From Roots to Tweets: Leveraging Linguistic Features for Arabic Social Media Content Classification

Hamda Slimi
Laboratory of Mathematics, Informatics and Systems, University of Echahid Cheikh Larbi Tebessi, Algeria
slimi.hamda@univ-tebessa.dz

Ibrahim Bounhas
Research Laboratory on Information Science, Higher Institute of Documentation, University of Manouba, Tunisia
ibrahim.bounhas@isd.uma.tn

Abstract: *The proliferation of disinformation and hate speech across Arabic-speaking social media requires more effective content classification. This paper introduces a novel framework that constructs morpho-syntactic knowledge graphs Arabic Knowledge Graph (ARKG) specifically designed to exploit Arabic's rich morphological structure for social media content classification. Our approach deconstructs tweets into their core linguistic components-words, lemmas, and roots-and uses Term Frequency-Inverse Document Frequency (TF-IDF) to weigh their relationships, creating rich, interpretable feature vectors that we evaluate using both traditional machine learning classifiers and graph neural network embeddings via metapath2vec. We evaluate the ARKG on three distinct tasks: fake news, hate speech, and sarcasm detection. Experimental results demonstrate that our linguistically-motivated features consistently outperform graph embeddings, achieving F1-scores of 95% for fake news, 84% for hate speech, and 82% for sarcasm detection-with our approach proving competitive with state-of-the-art transformer models while offering greater interpretability. Notably, our graph-based method outperforms a Masked Arabic Bidirectional Encoder Representations from Transformers (MARBERT) baseline by 5% in sarcasm detection while maintaining comparable performance on the other tasks. This work demonstrates that morphologically-aware graph representations can match transformer performance while providing interpretable insights into Arabic text classification.*

Keywords: *Arabic tweet classification, graph representation learning, machine learning, knowledge graphs, heteregenous graphs.*

Received March 27, 2025; accepted December 28, 2025
https://doi.org/10.34028/iajit/23/5/6

1. Introduction

In this paper, we consider the general task of Arabic text classification [12]. One of the primary data sources for this task is social media platforms like Twitter, Reddit, and Facebook. These platforms serve as breeding grounds for the dissemination of disinformation, hate speech, and other harmful content. Consequently, various datasets are collected from these platforms to investigate and mitigate the spread of detrimental content [2, 39, 49].

Arabic, known for its semantic nuances, poses challenges when utilizing computer tools for language understanding and interpretation [22]. In recent years, two main advancements in the field of deep learning revolutionized Natural Language Processing (NLP). On the one hand, it has witnessed the emergence of language models pretrained on the general task of next-token prediction using a transformer architecture [15]. These models have rapidly achieved State-Of-The-Art (SOTA) results across languages and NLP Tasks [59]. The performance is partially attributed to the large, high-quality datasets on which these models are trained and the neural network encoder-decoder/decoder-decoder architecture, which leverages the information present in these datasets. Despite their considerable performance, their lack of interpretability did not allow linguists to gain insight into the intricacies of language. This conventional method ensures a streamlined process and consistently achieves competitive results in Arabic text classification [48].

On the other hand, graphs have proven to be reliable and interpretable across a variety of NLP tasks [27]. In the context of Arabic NLP, we can observe a scarcity in this approach to detecting hateful content (cf. Section 2.3). Therefore, we aim to propose a back-to-basics approach that leverages the morpho-syntactic structures, the derivation and inflection processes of the Arabic language to infer information about the text's potential veracity, sarcasm or whether it contains hate speech. The proposed approach utilizes metapath2vec [27] to obtain graph-sensitive feature vectors, which are then employed to predict the class of a tweet. This study proposes a graph-based approach for NLP tasks, aimed at evaluating the effectiveness of an Arabic Knowledge Graph (ARKG) in conveying information for tweet classification. Initially, ARKG is constructed by linking tweets to words, lemmas, and roots, modeling the derivational and inflectional processes of the Arabic language. Efficient text mining techniques are then employed to compute the node features in ARKG. Subsequently, graph embedding is used to mine the entire graph and generate high-quality tweet vector representations, which serve as input for a classifier. The approach is evaluated across three specific tasks: hate speech detection, fake news detection, and sarcasm detection.

This paper is structured as follows. Section 2 presents an extensive review of related research. Section 3 explores the research proposals, providing a detailed explanation of the proposed approach and its constituent steps. In Section 4, we describe the experiments conducted to validate the relevance of the morphological graph. Finally, Section 5 concludes the paper and suggests potential avenues for future improvements.

2. Literature Review

Arabic tweet classification is of great importance since the large adoption of social media use across the Arab world [18]. The online scene has noticed the proliferation of online content creation, hence imposing responsibilities on authorities to keep at bay citizens from disinformation and hate speech. NLP tasks such as hate speech detection [49], sentiment analysis [4, 54, 56], sarcasm detection [2] and fake news/disinformation detection [38, 53] are mature fields for English and have assembled substantial attention from the research community, where researchers have achieved SOTA results across tasks using a wide range of approaches including text-focused methods, where authors use language models such as Bidirectional Encoder Representations from Transformers (BERT) [26], Robustly Optimized BERT Pretraining Approach (RoBERTa) [46], and Generalized Autoregressive Pretraining for Language Understanding (XLNet) [61] to detect disinformation solely from tweets' text [59]. Other approaches have turned to utilizing feature engineering and network structures to acquire information about tweet or user trustworthiness [24, 37, 57, 58].

