InfoTech Spectrum: Iraqi Journal of Data Science (IJDS)
ISSN: 3007-5467, Vol 3 (No. 2), Online First (March)- (2026)
DOI: https://doi.org/10.51173/ijds.v3i2.61
https://ijds.mtu.edu.iq

DESIGN OF AN INTELLIGENT AGENT FOR EMAIL AUTOMATION THROUGH NLP-BASED INFORMATION EXTRACTION FROM ARABIC TEXT

Faiez Musa Lahmood Alrufaye[1*], Mohammed Ibraheem Hussein[2], Amaal Ghazi Hamad Rafash[3]

[1] *Technical Instructors Training Institute, Middle Technical University, Baghdad, Iraq*

[2] *Ministry of Higher Education and Scientific Research, Baghdad, Iraq*

[3] *Technical Engineering College of Artificial Intelligence, Middle Technical University, Baghdad, Iraq*

** Corresponding author E-mail: Faiez.Alrufaye@mtu.edu.iq*

RESEARCH ARTICLE

ARTICLE INFORMATION

SUBMISSION HISTORY:
Received: 10 September 2025
Revised: 2 December 2025
Accepted: 12 December 2025
Online First (March 2026)

KEYWORDS:

Natural Language Processing;
Information extraction;
Intelligent Agent;
Email Automation;

ABSTRACT

Through the tremendous results achieved by its numerous technologies across a variety of sectors, including scientific, technical, medical, and other fields, artificial intelligence has exceeded expectations and overcome constraints. One area of artificial intelligence called "natural language processing" has demonstrated success in a number of natural language applications, including English, Arabic, German, and other languages. Automatic Natural Language Processing is a technique that uses algorithms to mimic human labor in natural language processing, reducing the time and effort required for an individual to undertake tasks necessary for that processing. The goal of information extraction is to automatically extract a tailored set of data from a massive volume of input text. Many online applications depend substantially on data extraction to function. In this research, we will use Natural Language Processing (NLP) to extract relevant information from an Arabic text and transmit it to the recipient via email. Every system or algorithm for natural language processing is assessed not only for performance, efficacy, and efficiency, but also for the discovery of novel processing techniques applicable to the NLP domain. Three techniques are available for assessing the results: F-measure, precision, and recall. When lexical phrases are used, the information extraction methodology produces very good results; its effectiveness ranged from 86% to 100%, with an average precision of 91.4%, average recall of 90.2%, and average F-measure of 90.7%.

1. INTRODUCTION

Artificial intelligence (AI) has exceeded expectations and set new limits through the great successes of its technologies across multiple fields, including science, engineering, medicine, and others [1]. Natural Language Processing is one of the fields of artificial intelligence, which has shown several successes in various applications in the field of natural languages, including English, Arabic, German, and other languages [2]. Automatic Natural Language Processing is used to design programs that simulate human work in natural language processing, while saving time to complete that processing and reducing the effort a person must put into accomplishing what is required by that processing [3]. Among the most important applications of natural language processors:

a. **Information Extraction:** Its purpose is to extract automatically customized data collection from a large amount of text. In order to function, a vast number of web apps rely heavily on data extraction. Gaizauskas, a researcher, created the M-Lasie system in 1997, an essential technique for Multilingual Information Extraction [3][4].

b. **Automated Summarization:** Is to transform a source document into a concise, user-friendly format for the reader. This involves removing unnecessary details from the original text while preserving its core ideas. Summaries can be created from a single document (single-document summarization) or from multiple documents combined into a single summary (multi-document summarization) [5][6].

DOI: https://doi.org/10.51173/ijds.v3i2.61

c. **Machine Translation:** Machine Translation converts text or speech from one language into multiple others. In this process, word clustering is often a practical and efficient technique for building automated translation systems used in language transformations that involve morphological contradictions, such as the transition from French to English [7]. The processing of unstructured data and the extraction of vital information into well-structured, organized formats that are easier to edit and interpret is known as information extraction.

Consider the following scenario: we're reviewing the financial details of a corporation based on a few papers. When dealing with digital data, we usually seek specific information or manually examine it. Add to that the difficulties inherited in Arabic due to its rich vocabulary that exceeded 12 million words with complex structure and root-and-pattern morphology, which yield challenges for tokenization, text processing, and generation. The main issue in this study is the absence of an integrated system that can comprehend natural Arabic text, extract important information (such as email addresses) and user intent, and automatically transition from comprehension to a useful application, such as creating and sending an email to the extracted address. With millions of words and an enormous variety of grammatical structures and derivations, Arabic is characterized by unparalleled morphological and linguistic richness, which makes this issue even worse. A serious lack of trustworthy semantic resources, like the English WordNet database, which is essential for comprehending the context and meaning of words in other languages, is another issue facing the subject.

However, using NLP information extraction algorithms, we can automate the extraction of all essential information from various document types, including tables, corporate growth metrics, and other financial facts (PDFs, Docs, Images, etc.). In this paper, information extracted from an Arabic text will be sent via email by extracting important information about the email address using Natural Language Processing (NLP).

