Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

Optimizing n-gram lengths for cross-linguistic text classification: A comparative analysis of English and Arabic morphosyntactic structures

Boumedyen Shannaq *

Management Information System Department, University of Buraimi, Al Buraimi, Oman

ARTICLE INFO

Article history:
Received 15 November 2024
Received in revised form
24 March 2025
Accepted 23 April 2025

Keywords:
N-gram length
Text classification
English language
Arabic language
Morphological features

ABSTRACT

This paper investigates the impact of n-gram length on text classification in English and Arabic, two languages with different writing systems. The study aims to examine how language characteristics influence the optimal n-gram length for text classification. The English dataset comprises 4,450 articles categorized into business, technology, entertainment, sports, and politics, with 2,225 records used for training and 2,225 for testing. The Arabic dataset includes 5,000 randomly selected documents from a total of 111,728 documents. The findings indicate that for English text classification, 2-grams provide the best performance with a precision of 0.482, recall of 0.489, and F1 score of 0.472. In contrast, Arabic text classification achieves optimal performance with 6-grams, reaching an F1 score close to 0.85. These results highlight that language-dependent morphological and syntactic features can significantly affect the performance of n-gram-based models. This study provides valuable insights for enhancing language-sensitive text classification techniques, particularly for accurately and efficiently categorizing documents in different languages.

1. Introduction

Most of the techniques of natural language processing, including text classification, now depend on n-gram-based models because of their simplistic nature as well as their efficiency in terms of capturing contextual features (Khurana et al., 2023). However, there is no unified study in the existing literature that measures the effect of n-gram length on classification accuracy where different languages with different morphological and syntactic properties, such as English and Arabic, are included. The research conducted by Shannaq et al. (2024) highlighted Arabic and English text preprocessing techniques for user classification, optimizing machine learning models to strengthen faculty password policies effectively. It was quite useful and really gained a lot of practical and valuable benefits to understand the Arabic and English text as well; for other languages, it might assist in the advancement of the development of an intelligent information system using AI and machine learning to analyze English and Arabic course registration texts for the automation of processes to improve performance. To analyze English and Arabic course registration texts, automating processes to enhance academic performance (Shannaq and Al-Zeidi, 2024).

Prior works have concentrated mainly on English or, more broadly, analytic languages with less complex morphology; this practice utilizes n-gram lengths of 1–3. This work proposes a new comparative approach using various n-gram sizes from 2 to 10 for English and Arabic texts to understand how specific features affect the classification.

The research problem is "What influence does the nativeness of language on certain characteristics reach for the suitable n-gram length in text classification?" To the best of the authors' knowledge, few studies have been conducted to analyze n-gram length distinctions with reference to the dissimilar syntax of the two languages, English and Arabic, comprehensively and systematically. This gap prevents the building of optimized, language-sensitive text classification models, especially for morphologically rich languages such as Arabic. This systematic review of Arabic text classification (ATC) highlights key methodologies, challenges, and trends, analyzing 60 studies across sectors, tasks, and phases, and recommends improved datasets and preprocessing techniques for future research (Wahdan et al., 2024). Another

* Corresponding Author.
Email Address: boumedyen@uob.edu.om
https://doi.org/10.21833/ijaas.2025.04.015
Corresponding author's ORCID profile:
https://orcid.org/0000-0001-5867-3986

related work in this path performs review analyses for Arabic sentiment analysis techniques, examining 100 publications to identify optimal methods (Al Katat et al., 2024). Deep learning outperforms other models, especially with multi-level embedding, enhancing classification accuracy. Which focuses on optimizing n-gram lengths for English and Arabic classification, this study highlights key distinctions between sentiment analysis and structural text classification (AlMahmoud and Hammo, 2024). The n-gram study offers insights into language-specific morphosyntactic structures, whereas this sentiment analysis work refines deep learning-based accuracy through multi-level embeddings, addressing Arabic's unique linguistic challenges. Feature selection (FS) enhances text classification by isolating essential features, optimizing accuracy, and reducing complexity. This study reviews 108 meta-heuristic (MH) FS methods from 2015-2022, finding MH techniques superior to traditional methods, with promising approaches like Ringed Seal Search (RSS) (Al-Shalif et al., 2024).

In contrast, the proposed work emphasizes refining n-gram length for better classification accuracy across languages, focusing on how morphosyntactic variations in English and Arabic require different n-gram selections for optimal performance, rather than feature isolation.

The research question is “In what way do the morphological and syntactic features of English and Arabic affect the choice of the best n-gram size for the text classification task?”

In this paper, the research question will focus on investigating the impact of varying the n-gram size in relation to the accuracy of the text classification between English and Arabic texts.

- It aimed at determining the appropriateness of n-gram lengths in the morphological distinction between languages like Arabic and analytical languages like English.
- In order to provide an estimation regarding the specific influence of language-specific morphosyntactic structures on classification performance.
- It is also a best practice guide for multilingual Text Classification, opinion mining, and Information Retrieval applications.

This work enriches the knowledge of how the morphological and syntactic characteristics of texts affected their classification using n-gram algorithms. The proposed results that 2-grams perform well for English while 6-grams for Arabic gives specific guidelines for text classification enhancement. All of these enhancements promote applicability in the fields of the classification of content in environments that encompass multiple languages, sentiment analysis for content categorization, and information retrieval systems; especially in the realms of education and government where multilingual systems demand high accuracy.