Adapting these approaches to Arabic language requires handling its specificities at the morpho-syntactic level. That's why, we start by studying the derivational and inflection processes of this language and the NLP tools which may be used to handle Arabic morphology and word representation levels (cf. Sections 2.1 and 2.2). We also provide a summary of the related works in the field of Arabic tweet classification and heterogenous graphs embedding (cf. Sections 2.3 and 2.4).

2.1. Arabic Derivational and Inflectional Morphology

Arabic word construction follows a set of rules that allow for the study of their structure [21, 35]. Two classes of Arabic words can be distinguished: non-derived words, such as proper nouns, and derived words, which are built through derivation and inflection processes. Given the complexity and expressiveness of Arabic, the lexicon can be modeled at several levels of abstraction [22], as follows:

- **Roots**: The most abstract units, consisting of 2 to 5 letters (typically tri-literal), encapsulating core meanings. There are approximately 10,000 roots, from which 1,000 stems and dozens of thousands of words can be generated.
- **Verbal lemmas**: Derived from roots through 15 verbal patterns, these simple and augmented verbs can have similar or vastly different meanings, increasing the lexicon size and enhancing expressiveness.
- **Lemmas**: In addition to verbal lemmas, this set includes nouns and adjectives derived from verbs. A verbal lemma can generate hundreds of tokens expressing various meanings related to place, time, adverbs, etc. The lemma is the minimal lexical unit with a sense without inflection, representing the singular masculine form of nouns or the third singular masculine person for verbs [22].
- **Stems**: The inflected (or conjugated) forms of verbs and nouns, obtained by adding affixes (prefixes, infixes, and suffixes) to the lemma. Inflection allows for varying morpho-syntactic attributes such as tense, person, number, mood, gender, and voice for verbs, and number, case, gender, and definiteness for nouns and adjectives [22].
- **Words**: written forms also called surface words are obtained by agglutinating enclitics and/or proclitics to stems.

According to this process, an Arabic word is composed of five main components:

word = proclitic + prefix + base + suffix + enclitic

Figure 1 illustrates the process of Arabic word construction through an example.

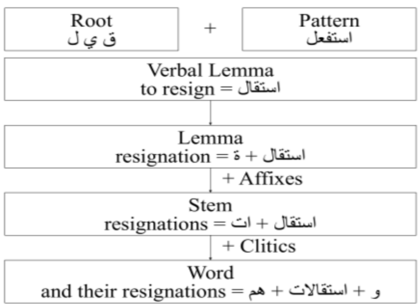


Figure 1. Example of arabic word construction.

2.2. Handling Arabic Morphology and Word Representation Levels

Several types of NLP tools have been developed to process Arabic texts and reduce Arabic words to their canonical forms and recognize their structure. Table 1 summarizes the main characteristics of the most used tools. Each tool allows for retrieving a sub-set of all the possible attributes of a given word. Furthermore, some tools render short diacritics, which are crucial for reducing the ambiguity of Arabic texts [22]. We also distinguish between deterministic and non-deterministic tools, with the latter being unsuitable for text classification without a disambiguation utility, as they return all possible solutions for a given word out of context, increasing ambiguity.

Table 1. A comparative study of Arabic NLP tools.

Tool	Morpho syntactic attributes	Root	VL	Lemma	Stem	Vowels
Khoja		✓				
Ghwanmeh		✓	✓		✓	
Light10					✓	
SAMA	✓			✓	✓	✓
Alkhalil 1.0	✓	✓	✓		✓	✓
Alkhalil 2.0	✓	✓	✓	✓	✓	✓
Stanford POS tagger	Only POS				✓	
MADAMIRA	✓			✓	✓	✓
FARASA	POS, gender and Number	✓	✓	✓	✓	
CAMeL	✓	✓		✓	✓	✓

While some morphological analyzers provide rich morphosyntactic information, cover all word representation levels, and produce voweled output (e.g., Alkhalil 2.0), light morphology-based tools may suffer from several limitations. For instance, root extraction and light stemming may fail to extract the root or stem correctly or omit important parts of the basic form [36]. Light stemming partially handles inflectional morphology by removing a small set of prefixes and suffixes, leading to term mismatch due to the inability to handle flectional irregularities, such as broken plurals.