2. LITERATURE REVIEW

NLP techniques must first be used to understand the text before the e-mail can be directed to the intended recipient. This task of extracting e-mail information or its evidence was represented in identifying and classifying the text, which is very new to the Arabic language. Most existing work focuses on extraction (tokenization, summarization, etc.), while our study bridges the gap between extraction and automated email composition. The most significant contributions to the field of information extraction will be discussed in this section of the study, along with the key discoveries made by scholars in this area.

Hawraa Fadhil Khelil focuses on classifying Arabic customer reviews using four classifiers: K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), and Naïve Bayes (NB). Three stemming methods are used in the categorization process: Tashaphyne, Khoja, and Snowball. The HARD dataset is used in experiments, and the results show that stemming techniques enhance classification performance despite the difficulties caused by Arabic characters and dialect variances [8]. Amjad A. Alsuwaylimi suggested a hybrid strategy that combines deep learning and machine learning methods to improve the identification of phishing emails in Arabic. By finding the most relevant characteristics, a genetic algorithm is used to improve feature selection and enhance the model's performance. The 1,173 items in the dataset are separated into two groups: phishing and authentic emails. To assess and contrast the performance of the suggested model, extensive tests were carried out [9].

A research by Masri A. and Al-Jabi M. employed natural language processing (NLP) methods to categorize Arabic business emails. The study used a dataset of 63,257 emails and divided them into three categories: subject, urgency, and sentiment. To detect email properties, their method used a lexicon-based system in conjunction with machine learning approaches. For every categorization situation, a different model was constructed, trained, and evaluated using different configurations of convolutional neural networks (CNNs). Their findings showed a loss of less than 8% and an accuracy of almost 92% [10].

The development of Arabic Noun Entity Recognition (NER) systems, a basic method for information extraction that supports a variety of applications, including knowledge mapping and question-and-answer systems, is thoroughly reviewed by Qu et al. The first section of the paper gives background information on the traits and difficulties of the Arabic language, as well as the databases and linguistic resources that are accessible for training and assessing Arabic NER models. The progression of methods from rule-based models and manual feature engineering to models based on sequential vector representations and deep learning techniques like recurrent neural

networks, transformer models, and pre-trained transformers particularly for Arabic is then reviewed by the researchers. In terms of resource accessibility, dataset quality, and the use of pre-trained models, the study also addresses the methodological differences between Arabic NER and its equivalents in other languages. Finally, it proposes several future research directions to enhance the accuracy of Arabic systems and their generalizability to different contexts and dialects [11].

A method for translating Arabic text into a Resource Description Framework (RDF) semantic representation is presented in the paper by Gehad Zakria et al. The proposed system incorporated a grammatical parser to extract triads (subject-subject-object) from previously analyzed Arabic text. In addition, entities mapped to DBpedia were extracted to derive URIs using named entity recognition. The Arabic script's semantics are then captured in a comparable RDF representation [12]. Emna Hkiri and colleagues described a project still in progress to develop a model for extracting events from Arabic texts using the portal platform and other technologies. Event extraction involved finding and classifying events in the open field text. While it has matured for certain languages, including English and French, it is still extremely new to the Arabic language. It has been demonstrated that event extraction can improve the performance of natural language processing tasks, including information retrieval and question answering, text mining, and machine translation etc. [13].

The goal of the study provided by Sally Mohamed Ali El-Morsy et al. was to demonstrate an OIE system that carefully analyzed all potential textual linkages and retrieved the set of relations from Arabic online text using Arabic dependency analysis. The identities of the associated clause types were constructed based on the recommendations of clause types as extractable relations and the syntactic functions of the components. The suggested Arabic Open Information Extraction (AOIE) technology is domain-neutral and can extract highly scalable Arabic text relations. While the system depends on unsupervised extraction procedures, the suggested system's implementation used supervised strategies to overcome the issue. In order to prevent information from being extracted in one specific region, the system has also been established in other locations. The outcomes demonstrate that the algorithm was highly effective at separating out sentences from enormous volumes of material [14]. Our system is designed to extract email information or any information that indicates it by applying NLP techniques to analyze and understand the text, and in light of the results, an email will be sent to the person concerned.

3. NLP AND INFORMATION EXTRACTING

Daily collection, analysis, and organization of vast amounts of data are necessary. Technology is employed to retrieve the information since doing it manually would be both time-consuming and chaotic. The machine learning and natural language processing methods that assist in achieving this are discussed in this section of the article [15]. Information extraction is the process of analyzing unstructured data to retrieve essential information and convert it into structured, easily editable formats [16]. Usually, dealing with a lot of text material is time-consuming and frustrating. Therefore, a lot of companies and organizations depend heavily on Information Extraction methods to apply innovative NLP algorithms to automate their operations. By minimizing human labor and improving the process's accuracy and efficiency, information extraction may help businesses save time and money [17]. Information may be gleaned from text input using deep learning and NLP methods like named entity recognition. The data type to be working with, such as bills or records, should be considered if we're beginning from scratch [18].