2. Literature review

The rapid growth of textual data presents significant management challenges, notably due to storage and processing costs. Text classification, key to text mining, enables efficient data categorization and insight extraction. This survey introduces a research field-based taxonomy, using empirical and experimental evaluations to enhance understanding and decision-making in classification techniques (Taha et al., 2024). Research by Shannaq (2024b) found that Arabic students often switch between Arabic and English typing on mobile keypads, reflecting their bilingual adaptability and the influence of smartphone integration in academic settings. The study by Shannaq et al. (2019) explored English and Arabic text analysis in a Management Information System, enhancing computer learning for predicting material quantities effectively in multilingual contexts.

Related work proposed framework includes knowledge distillation and text cleansing. It aims to enhance English word prediction by employing higher-order N-grams (4-grams, 5-grams) to capture broader language dependencies, overcoming traditional N-gram limitations in context range. Using large English corpora, this method addresses sparsity challenges with smoothing techniques, improving analysis, prediction, and sentiment accuracy, contributing to NLP advancements (Kumar and Thirumaran, 2024). This makes efficient text classification algorithms essential for addressing the growing problem of cyberbullying. In this study, both TF-IDF and n-gram techniques—using unigrams, bigrams, and trigrams—are applied to classify cyberbullying texts, with particular attention to how parameter changes are affected by text length.

The results close gaps in previous approaches and support a more flexible prevention of online harassment as n-gram proves useful in detecting harassment patterns, as well as the improved generation of term value enhancements (Setiawan et al., 2024).

Our work looks at the effects of language on n-gram for English and Arabic text classification. Using both English and Arabic data, we empirically demonstrate that while English gets the highest F1 score by using only 2-grams, Arabic gets the same level of F1 ~0.85 by using 6-grams Identification of the structure of language in the context of text categorization shows language sensitivity of the proposed methodology, which helps in enhancing the text categorization across different languages.

In contrast, the proposed study examines how N-gram length affects the accuracy of classification in Arabic and English taking into account syntactic and morphological differences between the two languages. Language-sensitive improvements for precise effective document categorization across languages are highlighted by the fact that Arabic performed best with 6-grams (F1 score ~0. 85) while English classification was optimal with 2-grams. Shannaq (2024a) leveraged TF-IDF to extract

unique password features, evaluating six machine-learning models for enhanced password-based authentication. Naive Bayes achieved the highest precision (96.38%).

Another related study introduces SEWAR, an algorithm that extracts high-quality multiword terms (MWTs) from Arabic medical texts using FastText, supporting medical question classification and broader NLP tasks. Evaluated by PMI, cosine similarity, and clustering purity, SEWAR shows promising results (AlMahmoud and Hammo, 2024). A related study reviews recent advancements in implementing artificial neural networks (ANN) and natural language processing (NLP) during construction budgeting. It assesses task scope and client expectations, emphasizing the need for improved data sets and algorithms to enhance budgeting automation (Jacques De Sousa et al., 2024). A related study presents the concept of mismatches in Q&A sites due to increasing user engagement. It compares morphological analysis and N-gram methodologies, showing that 4-gram achieves comparable accuracy to morphological analysis for English Q&A statements through multiple regression analysis (Yokoyama, 2024). Another related study proposes an improved word embedding model based on pre-trained Word2vec, addressing the limitations of static and dynamic models. It introduces Term Document Frequency (TDF), positional encoding, and an adaptive segmentation model, achieving superior accuracy and efficiency in text classification (Sun et al., 2024). A related study examines statistical challenges faced by university students, particularly those accessing the Maths Support Centre at University College Dublin. Analyzing data since 2015/16, it categorizes issues, revealing difficulties with university-level concepts, and aiding in developing educational resources for undergraduates (De La Hoz-Ruiz et al., 2024). A related study introduces SLIME, an explainable LLM method that identifies lexical features of Alzheimer's Disease from spontaneous speech, enhancing understanding of speech production impairments while improving confidence in LLM applications for neurological disorders (Ribeiro et al., 2024). The exponential data growth demands effective text classification for key insights, driving research in classical and deep learning methods. A related study presents a novel taxonomy, hierarchically classifying text algorithms into fine-grained categories, enabling precise evaluation and ranking across techniques (Taha et al., 2024). Another study addresses query-document vocabulary mismatch in information retrieval by proposing automatic query expansion (AQE) techniques. Utilizing median vectors from deep networks enhances query similarity and integrates with the BM25 model, outperforming baseline methods in experimental evaluations. Tokenization, crucial in NLP, splits text into tokens for processing. While word tokens are common, sub-word and character tokens offer alternatives, especially for complex languages like Arabic. Another related study assesses character trigrams for Arabic sentiment analysis and text classification, focusing on misspellings. Character trigrams maintain manageable vocabulary sizes, enhancing model stability, while Word-Piece and word embedding struggle with misspellings, experiencing up to 19% performance declines.

In contrast, character trigram models demonstrate stable performance, dropping only 0%-8% under similar conditions. Analysis in the literature confirms the robustness of character trigrams for challenging datasets compared to traditional tokenization approaches (Alomari and Ahmad, 2024). The rise of AI text applications highlights the need for advanced Arabic NLP tools. This study classifies Arabic customer reviews using the HARD dataset, employing machine and deep learning (CNN, RNN, NB, LR). With Snowball Stemmer and N-gram, CNN achieved 93.5% accuracy, revealing dialectal challenges and feature extraction impact. Results support further Arabic NLP exploration as stated by Alshammary et al. (2024). A related study evaluates Arabic Text Classification using machine learning with preprocessing and feature selection techniques. Classifiers like SGDC and LSVC outperform others on benchmark datasets, influenced by preprocessing and representation models (Masadeh et al., 2024). A systematic review of Arabic hate speech identification in tweets highlights current research trends, common language varieties, classification techniques, and key findings from 24 studies (2018-2023), guiding future model improvements and applications (Alhazmi et al., 2024).