In contrast, root extractors aim to handle both derivational and inflectional morphology, but they can introduce ambiguity and decrease precision [45]. Stemmers often produce poor results for text mining applications, as using roots to represent concepts is not appropriate [23]. More sophisticated tools, such as Morphological Analysis and Disambiguation for Arabic (MADAMIRA) [52] and its successor Computational Approaches to Modeling Language (CAMeL) Tools [51], are trained on tagged corpora and provide richer information. These tools have also been adapted to process dialects. Compared to A Fast and Accurate Arabic Segmenter and Morphological Analyser (FARASA) and the Stanford Part-Of-Speech (POS) tagger, CAMeL returns more comprehensive information about words and produces voweled output. However, these tools can still produce an erroneous morphological analysis for a word, leading to term mismatch. In summary, MADAMIRA and CAMeL are the only deterministic tools that produce voweled output, which is crucial for feature extraction, as non-voweled lemmas increase ambiguity due to the lack of diacritics.

The choice of the best "indexing unit" to represent Arabic words effectively remains a challenge, despite the various basic forms outputted by the previously studied Arabic NLP tools at different representation levels [20, 32, 33, 34]. Several light stemmers and root extractors have been evaluated for document classification tasks [36], with Ghwanmeh's tool retrieving the best results. In contrast, Khoja's tool outperformed light stemming and n-gram-based approaches. Other studies have also highlighted the contribution of root-based text classification [3, 7, 8, 36].

Roots and stems are two widely adopted alternatives for word representation [7, 8], often achieving better results compared to surface words in several studies [11, 44]. However, root-based representation has been shown to suffer from low precision while increasing recall [11]. Most research findings indicate that stem-based indexing significantly outperforms root-based approaches [19, 29]. Recent results suggest that lemma-based indexing can enhance results and serve as an intermediate solution that balances precision and recall compared to roots and stems [22]. For instance, the experiments conducted by [28] on standard Text REtrieval Conference (TREC) Arabic collections confirm that the choice of indexing unit remains central to Arabic information retrieval performance.

Based on the aforementioned works, it can be concluded that the use of taggers providing rich morpho-syntactic information, particularly lemma-based indexing, remains an area that warrants further investigation [22]. While progress has been made, identifying the optimal word representation approach for Arabic NLP tasks remains an ongoing challenge.

2.3. Arabic Tweet Classification

Over the past decade, social media has gained significant momentum and played a pivotal role in civil movements in Tunisia and Egypt. With Saudi Arabia ranking 9th globally in terms of the number of Twitter users, there has been a substantial surge in online content in Modern Standard Arabic (MSA) and various Arabic dialects. This trend has captured the attention of researchers, who sought to explore and interpret the large corpus of user-generated Arabic content [18].

Arabic text classification is a research field that aims to classify documents into pre-defined classes using Machine Learning (ML) and Deep Learning (DL) techniques based on textual content. Twitter, a platform primarily centered around short-form text where users share opinions and knowledge on various topics, has given rise to a sub-field called Arabic tweet classification [13]. Its main purpose is to leverage social media features to understand societal characteristics of the Arabic community. Additionally, the absence of gatekeepers in social media has allowed the dissemination of harmful content, necessitating the development of methods to classify tweets into predefined classes, leading to the emergence of numerous downstream tasks.

Within the scope of the presented approach, we specifically focus on three distinct tasks: hate speech detection [6, 10, 60], fake news detection [16, 37, 58], and sarcasm detection [5, 30]. It is worth noting the scarcity of up-to-date labeled datasets available in the Arabic language, in contrast to the abundance of datasets in the English language [38, 53].

In hate speech detection, the objective is to identify posts that include derogatory behaviors or threats aimed at a particular group using ML and DL approaches. Derogatory behaviors encompass expressions or incitements of disdain and insult, while threats may manifest as violent actions, incitements of isolation, or expressions of hatred. This task is the most vocabulary-sensitive among the three, as derogatory terms used in hate speech are often reiterated [14, 17, 42, 49].

For fake news detection, the goal is to recognize check-worthy information and validate its factuality. Fake news is defined by the Cambridge dictionary as "false stories that appear to be news, spread on the internet or using other media, usually created to influence political views or as a joke". This domain explores ML and DL techniques to recognize fake news by leveraging text, graph, and image features [25, 38, 47, 53], while Arabic-specific studies have explored fact-checking and textual-analysis features [40, 41].

In sarcasm detection, the implications are less substantial compared to the previous tasks with inherent political dimensions. Nonetheless, it presents a distinctive challenge due to the absence of social features that help recognize sarcasm. Additionally, conventional vocabulary can often be employed to convey sarcasm, rendering it challenging to differentiate from literal statements. As a result, the principal difficulties in sarcasm detection arise from its nuanced meaning, which can pose challenges for both automated systems and human interpretation [5, 30].

2.4. Heterogenous Graph Neural Networks

Social graphs are an essential part of social media, they link users, posts and entire communities and allow for information dissemination. In the context of tweet classification, user social network and tweet propagation networks were heavily used in previous works. They achieved competitive results compared with approaches that were limited to text content. Furthermore, knowledge graphs which represent networks with node and/or edge features were widely adopted due to the rich information they can represent. However, extracting meaningful knowledge from these graphs was a cumbersome task, until the introduction of Graph Neural Networks (GNNs).