Many NLP-based applications employ the information extraction mechanism. Examples include the extraction of abstracts from big text collections like Wikipedia, conversational AI tools like chatbots, etc. You must first comprehend the sort of data being handled to comprehend how natural language processing algorithms for information extraction work. This will provide vast help in separating the data we acquired from the unstructured data [19]. Despite the large amount of textual data, it is highly challenging to extract well-structured, editable information from it due to the natural language complexity. Despite the challenges of the information extraction process, nearly all information extraction systems follow a pipeline of recurring steps [20].

Due to the many applications catering to user needs in the fields of natural languages, NLP has emerged as one of the most important processors in the software industry. The most important applications include text summarization, information extraction, machine translation, and others [18]. When processing data, all these applications go through two stages: the first is the natural language understanding stage, which involves a number of stages such as lexical analysis, syntactic analysis, semantic analysis, and so on. This step, known as natural language generation, will be taken into account when an option is reached. Another level in natural language processing [19]. The main difficulty with natural language processing's decision-making process is when the

conclusion is made based on an inaccurate comprehension of the text during the first step, known as "natural language understanding" [20]. The stages of natural language processing in most text-based applications are described in Fig.1.

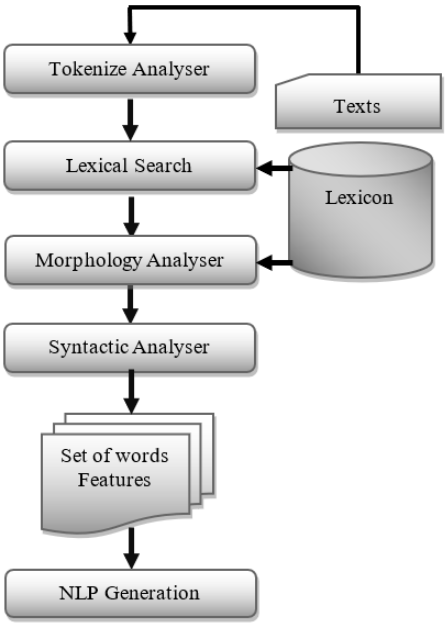

Figure 1. The stages of natural language processing [20]

4. THE PROPOSED INTELLIGENT AGENT

The proposed system goes through five stages, four of which fall under the concept of (stages of language understanding), through which the language is understood, and words are classified into verbs, nouns, and letters. This stage produces word features that are later used in the language generation stage. The first four stages are the same as those of natural language processing. The fifth stage is the debriefing stage, during which a decision is made based on the data of the previous stages. The result of the information extraction stage is the email information extracted from the text (Fig. 2).

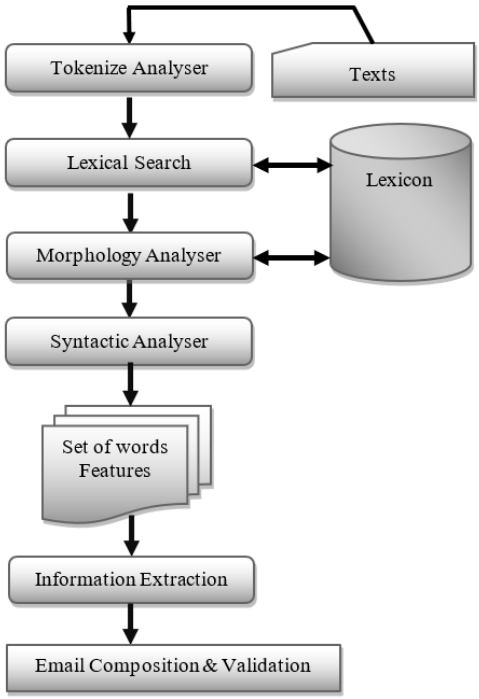

Figure 2. The stages of the proposed intelligent agent

4.1 Dataset Characteristics

The proposed intelligent agent was evaluated using a specially designed Arabic text set. To our knowledge, there is no publicly available Arabic reference standard that provides standard texts for assessing natural language understanding and information extraction in this field. One hundred Arabic texts from various real-world sources, such as official administrative letters, instructional messaging, and casual social conversation, make up the dataset. One or more linguistic clues that obliquely explain the target email address to which the message should be sent are included in each text, which is composed as a brief, cohesive paragraph about a particular subject (e.g., identifying the organization's domain and username separately in the text). Instead of depending just on pattern matching, this approach requires the system to conduct true language interpretation and information extraction.

Ninety-three of the 100 messages have at least one working email address; the other texts are used to evaluate the system's capacity to prevent false positives. A more thorough examination of performance under various linguistic conditions (standard versus informal language, the presence of dialectal forms, and varying grammatical complexities) was made possible by the grouping of the texts by source and style into 50 formal correspondences, 30 educational texts, and 20 social messages. A realistic benchmark for evaluating the robustness and generalizability of the suggested natural language processing pipeline is provided by this stratified dataset structure, which guarantees that the evaluation covers a wide range of Arabic writing styles the system is expected to encounter in practical applications, from well-structured corporate emails to less controlled social messages (Fig. 3).