Some of the current developments made in multilingual text classification imply that the use of n-gram-based models is quite viable given the nature of the different varieties of the language (Gurgurov et al., 2024; Fazal and Farook, 2024). N-grams, which represent patterns of contextual regularities in text, are particularly useful for morphologically complex languages such as Arabic, where word variations make modeling systems more challenging (Yusuf et al., 2024). Studies by Sabty (2024) have emphasized the adaptability of n-gram models in multilingual contexts, demonstrating their superior performance in distinguishing syntactic and semantic nuances between Arabic and English texts.

However, addressing multilingual challenges involves trade-offs, such as balancing the time required to extract features from text with the need to preserve meaningful content. Miah et al. (2024) investigated these challenges in cross-lingual sentiment analysis and demonstrated that n-grams can significantly improve feature representation while maintaining high classification accuracy. Similarly, Wahdan et al. (2024) and Moniri et al. (2024) highlighted the potential of n-gram models in improving language-specific search engine results, particularly in Arabic, where higher-order n-grams effectively capture contextual dependencies. Following these works, our approach uses n-gram models with enhanced functionality designed for

multilingual data, which takes into account language-specific issues but enforces cross-lingual optimization. It is done so to establish the novelty of the present work and its significance in terms of its ability to further the state of the art of multilingual text classification. Comparative studies on multilingual text classification in recent literature have highlighted the effectiveness of n-gram-based methods across different languages. These studies suggest that n-grams, which capture contextual information in text, are particularly valuable for languages with complex morphology, such as Arabic, where word form variations are common. As mentioned by Sindane and Marivate (2024), the n-gram models have highlighted that these models are transferable into the multilingual environment by performing higher accuracy in differentiating syntactic and semantic differences between the languages.

However, multilingual applications involve trade-offs, such as balancing effective feature extraction with the preservation of semantic similarity. Svete et al. (2024) explored these challenges and found that transformers can generate n-gram language models and, in some cases, outperform traditional methods. This study builds on previous findings by using n-gram models designed for multilingual datasets, taking into account the specific features of each language while supporting cross-lingual use. The broader literature review shows the originality of this work and its potential to advance research in multilingual text classification.

3. Methodology

To examine the impact of n-gram length on classification performance in English and Arabic, Fig. 1 describes the methodological steps.

3.1. Data collection

In selecting example documents, we used both English and Arabic datasets. The documents were randomly selected and evenly distributed across classes to ensure balanced representation within each domain. The English dataset consists of replicated data originally collected from the Kaggle website. It includes 2,225 labeled text documents categorized into five classes: business, technology, entertainment, sports, and politics. The dataset contains two features—text and label—with 2,225 rows and two columns. The 'text' feature includes the content of each document, while the 'label' identifies its category.

A second English dataset, also containing 2,225 articles classified into the same five categories, was collected from kaggle.com/competitions/learn-ai-bbc. For the Arabic dataset, the study randomly selected 5,000 documents from a total of 111,728 available documents across five categories. These texts, comprising approximately 319 million words, were collected from three online newspapers using a semi-automated web crawling approach.

3.2. Preprocessing

The preprocessing stage involved morphological processing, such as stop-word removal and the elimination of inflections, to preserve essential meaning in the text. Text preprocessing is one of the initial steps in the text mining process, which prepares raw textual data for machine learning algorithms by converting it into a suitable format for analysis. This step helps retain meaningful content while removing noise, allowing models to operate with greater precision and accuracy. Table 1 presents examples of preprocessing applied to both English and Arabic texts.

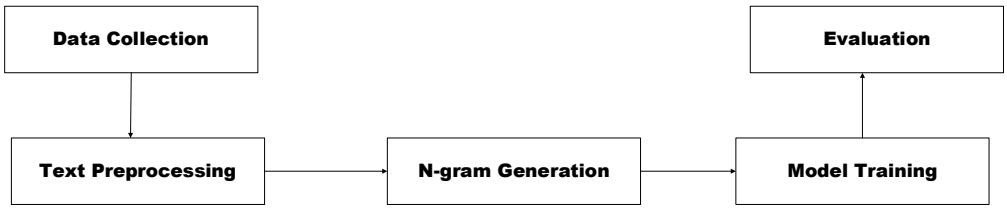

Fig. 1: Research methodology

Table 1: Examples of preprocessing from English and Arabic

Step	English example	Arabic (romanized) example	Description
Original sentence	The student goes to school every day	Al-talib yadhhab ila al-madrasah kul yawm	Raw text before any preprocessing
Tokenization	['The', 'student', 'goes', 'to', 'school', 'every', 'day']	['Al-talib', 'yadhhab', 'ila', 'al-madrasah', 'kul', 'yawm']	Breaking the text into individual words or tokens
Stop-word removal	['student', 'goes', 'school', 'day']	['talib', 'yadhhab', 'madrasah', 'yawm']	Removing common, less meaningful words (e.g., "ila")
Stemming	['student', 'go', 'school', 'day']	['talib', 'dhahab', 'madrasah', 'yawm']	Reducing words to their root or base forms (e.g., "yadhhab" to "dhahab")
N-gram generation (2-grams)	[('student', 'go'), ('go', 'school')]	[('talib', 'dhahab'), ('dhahab', 'madrasah')]	Creating combinations of consecutive tokens for context representation

3.2.1. Detailed of preprocessing explanation

A. Morphological Processing: The preprocessing phase begins with morphological processing, which involves converting text into tokens and removing words whose roots are considered stop words. Arabic, in particular, has complex morphology, presenting challenges like prefixes, suffixes, and infixes. For example, the Arabic word (al-talib) is

broken down to its root (talab). Similarly, in English, words like 'going' are reduced to their root form 'go.'