GNNs are a class of deep learning models designed for processing and extracting information from graphs. They are characterized by the exchange of vector messages between nodes in a graph, which are updated using neural networks. The core concept in GNNs is neural message passing, where each node's embedding is updated based on information aggregated from its neighborhood in the graph. For graph learning tasks, there are usually three kinds of tasks which include node-level (classification, clustering), edge-level (classification, prediction), and graph-level (classification, matching) [55].

The structure of graphs can convey valuable information for node classification tasks. Indeed, Node2vec [31] has been extensively utilized to leverage node embeddings in node classification tasks. However, it is limited to homogeneous graphs. On the other hand, Twitter social interactions require representation through heterogeneous graphs, which can depict post-post interactions, user-post interactions, and user-user interactions accurately. This need for a graph representation learning approach tailored to heterogeneous graphs played a significant role in driving the development of Metapath2vec [27].

Metapath2vec is a network embedding algorithm suited for heterogeneous information networks. It generates vector representations for nodes whilst considering various node and relationship types. Metapath2vec extends the concept of node embeddings, similar to Node2Vec, to heterogeneous networks by integrating meta-paths. They are comprised of sequences of node and edge types, which define structured paths within the network. Furthermore, they guide random walks in the network, enabling the algorithm to capture network heterogeneity and produce embeddings that align with the network's inherent structure [27].

In the context of the research tasks addressed within this study, and to the best of our knowledge metapath2vec was only employed for the purpose of fake news detection in the English language [53]. In contrast, a multitude of GNN-based approaches have been applied to the domain of English fake news detection. Where they have been instrumental in identifying and characterizing fake news communities [25], leveraging message propagation networks to recognize fake news [47], and incorporating knowledge graphs within a multi-modal framework to improve fake news detection [38]. However, the domains of hate speech and sarcasm detection have not yet witnessed a substantial exploration of graph-based techniques, whether in the context of the English or Arabic languages.

3. Research Proposals

Harmful content is almost always conveyed in textual form, which explains the outstanding performance of Large Language Model (LLM)-based approaches [48]. However, there exists untapped potential in approaching text-related problems from a unique perspective. In the context of Arabic language, these graphs can greatly aid in deciphering the intricacies of the language [22]. Morpho-syntactic graphs offer a structural representation of the relationships between words, their lemmas, and roots. Our proposed approach acknowledges the promise of graph representation learning while also recognizing the crucial role of textual analysis in dealing with harmful content. Consequently, we propose an approach that constructs Morpho-syntactic graphs for tweets, each node of which is equipped with a feature vector designed to convey task-specific information.

We propose a hybrid approach which combines linguistic information, word, and document embeddings to classify Arabic texts. Given an input dataset composed of set of texts, we build an Arabic knowledge graph which models four distinct nodes, namely tweets, normalized words, lemmas, and roots. Indeed, previous works in Arabic NLP and Information Retrieval (IR) showed that combining different word representation levels (e.g. words, lemmas and roots) significantly enhances results [20, 32, 33, 34].

What's more, we are inspired by the work of Bounhas *et al*. [22] which proposed to model the Arabic lexicon with a multi-level knowledge graph, which enhanced query expansion-based IR. To the best of our knowledge, this approach has not been exploited for Arabic text classification. Furthermore, existing works did not employ recent advances in the field of GNN to mine the built graph. Finally, authors in [22] combined Ghwanmeh and MADAMIRA tools to obtain both the lemmas and the roots of the words, which negatively influenced the results. In our case, we exploit CAMel which provides both information and allows to build a morpho-syntactic graph containing derivational and flectional relations. Details about CAMel and the structure of the given graph are given in Section 3.1. Then, we exploit a pipeline of statistical and graph mining methods which generate efficient word and document embeddings (cf. Section 3.2).

3.1. Arabic Morphological Analysis and Disambiguation

Our approach integrates CAMeL [51], which performs tokenization and morphological disambiguation, thus identifying the most suitable morpho-syntactic tags and word segmentation based on the context. It is the only Python open-source tool of its kind which returns both the lemmas and the roots of the words. Besides, the POS, it retrieves all the morpho-syntactic features of words like gender and number for nouns. Our choice is also justified by the fact that CAMeL allows to handle Arabic dialects, namely Egyptian, Levantine, and Gulf, as datasets extracted from social networks may contain dialectal texts.

Table 2 displays the results of processing two sample sentences.

Table 2. Processing results of two sample sentences.