مرحبا احمد
انا فائز موسى مدرس في الجامعة التقنية الوسطى، تم اخذ بياناتك من وزارة التربية، انت طالب في الصف السادس الاعدادي.
اود ان اعطيك معلومات عن الجامعة التقنية الوسطى في حال إذا رغبت بالتقديم للدراسة فيها، تتكون الجامعة التقنية من مجموعة من الكليات والمعاهد التقنية والتي يكون فيها الجانب العملي والتدريب أكثر من الجانب الأكاديمي مما يؤهلك لسوق العمل.
يمكنك مراسلتي في اي وقت البريد الالكتروني هو نفسه نطاق موقع الجامعة التقنية الوسطى mtu.edu.iq والمعرف الخاص ببريدي الالكتروني هو اسمي الاول باللغة الانكليزية Faiez واسمي الثاني Musa بينهما نقطة.

Figure 3. An example of blog texts

4.2 Stages of Language Understanding

4.2.1 Tokenize Analyzer

At this stage, the text will be divided into words to make it easier to analyze the words and extract their features.

4.2.2 Lexical Search

At this stage, a search will be conducted to determine the type of word (verb, noun, or letter). Some words will not be found by the system in the lexicon, because they contain extra letters at their beginning, which are called prefixes, and additional letters at the end, which are called suffixes. In this case, the system will move to the morphological analysis stage. A lexicon was built consisting of three tables: a table of verbs, which contains the roots of verbs, a table of nouns, which contains proper nouns and adjectives, and a table of letters, which includes letters, pronouns, and other words that are not verbs or nouns.

4.2.3 Morphology Analyzer

At this stage, the word will be segmented to return it to its origin. Prefixes and suffixes are cut out and then searched in the lexicon. The morphological analysis stage is considered one of the most difficult stages of the system because it analyzes the root of a word and segments it by deleting prefixes and suffixes to distinguish it, as Arabic morphology is one of the most distinctive characteristics of the Arabic language.

4.2.4 Syntactic Analyzer

The morphological analyzer may not be able to recognize some words, due to the large number of prefixes and suffixes a word may have, which creates difficulty in distinguishing it. Hence, the system moves to the syntactic analyzer, where the proposed system can determine the word type through its position in the sentence, whether it is a verb or a noun.

4.3 Language generation stage

4.3.1 Information Extraction

At this stage, information will be extracted based on the characteristics identified in the previous stages. The system performs the process of extracting information based on the entity recognition algorithm, by searching for a word that indicates the email or message, and by understanding the correct text in the previous stages, the email will be extracted. According to the previous example, the system will search for words such as (email, message, messaging, domain, and email ID), and then search the text for words related to those words (Entities). It links those words to get the email based on the following equation 1:

$$dist(E_i.E_j) = \sum_{f \in F} w_f \text{incompatibility}_f \left(E_i.E_j\right) \quad \ldots (1)$$

The incompatibility function compares two items based on attributes such as name, location, number, gender, and semantic class. Two entities, Ei and Ej, are said to be part of the same cluster if their distance from one another is smaller than the pre-established cluster radius.

4.3.2 Email collection

The terms that denote email will be gathered and put in a certain location to infer the email, in accordance with the earlier phases of the proposed system. The above example suggests that the following terms denote email:

(مراسلتي، نطاق الجامعة التقنية الوسطى، المعرف الخاص بي)

The email address is clearly indicated by the phrases above, and based on the preprocessing to interpret the language, an email address will be created and supplied with the data.

4.3.3 Email Validation

The extracted information is formatted into a predefined email template and validated for accuracy before sending. Manual validations are used to ensure the robust automation quality performance and to integrate feedback mechanisms to improve future extractions and compositions. As shown in Algorithm 1

Algorithm 1. Intelligent Agent for Email Automation from Arabic Text

Input: Raw Arabic text (e.g., a paragraph or message).
Output: Extracted and validated the email address, then composed and sent an email to it.
Basic Structures:

- Lexicon: A database containing:
- Verb table (including verb roots).
- Noun table (including proper nouns and adjectives).
- Character table (including prepositions, pronouns, interrogative words, etc.).
- Processed_Text: A list of words (tokens) with the linguistic attributes associated with each (e.g., {word: "winner", type: "ism", root: "فاز"}).