B. N-gram Generation: For each language dataset, we generate n-grams ranging from two to ten characters long. This allows us to evaluate the classification algorithm's performance across different input sizes. N-grams represent sequences of tokens where n varies between 2 and 10. For instance, in English, the words "student" and "goes" form a 2-gram based on the phrase "student goes." In Arabic, examples include (talib) and (dhahaba) from the sentence (al-talib yadhhab). This step helps capture relationships between words in specific contexts.

C. Model Training: The datasets are transformed using vectors of n-grams, where each vector is weighted by TF-IDF (Term Frequency-Inverse Document Frequency) to highlight unique terms within the dataset. The English and Arabic datasets are then separately used to train a logistic regression classifier to assess performance independently for each language.

D. Evaluation: For each n-gram configuration, we calculate Precision, Recall, and F1 Score to determine the best configuration for classification. Due to the morphological nature of Arabic, longer n-grams are generally more effective, whereas English can be effectively processed with shorter n-grams. This preprocessing approach addresses language differences, enhancing model accuracy.

4. Experiment and results

The proposed experiment is presented in Fig. 2.

4.1. Experimental steps

1. Define language-specific preprocessing techniques:

- English text was processed with tokenization and stop-word removal.
- Arabic text preprocessing accounted for root-based morphological structures, applying techniques like root extraction and affix removal.

Fig. 3 demonstrates a screen shoot of the Language-specific Preprocessing Techniques using Python.

2. TF-IDF feature extraction for n-grams:

- Calculated TF-IDF for each n-gram to standardize the importance of words within a document and improve classification accuracy.

Fig. 4 describes the sample code for processing the selected top N words for each document in the testing dataset.

3. Model training and testing:

- Models were trained using a logistic regression classifier, with performance evaluated across various n-gram lengths.

4. Comparison of performance by n-gram length:

- For each language, the n-gram values for n=2 to 10

Table 2 shows a selection of generated documents that match the top two most frequent English words. Table 3 displays the top three words, and Table 4 shows the top eight words. This process was repeated for all n-grams from 2 to 10. Similarly, Table 5 presents a portion of generated documents matching the top two most frequent Arabic words, while Table 6 shows the top six words. The same process was repeated for all n-grams from 2 to 10. Table 7 presents the obtained results after running the experiments of Top N Words (n-grams) from 2 to 10 for English text. Table 8 presents the obtained results after running the experiments of Top N Words (n-grams) from 2 to 10 for Arabic text.

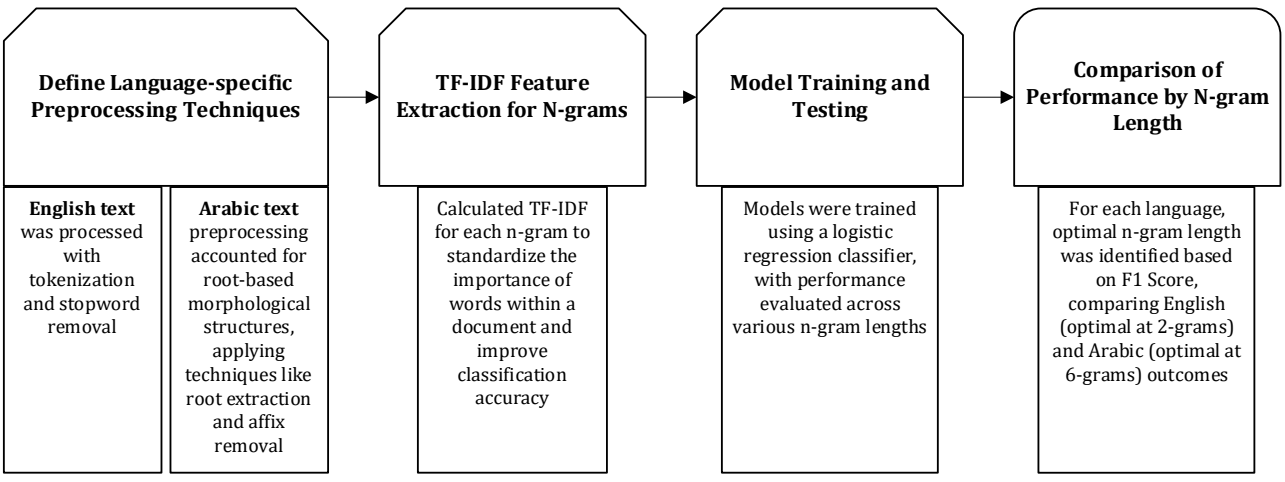

Fig. 2: Experiment

```
# For English
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import SnowballStemmer
from nltk import download
import re
# Download necessary NLTK resources
download('punkt')
download('stopwords')
# Function to preprocess English text
def preprocess_english(text):
    # Tokenization
    tokens = word_tokenize(text.lower())
    # Stop word elimination
    stop_words = set(stopwords.words('english'))
    filtered_tokens = [word for word in tokens if word.isalnum() and word not in stop_words]
    # Stemming
    stemmer = SnowballStemmer('english')
    stemmed_tokens = [stemmer.stem(word) for word in filtered_tokens]
    return stemmed_tokens
```