	Token	Translation	Normalized word	Lemma	Root	POS	Other attributes
Sentence 1	أعلن	(He) announced	أعلن	أعلن	علن	Verb	Person: 3 Voice: active Number: singular Gender: masculine
	المديرون	The + directors	المديرون	مدير	در#	Noun	Number: plural Gender: masculine
	استقالتهم	Their + Resignations	استقالتهم	استقالة	ق#ل	Noun	Number: M Gender: feminine
	مباشرة	Immediately	مباشرة	مباشرة	بشر	Noun	Number: singular Gender: feminine
Sentence 2	مباشرة	Immediately	مباشرة	مباشرة	بشر	Noun	Number: singular Gender: feminine
	قرر	(He) + Decided	قرر	قرر	قرر	Verb	Person: 3 Voice: active Number: singular Gender: masculine
	الوزير	The minister	الوزير	وزير	زر#	Noun	Number: singular Gender: masculine
	إعلان	Announcement	إعلان	إعلان		Noun	Number: singular Gender: masculine
	استقالته	His + Resignations	استقالته	استقالة	ق#ل	Noun	Number: singular Gender: feminine

Figure 2 illustrates the graph generated from the same data. For each token, we provide three-word representations returned by CAMeL (namely the normalized word, the lemma, and the root) and the morpho-syntactic attributes including the POS. We notice that the output tokens are filtered by POS to remove empty words. In the following, the nodes whose type is "word" represent the normalized words returned by CAMeL and not surface words which appear in the original texts.

Our goal is to utilize subcomponents of the Arabic language to reduce its complexity by mapping tweet content to different levels of granularity. By projecting tweets into a list of lemma nodes that are also connected to other tweets, we can gain insights into the nuances of the words from which the lemmas were derived [20]. This graph structure implicitly links tweets at various analytical levels. We propose that each type of these implicit links captures aspects of the semantic relationships between tweets. Efficient graph embedding algorithms can exploit these links to uncover hidden semantic features that are useful for classification.

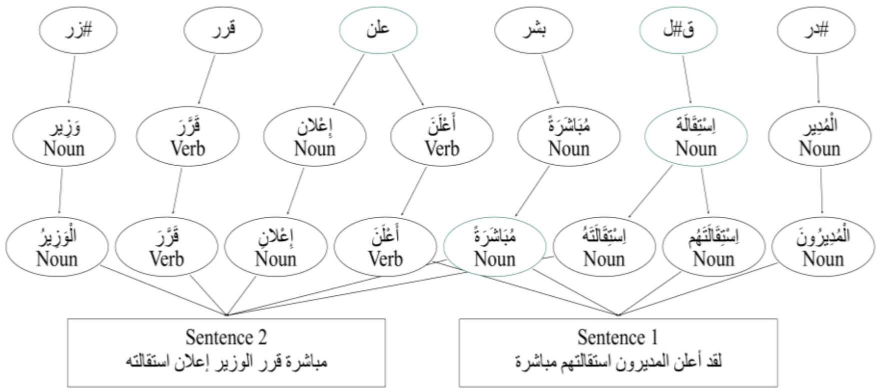


Figure 2. Example of Arabic knowledge graph extracted from two sample sentences. Nodes outlined in green are the components shared by both sentences.

3.2. Graph Mining and Node Embedding

The whole process for building ARKG and exploiting it for tweet classification is illustrated by Figure 3. Graph construction consists of two main stages. During the initial stage, the nodes lack feature vectors, and only the edges carry weights. These weights are determined based on the prevalence of a word node, lemma node, or root node in relation to the tweet node. These weights are computed with TF-IDF, which allows to evaluate the discriminative power of terms over the set of tweets and classes.

To transition from the initial graph to the final graph, a series of processing steps are applied. First, for each word, lemma, and root, we create an embedding vector. This vector is formed by concatenating the weights associated with the target node across all tweets in the dataset. For example, if a root node, denoted as ri, is linked to five different tweets, each tweet-root has its respective edge weight. We create an embedding vector with a length equal to the number of tweets in the dataset. Initially, all values are set to 0, and only the indices corresponding to the tweets to which ri is connected are assigned the TF-IDF weight value.

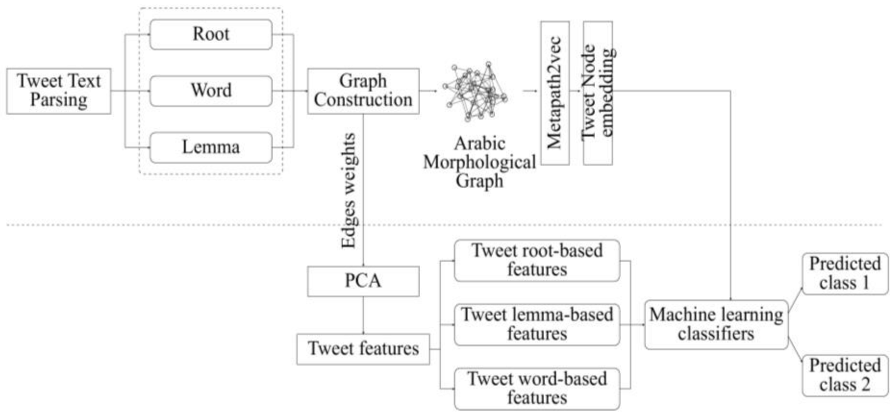


Figure 3. Arabic knowledge graph construction and tweet classification pipeline.