Stage 1: Text Understanding
Goal: Analyze the text to understand its linguistic structure.
Tokenization
Words = Arabic_text_segmentation(raw_Arabic_text)
Segment the input text into separate words.
Morphological and Syntactic Analysis Loop
For each word in the word list:
Word_attributes = {} (Creates an empty object to store the attributes of the current word)
Step 2.1: Lexical Search
Search for a word in the dictionary.
If found:
Set word_attributes['type'] (verb, noun, particle).
Set word_attributes['root'] = word (assuming it is in its root).
If not, go to Step 2.2.
Step 2.2: Morphological Analysis
Root = morphological_analyzer(word)
Remove possible prefixes and suffixes to find the root of the word.
Search for the root in the dictionary.
If found:
Set word_attributes['type'] and word_attributes['root'] = root.
If the root is not found yet, go to step 2.3.
Step 2.3: Syntactic Analysis
word_attributes['type'] = syntactic_analyzer(word, sentence_context)

Determine the word type (verb, noun, etc.) based on its position and role in the sentence.
Add the word_attributes of the current word to the processed_text list.
Stage 2: NLP Generation
Goal: Use the understood text to generate the desired output (email).
Information Extraction
Define a set of email_semantic_words: ["mail", "email", "correspondence", "domain", "id", "message", "email"].
Scan the processed text to find any email semantic words.
For each semantic word found, search for associated entities (e.g., proper nouns, domain strings) within a specified range of surrounding words.
Apply the entity clustering equation to join related entities:
Distance(k_i, k_j) = Sum of (attribute_weight * attribute_mismatch(k_i, k_j)) for all attributes
Where k_i and k_j are entities (e.g., "Faiz" and "mtu.edu.iq").
K_i and k_j are clustered if the distance(k_i, k_j) is less than a predefined threshold value.
Email Address Construction
From the set of clustered entities, extract:
Local_Part: This is often a name (e.g., "Faiez.Musa" from the names "Faiez" and "Musa").
Domain: A recognized domain (e.g., "mtu.edu.iq").
Construct the candidate email address: candidate_mail = local_part + "@" + domain.
Step 3: Email Automation
Email Validation
If the candidate_mail matches a standard email pattern (using a Regex):
validated_mail = candidate_mail
Otherwise:
Log the error for manual review.
Stop the algorithm.
Email Composer and Send
Get a pre-prepared email template.
Populate the template with validated_mail and context extracted from the processed text.
Send_Email(recipient_address=verified_email, message=composed_message)
Stage 4: Offline Evaluation
Note: This step evaluates the system as a whole and is not part of the real-time operation.
Calculating Performance Metrics
After processing a dataset of n texts, calculate for each text n:
Precision (P) = (Number of email addresses correctly extracted) / (Total email addresses extracted by the system)
Recall (R) = (Number of email addresses correctly extracted) / (Total real email addresses in the text)
F-Measure (F) = (2 * P * R) / (P + R)
Report the average precision, recall, and F-Measure across the entire dataset.
End

5. RESULTS AND DISCUSSION

5.1 Evaluation Tools

Following the completion of each operation, the system's outputs need to be examined. Every algorithm or system for natural language processing is assessed not only for performance, efficiency, and effectiveness, but also for finding new processing methods that may be used in the NLP field. Three methods were used for evaluating the results [21]:

5.1.1 Precision Method

The ratio of accurate referrals the system found to the total number of accurate, incorrect referrals discovered using our approach is calculated using the following equation:

$$P = \frac{Tc}{Ta} \qquad \ldots (2)$$

Where Ta represents all true and erroneous classifications that are detected, and Tc is the number of accurate classifications made by our approach.

5.1.2 Recall Method

Using this measure, the ratio of valid categories found to the total number of actual valid categories in the text is obtained with the following equation:

$$R = \frac{Tc}{Tt} \qquad \ldots (3)$$

Where Tt is the total number of accurate textual classifications, and Tc is the number of correct classifications made by the system.

5.1.3 F-Measure

This measure is a derivative of the two previous measures, in which the following equation is the computation of the ratio of twice the factorial of the first method's results divided by the second method's results, which is then added to the total of the products of the two methods:

$$R = \frac{2PR}{P+R} \qquad ...(4)$$

The influence of outliers in individual texts is lessened and performance patterns are more clearly shown when measurements are aggregated across subsets of texts. The system's real-world applicability is correctly reflected by the random selection of texts from many formal and informal correspondences, which guarantees objective and generalizable outcomes.

5.2 System Evaluation

The accuracy of the information recovery (IE) core and the performance of the end-to-end automation pipeline were the two primary focuses of a thorough examination conducted to assess the impact of the suggested solutions. Ninety-three of the 100 Arabic texts in the tailored dataset used to test the system had at least one e-post address.

5.2.1 Information Extraction Accuracy

Using the usual metrics for accuracy, recall, and F-score, the effectiveness of the NLP pipeline in finding and extracting email addresses was appropriately assessed. Table 1 displays the system's performance variance and a sample of outcomes from 10 classes.

Table 1. Sample results for ten texts

Texts	Precision method	Recall method	F-Measure method
Text 1	91%	92%	91%
Text 2	92%	94%	93%
Text 3	87%	79%	83%
Text 4	100%	83%	91%
Text 5	88%	86%	87%
Text 6	89%	84%	86%
Text 7	93%	93%	93%
Text 8	96%	96%	96%
Text 9	91%	95%	93%
Text 10	87%	100%	93%

The system had an average accuracy of 91.4%, an average recall of 90.2%, and an average F-score of 90.7% when comparing the findings throughout the whole dataset. These results confirm the robustness of the hybrid (lexical-morphological-syntactical) approach to understand Arabic text and detect key information, Fig. 4.