Fig. 3: Language-specific preprocessing techniques

```
import pandas as pd
from sklearn.metrics.pairwise import cosine_similarity
# Get the top N words for each document in the testing dataset
def get_top_n_words(df, top_n):
    return df.sort_values(by='Weight', ascending=False).groupby('Document').head(top_n)
# Create a similarity function based on the cosine similarity of word weights
def cosine_similarity_for_words(doc1_words, doc2_words):
    merged = pd.merge(
        doc1_words[['Word', 'Weight']],
        doc2_words[['Word', 'Weight']],
        on='Word', how='outer', suffixes=('_train', '_test')
    ).fillna(0)
    vector1 = merged['Weight_train'].values.reshape(1, -1)
    vector2 = merged['Weight_test'].values.reshape(1, -1)
    similarity = cosine_similarity(vector1, vector2)
    return similarity[0][0]
# Function to evaluate classification based on top N words
def evaluate_classification(training_sampled, testing_top_n, top_n):
    result_rows = []
    for train_doc in training_sampled['Document'].unique():
        train_doc_words = training_sampled[training_sampled['Document'] == train_doc]
        similar_docs = []
        for test_doc in testing_top_n['Document'].unique():
            test_doc_words = testing_top_n[testing_top_n['Document'] == test_doc]
            similarity_score = cosine_similarity_for_words(train_doc_words, test_doc_words)
            test_doc_class = test_doc_words['Class'].values[0]
            similar_docs.append((test_doc, test_doc_class, similarity_score))
        # Sort by similarity and select top 4
        similar_docs = sorted(similar_docs, key=lambda x: x[2], reverse=True)[:4]
        result_row = {'Class/Doc': train_doc}
        for i, (test_doc, test_class, _) in enumerate(similar_docs, start=1):
            result_row[f'Test_{i}_class'] = test_class
        result_rows.append(result_row)
    return pd.DataFrame(result_rows)
# Experiment loop: Evaluate for top N words (2 to 10)
results = {}
for top_n in range(2, 11):  # Top 2 to 10 words
    testing_top_n = get_top_n_words(testing_df, top_n)
    result_df = evaluate_classification(training_sampled, testing_top_n, top_n)
    results[top_n] = result_df
```

Fig. 4: Sample code

Table 2: TOP English 2 words

Class/doc	Doc1_class	Doc1_top_words	Doc2_class	Doc2_top_words
Doc03140_sport	Doc00712_Sport_Sport	capriati, melbourn	Doc03163_sport_sport	melbourn, champion
Doc03448_sport	Doc00784_Sport_Sport	wenger, almunia	Doc00749_Sport_Sport	almunia, lehmann
Doc00546_Sport	Doc00888_Sport_Sport	owen, captain	Doc03440_sport_sport	owen, real
Doc01300_Technology	Doc00989_Technology_Technology	bluetooth, virus	Doc04141_Technology_Technology	handset, recondit
Doc03482_sport	Doc03250_sport_sport	arsenal, fabrega	Doc00453_Sport_Sport	arsenal, rey

Table 3: Top English 3 words

Class/doc	Doc1_class	Doc1_top_words	Doc2_class	Doc2_top_words
Doc03128_sport	Doc03624_sport_sport	athen, kenteri, charg	Doc00792_Sport_Sport	kenteri, test, athen
Doc03782_politics	Doc00118_Politics_Politics	eu, chirac, eastern	Doc00269_Politics_Politics	eu, straw, propaganda
Doc02729_business	Doc02806_business_business	yuko, oil, yuganskneftega	Doc02821_business_business	yuko, gazprom, russian
Doc00083_Politics	Doc03835_politics_politics	women, childcar, matern	Doc00400_Politics_Politics	kennedi, matern, parti

Table 4: Top English 8 words

Class/doc	Doc1_class	Doc1_top_words	Doc2_class	Doc2_top_words
Doc04253_Technology	Doc01162_Technology_Technology	halo, explos, cope, increas, broadband, provid, bandwidthhungri, schroth	Doc01145_Technology_Technology	traffic, net, data, ddos, capella, said, gambl, pipex
Doc03080_business	Doc02047_Business_Business	unit, club, manchest, mr, approach, support, debt, takeov	Doc03604_sport_sport	club, propos, unit, oppos, subject, confirm, magnier, billionair
Doc01554_Entertainment	Doc02272_entertainment_entertainment	film, award, life, star, boy, caouett, teena, bacon	Doc01696_Entertainment_Entertainment	film, antibush, movi, best, comedi, christ, mel, top
Doc01674_Entertainment	Doc01294_Technology_Technology	robot, walk, advanc, audienc, gait, yoga, beckon, kg	Doc04348_Technology_Technology	robot, robotiquett, hertfordshir, passtheparcel, professor, social, situat, find
Doc00599_Sport	Doc03558_sport_sport	souness, celestin, graem, reopen, surplus, strengthen, specif, advanc	Doc03424_sport_sport	souness, shearer, uefa, aggreg, done, jacki, graem, anyway

Table 5: Arabic text sample of matching_results_top_2_words

Class/doc	Doc1_class	Doc2_class	Doc3_class	Doc4_class
Doc36540_culture	Doc35322_culture_culture	Doc43092_culture_culture	Doc5555_sport_sport	Doc11887_sport_sport
Doc35693_culture	Doc34642_culture_culture	Doc41715_culture_culture	Doc20294_politic_politic	Doc31681_culture_culture
Doc44269_culture	Doc37768_culture_culture	Doc33144_culture_culture	Doc33491_culture_culture	Doc32089_culture_culture
Doc34860_culture	Doc32300_culture_culture	Doc43072_culture_culture	Doc35073_culture_culture	Doc31521_culture_culture