The resulting vector size can become substantial, depending on the number of tweets in the dataset. Consider a dataset with 1000 tweets; the word vector would have a dimension of [1000, 1], where only a small number of tweets containing the word would have a non-zero TF-IDF weight. Such a sparse vector results in large vector sizes with limited valuable information. To mitigate this issue, we stack the embedding vectors obtained for all words (respectively lemmas and roots) and apply Principal Component Analysis (PCA) [43] to reduce the embedding vector dimensionality to a predefined size of 500. This dimension was determined empirically as a compromise between computational constraints and performance. While hardware limitations in our setup prevented the use of significantly larger values, experiments with lower dimensions (e.g., 125, 250) yielded a clear drop in performance. The value of 500 was thus selected to best condense the sparse TF-IDF vectors while preserving their discriminative power.

Notably, the tweet node remains the only node without a feature vector. In our proposed approach, the primary objective is to predict the class of tweets within different contexts. Thus, we generate a feature vector for the tweet by averaging the embedding vectors of each node type (word, root, and lemma). As in our experiments, each tweet can be expressed by the subset of its words, roots, or lemmas; e.g.,

$$Tweet1 = R250, R5, R35, R60, R2 \tag{1}$$

$$Tweet1 = W350, W280, W1, W2, W1, W36 \tag{2}$$

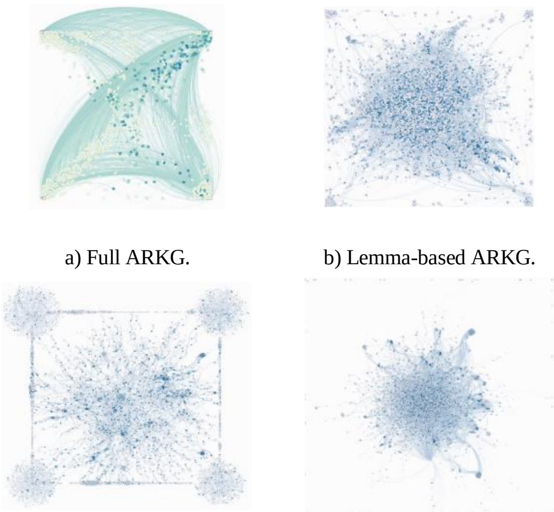
This process results in three distinct feature vectors for each tweet, each corresponding to its lemma, root, and word representations.

The heterogeneous morphological graph, denoted as G, comprises four distinct sets of nodes: tweet nodes (N_t), word nodes (N_o), lemma nodes (N_l), and root nodes (N_r). Each tweet node, represented as $t \in N_t$, is characterized by a weight vector $t = [w_1, w_2, ..., w_j, ..., w_{|v|}]$. Here, $|V|$ signifies the dimensionality, determined by the number of components in PCA, and each w_i element has values from $\{-1, 1\}$. These weight vectors convey the relevance of the respective nodes to the tweet.

In each dataset, we feature four distinct ARKG variants: a full ARKG version, while the other options are constrained to specific node types, such as roots, lemmas, and words.

In the depicted figures (4 to 6), nodes are assigned colors based on their in-degree, with the darkest nodes representing tweets that exhibit the highest number of sub-components. The initial graph in each task illustrates the complete ARKG along with all its constituent elements. However, the subsequent three graphs portray sub-graphs related to lemma-tweet, word-tweet, and root-tweet edges.

In figure 4 we observe the presence of four discernible clusters, which aligns with the typical nature of fake news datasets, often organized around specific topics. These clusters likely correspond to sub-topics within the dataset.

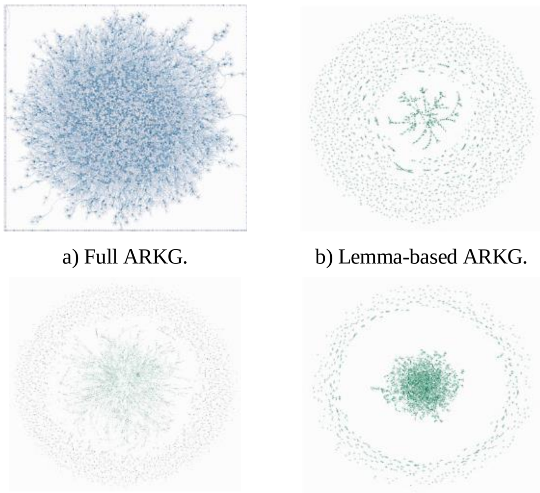


a) Full ARKG. b) Lemma-based ARKG.

c) Word-based ARKG. d) Root-based ARKG.

Figure 4. ARKG for fake news detection.

In figure 5, the graphs related to hate speech exhibit greater density compared to the other tasks. Moreover, the sub-graphs feature a central cluster which is an indicative of highly related tweets that share common roots, words, or lemmas. In the vicinity of this central cluster, we can identify outlier tweets that contain words unrelated to the main cluster. It is important to note that this pattern is not exclusive to the hate speech graph and is also observable in the sarcasm graph.