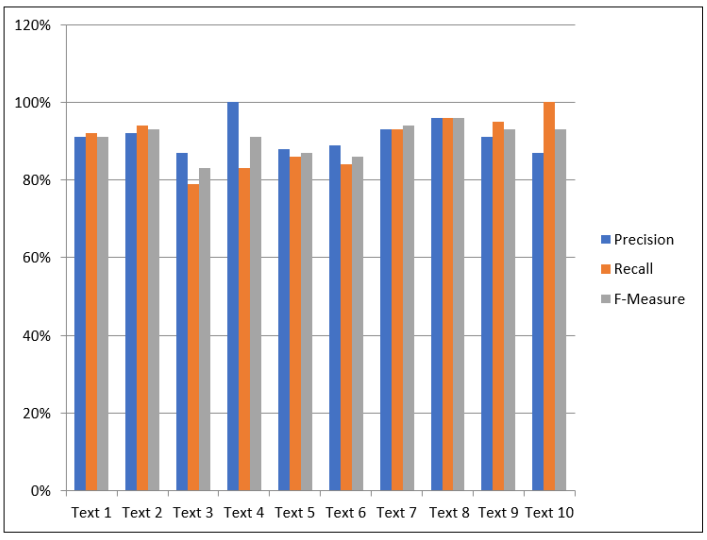

Figure 4. Comparison of sample results for ten texts

As shown in Table 2, a more detailed analysis was performed by classifying texts by source and linguistic style. This reveals an important insight into the system's performance under different conditions.

Table 2. Average results for text group categories

Text Category	No. of Texts	Avg. Precision	Avg. Recall	Avg. F-Measure
Formal Correspondence	50	94%	92%	93%
Educational Texts	30	89%	85%	87%
Social Messages	20	82%	78%	80%
Total/Average	**100**	**91.40%**	**90.20%**	**90.70%**

According to the data, the system works remarkably well with formal, structured texts. Nonetheless, there is a discernible drop in performance while using instructional and—more importantly—social literature. This gradient shows how much the existing system depends on formal language structures. It is strongly correlated with the rise in linguistic complexity, informal language use, and dialectal variances in these categories.

5.2.2 End-to-End Automation Success

The ultimate objective of this research is actionable automation, even though IE correctness is crucial. As a result, we included end-to-end success rate as a higher-level statistic. Only when the system accurately retrieves the email address and generates and sends a confirmation post to that address is a test deemed successful. The system achieved complete, automated success in 79 out of the 93 texts that contained emails, resulting in an end-to-end success percentage of 84.9%. The email creation and validation step is primarily to blame for the discrepancy between this rate and the high F-target (90.7%), because the local section for named units was formatted incorrectly (e.g., faiezmusa@mtu.edu.iq rather than faiez.musa@mtu.edu.iq).

5.2.3 Error Analysis and Limitations

Qualitative analysis of errors was performed to identify system restrictions. The errors were classified as:

- Lexicon-gap error (~ 60% error): The system failed to find email addresses when indicated in words that are not present in the lexicon, such as the colloquial word "ļbo: e-mail). This was the main reason for the low storage of social messages.
- Morphological complexity failure (~ 25% error): In some cases, the morphological analyzer could not segment words with complex or unusual prefixes/suffixes, which led to failure to identify the rotor and its role in the sentence.
- Syntactic ambiguity (~ 15% of the errors): The syntactic analyzer sometimes misclassifies the grammatical role of a word in a complex sentence, leading to the incorrect association of a name with a domain.

The shortcomings covered in the paper—mainly the reliance on a small lexicon and the lack of a reliable semantic analyzer to address contextual obscurity—are explicitly validated by this mistake analysis.

5.2.4 Comparative and Ablation Analysis

To put our results in context, a baseline comparison was made. A simple regular expression (regex) pattern was used to match the same data set. Technical strings like e-post and version numbers like V2.1.0 were often misread by Regex, which led to almost perfect recall (98%) but very poor accuracy (47%). The main advantage of our technology is its great accuracy, which is essential for automation to stop spam from getting to the wrong address. Additionally, a separate study was conducted without the morphological analyzer. This resulted in a significant 15% drop in overall recall, demonstrating how important this component is for controlling the root-and-pattern morphology of the Arabic language.

5.2.5 Comparison with Modern Methods

Phishing detection, generic Arabic information extraction, and sentiment and topic categorization are the three primary categories of recent study in Arabic email and text processing. Khalil et al. assessed four conventional classifiers (KNN, SVM, LR, and NB) using three clustering techniques on a HARD dataset in the domains of sentiment and subject classification. They showed

that appropriate preprocessing and clustering greatly enhance the performance of Arabic text classification [8]. In order to categorize Arabic business emails, Al-Masri and Al-Jabi suggested deep learning-based models that achieved an accuracy of about 92% with a loss of less than 8% using distinct CNN configurations trained on urgency, sentiment, and topic [10]. Despite achieving high classification accuracy, the main objective of these algorithms is to categorize emails rather than extract actionable entities or automate processes based on the collected data.