Table 6: Arabic sample of matching_results_top_6_words

Class/doc	Doc1_class	Doc2_class	Doc3_class	Doc4_class
Doc32869_culture	Doc43586_culture_culture	Doc34086_culture_culture	Doc38862_culture_culture	Doc42019_culture_culture
Doc44141_culture	Doc52730_economy_economy	Doc42263_culture_culture	Doc39839_culture_culture	Doc43250_culture_culture
Doc33788_culture	Doc33788_culture_culture	Doc40697_culture_culture	Doc42720_culture_culture	Doc37231_culture_culture
Doc38238_culture	Doc41543_culture_culture	Doc37574_culture_culture	Doc40461_culture_culture	Doc33108_culture_culture

Table 7: Top N Words (n-grams) from 2 to 10 for English text

Top N words (n-grams)	Precision	Recall	F1 score
2	0.482	0.489	0.472
3	0.395	0.411	0.395
4	0.38	0.4	0.384
5	0.367	0.378	0.368
6	0.321	0.333	0.324
7	0.336	0.356	0.344
8	0.326	0.344	0.333
9	0.334	0.356	0.342
10	0.317	0.333	0.324

Table 8: Top N Words (n-grams) from 2 to 10 for Arabic text

Top N words	Precision	Recall	F1 score
2	0.800606061	0.78	0.783494
3	0.798181818	0.78	0.782801
4	0.820454545	0.82	0.81443
5	0.816161616	0.82	0.816541
6	0.842272727	0.84	0.838413
7	0.825909091	0.82	0.820317
8	0.825909091	0.82	0.820317
9	0.805050505	0.8	0.801754
10	0.805050505	0.8	0.801754

4.2. Explanation of findings

2-grams Yield the Highest Performance: The classification results show that 2-grams achieve the best performance, with a precision of 0.482, recall of 0.489, and F1 score of 0.472. This study limited the analysis to n-grams up to size 2, as shorter n-grams provide sufficient information for class decomposition in English without introducing excessive string specificity.

Performance Declines with Larger N-grams: Overall, performance metrics tend to decrease as n-gram size increases. In English, longer n-grams are often too sparse and variable, capturing phrase patterns unique to specific documents, which limits generalization across the dataset.

These findings can be contrasted with previous research in Arabic text classification, where longer n-grams have proven more effective. For example, studies have reported that 6-grams can yield optimal classification results in Arabic, with F1 scores reaching as high as 85%.

This difference in optimal n-gram length can be explained by linguistic characteristics unique to each language:

1. English (optimal n-gram=2):

- Analytic Language Structure: English is now classified as an analytic language with standing lower complexity morphological characteristics. The meanings of words are many a time individual with restricted contextual interdependence, thus, brief n-grams can adequately capture context.
- Syntactic Simplicity: 2-grams are considered sufficient to acquire meaningful structures and relationships so they are more efficient in classification.

2. Arabic (optimal n-gram=6):

- Rich Morphology and Inflectional Complexity: Arabic is considered to be a heavily inflected language where a stem to which affixes can be attached in various forms such as prefixes, infixes, or suffixes indicate such features as tense, gender, and number. To capture this information, longer n-grams (6 words) have to be used in order to capture this information accurately.
- Contextual Dependency and Semantic Units: In Arabic, no less than a complete phrase and sentence to convey an idea, as the language possesses flexibility in the positioning of words and highly complicated morphemes with regard to inflection. Therefore, 6-grams are effective in helping to achieve cohesiveness and identify meaningful semantic structures to be used for classification.

Fig. 5 shows Precision, Recall, and F1 for English and Arabic text documents for various n-grams.ms. An overall pattern observed is that Arabic text yields better extraction accuracy based on n-gram comparisons at every n-value as compared with the

English text. Arabic stabilizes while performance maps for both degrade at the lower n-grams, yet Arabic noticeably outperforms English at the higher n-grams.

Fig. 5: Precision, Recall, and F1 scores comparison for English and Arabic texts across varying n-grams

4.3. Practical implications of findings in NLP applications

The recommendations derived from our work highlight the following practical implications of n-gram-based approaches in NLP tasks pertinent to English and Arabic, respectively. These implications can be applied to sentiment analysis and language-specific Web search, as well as many other multilingual models.

- Sentiment analysis: The study shows that 2-grams are the best in English due to their analytical nature and syntactically shallow profiling. Because of the lack of context, 2-grams are useful for sentiment analysis because they contain important emotional indications while not being long. In Arabic, the morphological complexity and contextual language dependencies translate to 6-grams for encompassing variations in meaning. The discovery of this suggests the need to develop applications where the linguistic structure of the languages involved is taken into account in order to achieve accurate emotion recognition and polarity identification.
- Language-specific search engine: Higher n-grams show better Arabic classifying, which can greatly improve the effectiveness of Arabic niche-related search engine results. Our previous and future work Both shorter and longer n-grams allow more accurate indexing of different variations of words, which makes the query matching set more diverse. For English, we see that in the case of 2-grams, it helps in improving search efficiency with a very low additional computation.
- Multilingual NLP models: To this end, the observed comparatively worse performance of Arabic relative to English underscores such an argument, as language properties should define the underlying framework of pre-trained models like GPT and BERT. Enriching the design with these insights results can support the development of more diverse AI systems that handle Multilingual Morphologically Rich Languages.

4.4. Recommendations

Therefore, for developers addressing these challenges, the design of NLP pipelines should incorporate language-specific optimizations. In practical applications—such as intelligent customer service assistants or text summarization tools—different n-gram lengths may be suitable depending on the language. For example, due to the morphological complexity of Arabic, 6-grams may be more effective, whereas 2-grams are often sufficient for English, which has a simpler linguistic structure.

These enhancements can benefit various industries, including education, media, and governance, by supporting informed decision-making and improving user experiences. Internet-based applications, especially those involving translation, localization, and imported content, stand to gain significantly. Network services, such as AI-driven chatbots for customer request management and sentiment analysis engines, also benefit from optimized language processing.