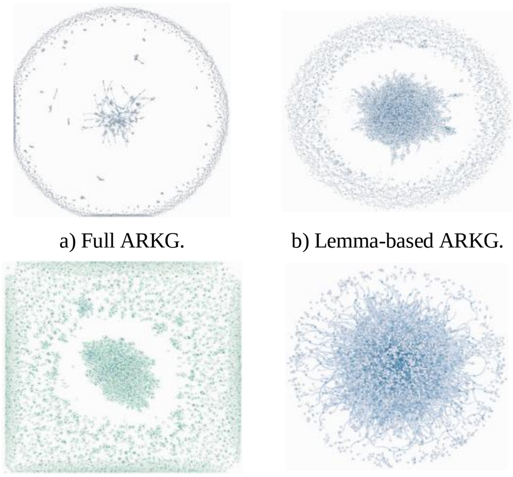


a) Full ARKG. b) Lemma-based ARKG.

c) Word-based ARKG. d) Root-based ARKG.

Figure 5. ARKG for hate speech detection.

Indeed, figure 6 reveals a smaller and less dense central cluster within the sarcasm graph, which reflects the absence of interwoven subjects. This finding is logical given the nature of sarcasm, which is typically less topic-oriented and relies heavily on the manipulation of words in various contexts. In conclusion, the visualization of ARKG provides valuable insights into the subjects of tweets and their relationships.



a) Full ARKG. b) Lemma-based ARKG.

c) Word-based ARKG. d) Root-based ARKG.

Figure 6. ARKG for sarcasm detection.

The graphs are fed to Metapath2vec to obtain a vector representation of tweet nodes based on their roots, lemmas and words components. The algorithm relies on metapaths to perform random walks across the heterogeneous information network i.e., the full ARKG. Random walks start at a specific node and navigate through the graph by following the meta-path constraints. During these walks, nodes are visited in a sequence, creating a corpus of node sequences. Metapath2vec employs a skip-gram model, a variant of

Word2Vec, to learn embeddings. In the skip-gram model, the goal is to predict the surrounding nodes (context) of a given node (target) in the generated node sequences.

The meta-path is a series of steps through the graph edges where the end of each step is the starting point for the next one. Before feeding an ARKG to Metapath2vec, it is transformed to and undirected graph to allow coherent steps across it. Metapath2vec has multiple hyperparameters, which include but are not limited to embedding vector size, walk length and walks per node. For the purposes of our study, the most effective configuration was found to be an embedding dimension of 16-32, a walk length of 10, and 8 walks per node. The choice of a 16-32 embedding dimension aligns with standard practices for capturing graph-structural information. Our empirical tests confirmed that this range is sufficient for encoding the relevant topological features of the ARKG, as using larger vectors provided no additional performance benefits in our classification tasks.

In summary, the proposed approach capitalizes on two key aspects of the graph. First, it leverages the TF-IDF edge weights, which are converted into three sets of tweet features (root-based, lemma-based, and word-based). Second, it employs metapath2vec to generate node embeddings that encapsulate the structural characteristics of the graph. These two distinct feature sets offer specific insights into the classification of tweet nodes.

3.3. Methodological Considerations for Applying Metapath2vec to ARKG

A crucial preliminary adjustment was the transformation of the ARKG into an undirected graph. The initial, natural representation of the graph is directed (e.g., Tweet→Lemma). However, for Metapath2vec to discover relationships between tweets, the random walker must be able to traverse from one tweet to a shared component (like a lemma) and then proceed to another tweet connected to that same component. Making the edges bidirectional (e.g., Tweet-Lemma, Lemma-Tweet) enables these essential Tweet→Component→Tweet walks, which form the basis of learning tweet-level similarities from shared linguistic features.

The core of Metapath2vec lies in guiding random walks with pre-defined Meta-path schemas. For the ARKG, this involves strategically selecting the sequence of node types for the walks. Our approach of using the "full ARKG" allows the random walks to flexibly traverse between words, lemmas, and roots. This implicitly allows the algorithm to find diverse paths. A further refinement, constituting a critical area for future work, would be to generate embeddings from distinct meta-paths (e.g., T-W-T, T-L-T, T-R-T) separately and then combine them. This would allow the model to learn from different semantic granularities simultaneously, potentially improving performance on complex tasks.

The optimal hyperparameters for the random walks are heavily influenced by the unique topology of the ARKG. We determined that a relatively long walk length (walk length=10) was necessary. This adjustment allows the walks to navigate from a starting tweet, through its various sub-components, and successfully reach other tweets in the network, thereby capturing non-local structural information. Shorter walks risk terminating before exploring inter-tweet relationships. Similarly, the number of walks per node must be sufficient to adequately sample the connections in a graph that can become dense around common lemmas and roots, which is a characteristic feature of morphologically rich languages like Arabic.

4. Experiments and Results

In the following sub-sections, we present the details of our experiments focusing on the datasets, the experimental setup and the results, before discussing the main findings.