A second area of research focuses on phishing and security-related email analysis. Al-Suwailemi presented a hybrid machine learning framework combined with a genetic feature-selection algorithm to improve the detection of Arabic phishing emails, demonstrating that optimized feature sets can significantly enhance detection performance on a categorized legitimate/phishing email dataset [9]. Ong et al. and Da Silva et al. conducted a study and created URL- and heuristic-based phishing prediction methods that have extremely high recall but often have poorer accuracy because of overgeneralized patterns [15, 16]. Larger studies, like those by Saloum et al. and Mukherjee et al., have thoroughly examined the automated detection of phishing emails using natural language processing (NLP), emphasizing the expanding role of NLP features and deep models, with a primary focus on binary or multi-class detection rather than fine-grained information extraction and subsequent automation [18, 20].

Arabic information extraction and generic NLP pipelines constitute a third field of related study. In order to extract information from Arabic hadith writings, Helmy and Daoud suggested an intelligent software that showed how supervised learning and domain knowledge may be used to achieve high accuracy without the need for machine translation [11]. Zakaria et al. introduced a technique that uses named entity recognition and syntactic analysis to extract semantic representations from Arabic text into RDF [12]. Despite Arabic's morphological complexity, a strong integration environment can be created for it, as demonstrated by Hackery et al.'s description of the automatic extraction of events from Arabic texts and Mohammed et al.'s presentation of an open Arabic information extraction system based on dependent analysis [13, 14]. More generally, Salloum et al. stressed the significance of NLP-based pipelines in identifying fraudulent emails, while Kurana et al. gave a summary of current developments and difficulties in natural language processing [19, 20].

In contrast to these contemporary techniques, the suggested intelligent software aims for a different but complementary goal: it closes the loop by automatically creating and verifying an email address for the recovered text in addition to extracting information from Arabic texts. Unlike purely classification systems that categorize emails or detect phishing attempts [8]–[10], [15], [16], [18], [20], our approach implements a hybrid pipeline combining lexical, morphological, and syntactic analysis specifically designed for Arabic, followed by entity aggregation and email construction, and is evaluated end-to-end on a custom dataset of 100 texts. The average accuracy achieved (91.4%), recall (90.2%), and F-score (90.7%) demonstrate that the system is capable of competing with the latest natural language processing (NLP) techniques, while handling a more complex task ranging from low-level linguistic analysis to high-level email automation in real-world Arabic communication scenarios [8]– [14], [19], [20].

6. CONCLUSION

Arabic is a powerful language in terms of morphology because of the abundance of letters that come before and after words. Nevertheless, this has a detrimental impact on the automatic morphological analyzer. As a result, it will have an impact on the morphological analyzer's output, which will have an impact on the classification of sentences. A significant obstacle to the system was the absence of a trustworthy semantic source for interpreting word semantics. This was because most other languages relied on publicly accessible semantic sources found online, which made semantic analysis much easier. The source is nonexistent in Arabic, yet it is a popular reference in those languages.

Because of its excellent performance, feature collection technology is regarded as one of the most potent and successful methods for natural language processing. The results of the very successful information extraction technique ranged from 86% to 100%. Several measures can be considered to further expand our work in the future, such as using larger datasets, covering dialect variation, and using vocal commands to assist with e-mail writing and editing in Arabic.

CONFLICT OF INTEREST

The authors declare that there is *no conflict* of interest regarding the publication of this paper.

REFERENCES

[1] S. Al-Azzawy and F. M. L. Al-Rufaye, "Arabic words clustering by using K-means algorithm," 2017 Annual Conference on New Trends in Information & Communications Technology Applications (NTICT), Baghdad, Iraq, 2017, pp. 263-267, doi: 10.1109/NTICT.2017.7976098.

[2] S. Mahmood and F. M. L. Al-Rufaye, "Arabic text mining based on clustering and coreference resolution," 2017 International Conference on Current Research in Computer Science and Information Technology (ICCIT), Sulaymaniyah, Iraq, 2017, pp. 140-144, doi: 10.1109/CRCSIT.2017.7965549.

[3] A. Petukhova, J. P. Matos-Carvalho, and N. Fachada, "Text clustering with large language model embeddings," International Journal of Cognitive Computing in Engineering, vol. 6, pp. 100–108, 2025, doi: 10.1016/j.ijcce.2024.11.004.

[4] G. Gunawan, I. Hosea, E. I. Setiawan, and K. Fujisawa, "Performance Analysis of End-to-End Neural Coreference Resolution in English and Indonesian Texts," Int. J. Intell. Eng. Syst., vol. 17, no. 3, pp. 290–313, Jun. 2024, doi: 10.22266/ijies2024.0630.24.

