \ldots\ldots, \text{"السومريه"}, \ldots, \text{"اللغه"}, \ldots, \text{"الاكديه"}, \ldots \text{"قرون"}\}$$

Since the source is digital, give a confidence score of 100% to each word.

$$C(W_i) = 1$$

1. Dictionary check stage

Each word is checked using the Arabic dictionary D

$$\left.\begin{matrix} If\ the\ words\ are\ in\ the\ dictionary & ,1 \\ If not\ the\ words\ are\ in\ the\ dictionary & ,0 \end{matrix}\right\} = D(W_i)$$

$$D(T) = \{\text{"1 "}, 1\ ,1, \ldots\ldots, 0, \ldots, 1, \ldots, 0, .1\}$$

Where:

Sumerian D ∌

Al-Akdiya D ∌

A value of (0) in the dictionary check function indicates that the word does not match any entry in the Arabic dictionary in its current form. This does not necessarily mean that it is linguistically incorrect, but rather that it is a candidate for checking and correction in a later stage.

2. . Syntactic verification stage

At this stage, the linguistic structure of adjacent words is analyzed and morphological and syntactic errors in the text above are detected. It appears that all grammatical structures in the text are correct, so this stage can be skipped without affecting the rest of the system.

That is $s(w_i, w_{i+1}) = 0$

3. .Hybrid decision function

At this stage, the results of the lexical check and syntactic analysis will be combined using equation 7. The results were $H(w_I) = 0.$

4. .Extracting suspicious words.

$$E = \{"الاكديه", "السومريه"\}$$

The above group contains only words that will pass to the correction stage.

5. Correction stage using dynamic programming

At this stage, corrective alternatives for each word are generated based on the dictionary, and then the Levenshtein distance is calculated using dynamic programming. The results were as follows.

Distance	Alternative	Incorrect words
1	السومرية	السومريه
1	الأكدية	الاكديه

Where the least costly alternative is chosen.

5. Statistical analysis

The text contained 128 words, with 2 errors, while the error rate before correction was 1.56%. After correction, the error rate became 0%.

text contained	errors	before correction	After correction
128 words	2	%1.56	%0

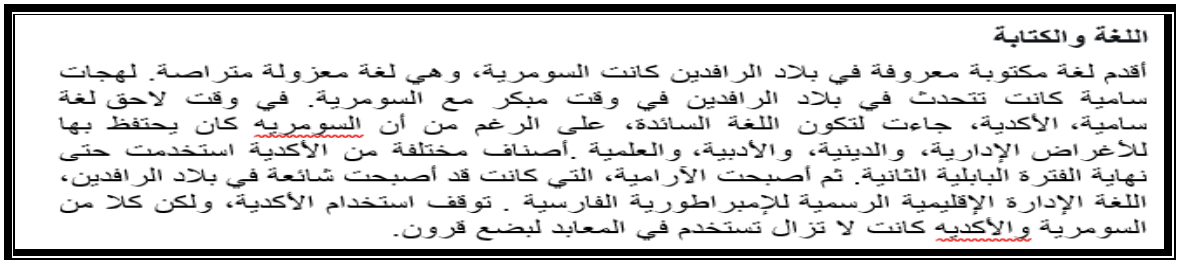
Part of a program for correcting text in Word format.

```
# ------------------------------
# دالة معالجة ملفات Word
# ------------------------------
def process_docx(filename, word_probs=None):
    doc = docx.Document(filename)
    total_words, correct_count, incorrect_count = 0,0,0
    for para in doc.paragraphs:
        words = para.text.split()
        if not words:
            continue
        corrected_words = optimal_correction(words, corrector, word_probs)
        para.clear()
        for orig, corr in zip(words, corrected_words):
            total_words += 1
            run = para.add_run(corr + " ")
            if orig == corr:
                correct_count +=1
                print(f"☑ {orig} صحيحة")
            else:
                incorrect_count +=1
                run.font.color.rgb = RGBColor(255,0,0)
```

Second: Checking in image or PDF format

When entering the text below in the form of a scanned image or PDF into the system, we notice that the words are not visually confirmed, but first pass through an optical character

recognition (OCR) system, which assigns a confidence score (a numerical value between 0 and 1) to each word, indicating the degree of certainty with which it has been recognized.



اللغة والكتابة

أقدم لغة مكتوبة معروفة في بلاد الرافدين كانت السومرية، وهي لغة معزولة متراصة. لهجات سامية كانت تتحدث في بلاد الرافدين في وقت مبكر مع السومرية. في وقت لاحق لغة سامية، الأكدية، جاءت لتكون اللغة السائدة، على الرغم من أن السومريه كان يحتفظ بها للأغراض الإدارية، والدينية، والأدبية، والعلمية .أصناف مختلفة من الأكدية استخدمت حتى نهاية الفترة البابلية الثانية. تم أصبحت الآرامية، التي كانت قد أصبحت شائعة في بلاد الرافدين، اللغة الإدارة الإقليمية الرسمية للإمبراطورية الفارسية . توقف استخدام الأكدية، ولكن كلا من السومرية والأكديه كانت لا تزال تستخدم في المعابد لبضع قرون.

1. After processing the image, the text is represented as a vector of words with a confidence score for each word.

Word	اللغة	الكتابة	اقدم	لغة	مكتوبة	معروفة	في	بلاد
Confidence level	0.98	0.97	0.96	0.99	0.97	0.98	0.99	0.98
Word	الرافدين	كانت	وهي	معزولة	متراصة	سامية	الاكديه	السومريه
Confidence level	0.95	0.97	0.99	0.96	0.70	0.74	0.68	0.69

We note that the Sumerian word obtained a score of$(\text{السومريه}) = 0.69$ due to the visual similarity between (ة,ه) and the (الاكديـه) word obtained a score of 0.68 due to the visual overlap between the letters ي and ه. When compared with the confidence threshold, , the two words are visually questionable.