Government and business systems that rely on textual data can leverage these improvements for better performance. The current study also highlights the effectiveness of Arabic n-gram analysis, confirming previous findings that Arabic outperforms other languages in terms of precision, recall, and F1 score across various n-gram levels.

This research contributes to the broader field of multilingual NLP, with a particular emphasis on low-resource languages like Arabic. The findings can support future research in areas such as style-specific text mining, opinion mining, sentiment analysis, and language-specific text summarization.

The superior performance of Arabic n-grams compared to other languages suggests that there is significant potential to enhance pretrained models such as GPT and BERT for better handling of morphologically rich languages. This aligns with the global movement toward developing more inclusive AI systems.

The findings also have implications for niche AI applications in Arabic, such as virtual assistants, intelligent tutoring systems, active learning models, and automated semantic interpretation systems. Advances in Arabic text processing will be particularly valuable in industries across the Arab world, including education, media, and public administration, by enabling improved decision-making and user engagement.

Moreover, integrating n-gram analysis with deep, externally trained models can improve the performance of NLP systems for low-resource languages. It can also enhance interpretability in text analysis by identifying the specific n-grams responsible for influencing decisions.

This work provides foundational insights that support the progressive development of AI-assisted environments for culturally and linguistically diverse (CLD) populations around the world.

5. Conclusion

This paper investigates the impact of n-gram size (ranging from 2 to 10) on cross-lingual text categorization using datasets in English and Arabic. The study begins with 2-grams, given the extensive prior research involving unigrams, and extends the range up to 10-grams to explore broader contextual patterns. The performance remained stable up to 8-grams but showed a slight decline beyond that point, supporting the decision to limit the maximum n-gram size to 10 in order to maintain system efficiency and reasonable accuracy.

Standard preprocessing techniques were applied, including the removal of stop words, inflected forms, tokenization, and stemming. For the Arabic dataset, specific transformations such as diacritic removal and language-tailored stemming were employed to preserve essential semantic content. These preprocessing methods, along with the implementation code, are provided to facilitate replication.

The study introduces a novel approach by focusing on repeated n-gram patterns rather than analyzing full document texts. To ensure consistency, the same domains and language distributions used in the initial experiment were retained. The evaluation relied on precision, recall, and F1 scores across various n-gram configurations to identify the optimal settings.

Analysis of variance (ANOVA) revealed a significant difference in performance between the two languages. Although Arabic initially showed lower overall performance compared to English, it ultimately demonstrated greater stability and higher accuracy at larger n-gram sizes. This indicates the effectiveness of n-gram optimization for language-specific classification and offers valuable insights into improving multilingual text analysis.

Due to the relatively simple structure and limited contextual variation in English, 2-grams were sufficient for effective classification. In contrast, Arabic classification achieved its best results with 6-grams, which more effectively captures the language's morphological complexity and contextual depth. This finding underscores the importance of accounting for linguistic characteristics when designing language-specific models and highlights how language structure influences optimal n-gram selection.

Acknowledgment

The author would also like to acknowledge the support of the University of Buraimi to thank the university for the financial assistance, resources, facilities, and academic assistance largely used in the completion of this work.

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Al Katat S, Zaki C, Hazimeh H, Bitar I, Angarita R, and Trojman L (2024). Natural language processing for Arabic sentiment analysis: A systematic literature review. IEEE Transactions on Big Data, 10(5): 576-594. https://doi.org/10.1109/TBDATA.2024.3366083

Alhazmi A, Mahmud R, Idris N, Abo MEM, and Eke C (2024). A systematic literature review of hate speech identification on Arabic Twitter data: Research challenges and future directions. PeerJ Computer Science, 10: e1966. https://doi.org/10.7717/peerj-cs.1966 **PMid:38660217 PMCid:PMC11041964**

AlMahmoud RH and Hammo BH (2024). SEWAR: A corpus-based N-gram approach for extracting semantically-related words from Arabic medical corpus. Expert Systems with Applications, 238: 121767. https://doi.org/10.1016/j.eswa.2023.121767

Alomari D and Ahmad I (2024). Exploring character trigrams for robust Arabic text classification: A comparative analysis in the face of vocabulary expansion and misspelled words. IEEE Access, 12: 57103-57116. https://doi.org/10.1109/ACCESS.2024.3390048

Al-Shalif SA, Senan N, Saeed F, Ghaban W, Ibrahim N, Aamir M, and Sharif W (2024). A systematic literature review on meta-heuristic based feature selection techniques for text classification. PeerJ Computer Science, 10: e2084. https://doi.org/10.7717/peerj-cs.2084 **PMid:38983195 PMCid:PMC11232610**

Alshammary H, Ibrahim MF, and Hussein HA (2024). Evaluating the impact of feature extraction techniques on Arabic reviews classification. InfoTech Spectrum: Iraqi Journal of Data Science, 1(1): 42-54. https://doi.org/10.51173/ijds.v1i1.10

de la Hoz-Ruiz A, Howard E, and Hijón-Neira R (2024). The enhancement of statistical literacy: A cross-institutional study using data analysis and text mining to identify statistical issues in the transition to university education. Information, 15(9): 567. https://doi.org/10.3390/info15090567

Fazal F and Farook C (2024). A machine learning approach for depression detection in Sinhala-English code-mixed. International Journal on Advances in ICT for Emerging Regions, 17(3): 102-112. https://doi.org/10.4038/icter.v17i3.7282

Gurgurov D, Bäumel T, and Anikina T (2024). Multilingual large language models and curse of multilinguality. Arxiv Preprint Arxiv:2406.10602. https://doi.org/10.48550/arXiv.2406.10602