[5] D. O. Cajueiro, A. G. Nery, I. Tavares, M. K. De Melo, S. A. dos Reis, L. Weigang, and V. R. R. Celestino, "A comprehensive review of automatic text summarization techniques: method, data, evaluation and coding," arXiv preprint arXiv:2301.03403, 2023, doi: 10.48550/arXiv.2301.03403.

[6] H. Zhang, P. S. Yu, and J. Zhang, "A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models," ACM Computing Surveys, vol. 57, no. 11, art. 277, pp. 1–41, 2025, doi: 10.1145/3731445.

[7] C. Park, J. Lim, J. Ryu, H. Kim, and C. Lee, "Simple and effective neural coreference resolution for Korean language," TRI Journal, vol. 43, no. 6, pp. 1038–1048, 2021, doi: 10.4218/etrij.2021-0129.

[8] H. F. Khelil, M. F. Ibrahim, H. A. Hussein, and R. K. Naser, "Evaluation of Different Stemming Techniques on Arabic Customer Reviews," Journal of Techniques, vol. 6, no. 2, pp. 1–8, 2024. doi: 10.51173/jt.v6i2.2313.

[9] A. A. Alsuwaylimi, "Enhancing Arabic Phishing Email Detection: A Hybrid Machine Learning Based on Genetic Algorithm Feature Selection," Int. J. Adv. Comput. Sci. Appl., vol. 15, no. 8, pp. 312–325, 2024. doi: 10.14569/IJACSA.2024.0150833.

[10] Masri and M. Al-Jabi, "A novel approach for Arabic business email classification based on deep learning machines," PeerJ Comput. Sci., vol. 9, e1221, 2023. doi: 10.7717/peerj-cs.1221.

[11] X. Qu, Y. Gu, Q. Xia, Z. Li, Z. Wang, and B. Huai, "A survey on Arabic named entity recognition: past, recent advances, and future trends," IEEE Transactions on Knowledge and Data Engineering, vol. 36, no. 3, pp. 943–959, 2024, doi: 10.1109/TKDE.2023.3303136.

[12] Zakria, M. Farouk, K. Fathy, and M. N. Makar, "Semantic Representation Extraction from Unstructured Arabic Text," in Proc. 2019 8th Int. Conf. Software and Information Engineering (ICSIE '19), New York, NY, USA: ACM, 2019, pp. 222–226. doi: 10.1145/3343023.3343046.

[13] E. Hkiri, S. Mallat, and M. Zrigui, "Events automatic extraction from Arabic texts," in Natural Language Processing: Concepts, Methodologies, Tools, and Applications, Information Resources Management Association (IRMA), Ed. Hershey, PA, USA: IGI Global, 2020, pp. 1686–1704, doi: 10.4018/978-1-7998-1502-6.ch082.

[14] S. Mohamed, M. Hussein, and H. Mousa, "Arabic open information extraction system using dependency parsing," Int. J. Electr. Comput. Eng., vol. 12, no. 1, pp. 541–551, 2022. doi: 10.11591/ijece.v12i1.pp 541-551.

[15] A. J. C. Trappey, C. V. Trappey, J.-L. Wu, and J. W. C. Wang, "Intelligent compilation of patent summaries using machine learning and natural language processing techniques," Advanced Engineering Informatics, vol. 43, p. 101027, Dec. 2019, doi: 10.1016/j.aei.2019.101027.

[16] M. R. D. Silva, E. L. Feitosa, and V. C. Garcia, "Heuristic-based strategy for phishing prediction: A survey of URL-based approach," Comput. Secur., vol. 88, Jan. 2020, doi:

10.1016/j.cose.2019.101628.

[17] S. P. Panda, "The evolution and defense against social engineering and phishing attacks," International Journal of Science and Research (IJSR), vol. 14, no. 5, pp. 397–408, May 2025, doi: 10.21275/sr25504223645.

[18] Md. F. Rabbi, A. I. Champa, and M. F. Zibran, "Phishy? Detecting phishing emails using machine learning and natural language processing," in Studies in computational intelligence, 2024, pp. 119–137. doi: 10.1007/978-3-031-55174-1_9.

[19] A. Khurana, K. Koli, K. Khatter, and M. Singh, "Natural language processing: state of the art, current trends and challenges," Multimedia Tools and Applications, vol. 82, pp. 3713–3744, 2023, doi: 10.1007/s11042-022-14547-6.

[20] S. Salloum, T. M. A. Gaber, S. Vadera, and K. Shaalan, "A systematic literature review on phishing email detection using natural language processing techniques," IEEE Access, vol. 10, pp. 65703–65727, 2022. doi: 10.1109/ACCESS.2022.3184355.

[21] E. Kummerfeld and D. Klein, "An analysis of coreference evaluation metrics," Transactions of the Association for Computational Linguistics, vol. 8, pp. 499–514, 2020, doi: 10.1162/tacl_a_00328.