2.Dictionary verification stage: In parallel with the visual evaluation, the existence of the word in the Arabic dictionary in the system is verified according to equation (5), where the two words (Sumerian and Akkadian) were found to be questionable. To achieve higher accuracy in detection, the results of the visual confidence and dictionary check are combined using a hybrid decision function, and it was found that $H(W_i) = 0$ correction. for the words nominated for.

3. Correction stage using dynamic programming: The correction stage relies on generating linguistic alternatives from the Arabic dictionary, then testing the optimal alternative using the Levenshtein distance calculated through dynamic programming. The results were as follows.

Incorrect words	Alternative	Distance
السومريه	السومرية	1
الاكديه	الاكدية	1

The least costly alternative is selected, after which the corrected words are replaced and the final text, free of spelling errors, is returned ready for statistical analysis.

4. Statistical analysis:

The number of words in the text was 128, and the number of incorrect words was 2. The error rate before correction was 1.56%, while the error rate after correction was 0%. A copy of the text processing program in PDF format is attached as Appendix 2.

text contained	errors	before correction	After correction
128 words	2	%1.56	%0

7-Conclusions:

1. The results show that using only dictionary-based detection is not adequate to find all the mistakes, especially those that happen when Arabic letters look similar or the image quality is poor.
2. The hybrid detection method, which combines dictionary checking and OCR confidence, has been shown to make it easier for humans to find misspelled words.
3. It's clear that dynamic programming works at the spell-checking stage to find the best answer for the complete text, taking into consideration how words fit together and what they mean in context.
4. The research found that spelling mistakes in Arabic digital and picture documents, especially those induced by optical character recognition (OCR) technology, are a serious problem that makes academic papers less accurate and of inferior quality.

8-Recommendations:

1. The study recommends adopting the proposed hybrid model, which combines dictionary based checking with the confidence level of the optical character recognition (OCR) system, as an effective approach to detecting spelling errors in Arabic texts extracted from digital and image documents.
2. It is necessary to integrate dynamic programming into the spell-checking stage, as it provides greater linguistic consistency when correcting entire academic texts compared to correction based on individual words.
3. The results recommend adjusting the confidence level criteria in the OCR system flexibly to match the quality of the original document, with the aim of reducing errors resulting from visual interference or the similarity of Arabic letters.

8.Recommendations:

1. Al-Najim &Dr. Saleh Rashid (2022). A spell-checking system for Arabic on the World Wide Web using a computer dictionary and spelling and phonetic rules. Journal of the Faculty of Arts, Alexandria University, Volume 72, Issue 110, October 2022.

2. Abozinadah, Ehab A.(2016). Improved micro-blog classification for detecting abusive Arabic Twitter accounts. International Journal of Data Mining & Knowledge Management Process.
3. Alkhafaji, Hussein and Abdullah, Suhail and Merza, Hanaa.(2013). A new algorithm to design and implementation of multilingual spellchecker and corrector. Journal of Al-Rafidain University College For Sciences. https://doi.org/10.55562/jrucs.v32i2.317.
4. Al-Qaraghuli, Mohammed and Jaafar, Ola Arif.(2024). Arabic Spelling Correction with T5. Jordanian Journal of Computers and Information Technology.
5. Al-Jefri, Majed Mohammed Abdulqader.(2016).Real-word error detection and correction in Arabic text. King Fahd University of Petroleum and Minerals.
6. A Alrehili, A Alhothali (2025). ARBESC+: ARABIC ENHANCED EDIT SELECTION SYSTEM COMBINATION FOR GRAMMATICAL ERROR CORRECTION RESOLVING CONFLICT AND MPROVING SYSTEM COMBINATION IN ARABIC GEC . https://doi.org/10.48550/arXiv.2511.14230.
7. Attia, Mohammed and Pecina, Pavel and Samih, Younes and Shaalan, Khaled and Van Genabith, Josef(2016).Arabic spelling error detection and correction. Natural Language Engineering, Cambridge University Press. https://doi.org/10.1017/S1351324915000030.
8. Essa, Nada and Elgayar, Mostafa and El-Daydamony, Eman.(2025). Arabic Grammar Correction for Arabic Text Summaries. Mansoura Journal for Computer and Information Sciences. http://doi:10.21608/mjcis.2025.353920.1009.
9. Mahdi, Adnan Abdo Mohammed.(2016). Spell checking and correction for Arabic text recognition. King Fahd University of Petroleum and Minerals ePrints.
10. Muaidi, Hasan and Al-Tarawneh, Rasha.(2012). Towards arabic spell-checker based on N-grams scores. International Journal of Computer Applications.
11. Musyafa, Ahmad and Gao, Ying and Solyman, Aiman and Khan, Siraj and Cai, Wentian and Khan, Muhammad Faizan.(2024). Dynamic decoding and dual synthetic data for automatic correction of grammar in low-resource scenario. PeerJ Computer Science. http://dx.doi.org/10.7717/peerj-cs.2122.
12. M Salhab, F Abu-Khzam (2023). AraSpell: A deep learning approach for Arabic https://doi.org/10.21203/rs.3.rs-2974359/v1.
13. Saty, Ahmed Abdalrhman and Bouzoubaa, Karim Bouzoubaa and Lhoussain, Aouragh Si.(2020). Survey of Arabic checker techniques. Journal of Engineering and Computer Science
14. Youness Chaabi, Fadoua Ataa Allah.(2022).Amazigh spell checker using Damerau-Levenshtein algorithm and N-gram. Journal of King Saud University – Computer and Information Sciences. https://doi.org/10.1016/j.jksuci.2021.07.015.

Appendix (1) Copy of program results