Jacques de Sousa L, Poças Martins J, Sanhudo L, and Santos Baptista J (2024). Automation of text document classification in the budgeting phase of the Construction process: A systematic literature review. Construction Innovation, 24(7): 292-318. https://doi.org/10.1108/CI-12-2022-0315

Khurana D, Koli A, Khatter K, and Singh S (2023). Natural language processing: State of the art, current trends and challenges. Multimedia Tools and Applications, 82(3): 3713-3744. https://doi.org/10.1007/s11042-022-13428-4 **PMid:35855771 PMCid:PMC9281254**

Kumar R and Thirumaran S (2024). Enhancing automatic English word analysis and prediction using higher-order n-gram models. In the International Conference on Science Technology Engineering and Management, IEEE, Coimbatore, India: 1-7. https://doi.org/10.1109/ICSTEM61137.2024.10560953

Masadeh M, Chola C, and Muaad AY (2024). Investigating the impact of preprocessing techniques and representation models on Arabic text classification using machine learning. International Journal of Advanced Computer Science and Applications, 15(1): 1115-1123. https://doi.org/10.14569/IJACSA.2024.01501110

Miah MSU, Kabir MM, Sarwar TB, Safran M, Alfarhood S, and Mridha MF (2024). A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and LLM. Scientific Reports, 14: 9603. https://doi.org/10.1038/s41598-024-60210-7 **PMid:38671064 PMCid:PMC11053029**

Moniri S, Schlosser T, and Kowerko D (2024). Investigating the challenges and opportunities in Persian language information retrieval through standardized data collections and deep learning. Computers, 13(8): 212. https://doi.org/10.3390/computers13080212

Ribeiro M, Malcorra B, Mota NB, Wilkens R, Villavicencio A, Hubner LC, and Rennó-Costa C (2024). A methodology for explainable large language models with integrated gradients and linguistic analysis in text classification. Arxiv Preprint Arxiv:2410.00250. https://doi.org/10.48550/arXiv.2410.00250

Sabty C (2024). Computational approaches to Arabic-English code-switching. Arxiv Preprint Arxiv:2410.13318. https://doi.org/10.48550/arXiv.2410.13318

Setiawan Y, Maulidevi NU, and Surendro K (2024). The optimization of n-gram feature extraction based on term occurrence for cyberbullying classification. Data Science Journal, 23: 31. https://doi.org/10.5334/dsj-2024-031

Shannaq B (2024a). Improving security in intelligent systems: How effective are machine learning models with TF-IDF vectorization for password-based user classification. Journal of Theoretical and Applied Information Technology, 102(22): 8340-8355.

Shannaq B (2024b). Unveiling the nexus: Exploring TAM components influencing professors' satisfaction with smartphone integration in lectures: A case study from Oman. Technology, Education, Management, Informatics Journal, 13(3): 2365-2375. https://doi.org/10.18421/TEM133-63

Shannaq B and Al-Zeidi A (2024). Intelligent information system: Leveraging AI and machine learning for university course registration and academic performance enhancement in educational systems. In: Hamdan A (Eds.), Achieving sustainable business through AI, technology education and computer science: Teaching technology and business sustainability: 51-65. Volume 2, Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-031-71213-5_5

Shannaq B, Al Shamsi I, and Majeed SNA (2019). Management information system for predicting quantity martials. Technology, Education, Management, Informatics Journal, 8(4): 1143-1149. https://doi.org/10.18421/TEM84-06

Shannaq B, Ali O, Maqbali SA, and Al-Zeidi A (2024). Advancing user classification models: A comparative analysis of machine learning approaches to enhance faculty password policies at the University of Buraimi. Journal of Infrastructure, Policy and Development, 8(13): 9311. https://doi.org/10.24294/jipd9311

Sindane T and Marivate V (2024). From n-grams to pre-trained multilingual models for language identification. Arxiv Preprint Arxiv:2410.08728. https://doi.org/10.48550/arXiv.2410.08728

Sun G, Cheng Y, Zhang Z, Tong X, and Chai T (2024). Text classification with improved word embedding and adaptive segmentation. Expert Systems with Applications, 238: 121852. https://doi.org/10.1016/j.eswa.2023.121852

Svete A, Borenstein N, Zhou M, Augenstein I, and Cotterell R (2024). Can transformers learn n-gram language models? In the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, USA: 9851–9867. https://doi.org/10.18653/v1/2024.emnlp-main.550

Taha K, Yoo PD, Yeun C, Homouz D, and Taha A (2024). A comprehensive survey of text classification techniques and their research applications: Observational and experimental insights. Computer Science Review, 54: 100664. https://doi.org/10.1016/j.cosrev.2024.100664

Wahdan A, Al-Emran M, and Shaalan K (2024). A systematic review of Arabic text classification: Areas, applications, and future directions. Soft Computing, 28(2): 1545-1566. https://doi.org/10.1007/s00500-023-08384-6

Yokoyama Y (2024). 4-gram applied to English Q&A statements to obtain factor scores. In the 16th IIAI International Congress on Advanced Applied Informatics (IIAI-AAI), IEEE, Takamatsu, Japan: 9-14. https://doi.org/10.1109/IIAI-AAI63651.2024.00011 **PMid:39015444 PMCid:PMC11247237**

Yusuf A, Sarlan A, Danyaro KU, Rahman ASB, and Abdullahi M (2024). Sentiment analysis in low-resource settings: A comprehensive review of approaches, languages, and data sources. IEEE Access, 12: 66883-66909. https://doi.org/10.1109/ACCESS.2024.3398635