4.1. Dataset Description

Our experimental evaluation employs three distinct datasets each tailored to a specific NLP task (cf. table 3). First, the Arabic COVID-19 Twitter dataset (ArCOV19) [39] comprises two subsets: one dedicated to claim verification and the other for tweet verification (Fake news/not fake news). In our experiments, we only use the second subset. Second, The Levantine Hate Speech and Abusive Language (L-HSAB) dataset [49] is centered around political tweets some of which are expressed in the Levantine dialect. These tweets are categorized into three classes: Hate, Normal, and Abusive. In the context of our contribution, we combine Abusive and Hate tweets into a single class termed "Hate speech". Lastly, Arsarcasm [2] uses a dataset originally intended for sentiment analysis and labels it for sarcasm detection. This dataset combines Semantic Evaluation (SemEval) and Arabic Sentiment Tweets Dataset (ASTD) [50] encompassing various dialects such as Egyptian, Gulf, Levantine, Maghrebi, and MSA. It is the largest dataset in terms of size, but exhibits significant class imbalance, with only 16% of the tweets labeled as sarcastic.

Table 3. Datasets description.

Dataset	Class	Count	Total count
Arcov19 [39]	Verified	1831 (51.1%)	03584
	Fake news	1753 (48.9%)	
L-HSAB [49]	Normal	3650 (62.4%)	05846
	Hate speech	2196 (37.6%)	
Arsarcasm [2]	Non-sarcastic	8865 (84%)	10547
	Sarcastic	1682 (16%)	

4.2. Experimental Setup

This section delineates the experimental setup, encompassing two key aspects: the baseline approaches for comparison and our model selection and evaluation methodology. These components provide the framework for assessing the efficacy of our morphosyntactic graph approach across multiple NLP tasks.

4.2.1. Baseline Description

The proposed approach demonstrates the practical applications of morpho-syntactic graphs in NLP tasks. Given that pre-trained language models such as Arabic Bidirectional Encoder Representations from Transformers (AraBERT) and Qatar Computing Research Institute Arabic and Dialectal BERT (QARiB) have shown strong performance across various tasks, we adopt these as our primary baselines. All baseline models were evaluated on identical datasets to ensure fair comparison.

In fake news detection, we benchmark against the comprehensive study by Al-Yahya *et al*. [9], which evaluates multiple language models on the ArCOV19-Rumors dataset. Their work compares deep learning architectures, namely Convolutional Neural Networks (CNN) and Gated Recurrent Units (GRU), with transformer-based models, namely AraBERT, Arabic Efficiently Learning an Encoder that Classifies Token Replacements Accurately (AraELECTRA), and QARiB. The best-performing model, QARiB, achieved a state-of-the-art F1-score of 95.6% using five training epochs with the Adam optimizer.

For hate speech detection, we approach this as a binary classification task following the configuration established by the dataset authors, Mulki *et al*. [49]. They merged the 'Hate' and 'Abusive' classes from their L-HSAB dataset of Levantine dialect tweets. Using traditional machine learning with n-gram features, their Naive Bayes classifier achieved the highest F1-score of 89.0%.

In sarcasm detection, our comparison baseline comes from Abdul-Mageed *et al*. [1], who fine-tuned Multi-dialectal MARBERT-a large transformer model pre-trained on one billion Arabic tweets-on the ArSarcasm dataset. Their model, trained for 25 epochs with a learning rate of 2e-6, achieved a state-of-the-art F1-score of 76.30%.

4.2.2. Model Selection and Evaluation

Our model leverages two distinct feature sets; edge weights derived through TF-IDF, and node embeddings extracted using Metapath2vec. These feature sets are employed independently and in conjunction to evaluate their respective impacts on each task. The combination of feature sets, whether for graph node types (word, root, lemma) or graph and text features, is achieved through a straightforward concatenation of the corresponding embedding vectors.

The approach employs ML models using the Lazypredict library in Python. It evaluates a large number of classifiers through four metrics, namely accuracy, balanced accuracy, F1-score and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). Then, we select the best performing classifier for further hyperparameters tuning to achieve optimal results.

To address the class imbalance in the Arsarcasm dataset (16% sarcastic tweets), we implemented cost-sensitive learning through the scale_pos_weight parameter, which was set to the ratio of negative (non-sarcastic) to positive (sarcastic) samples. This standard technique increases the cost of misclassifying the minority class and serves a similar purpose to data-level techniques like oversampling or undersampling by preventing the model from becoming biased towards the majority class.

eXtreme Gradient Boosting (XGBoost) consistently outperformed other models in the majority of experiments. It achieved competitive results with the following hyperparameters: binary logistic objective function, logarithmic loss as the evaluation metric, 500 estimators, a learning rate of 0.1, a maximum depth of 8, a minimum child weight of 1, a subsample ratio of 0.8, a column sample ratio of 0.8 for each tree, a gamma of 0, a lambda of 1, and an alpha of 0. The thorough evaluation of the best-performing feature set is based on four metrics: Precision, Recall, F1-score, and accuracy. The model is evaluated using a 5-fold cross validation approach to ensure the reliability of the results.

4.3. Experimental Results

First, we showcase the performance of each text-based feature set for each task. Then, we select the best performing set and compare it to the performance of the graph-embedding based approach, and their combination (graph embedding + text features).

4.3.1. Text Features Results

The experimental results reveal distinct patterns across the three tasks, providing insights into the complementary nature of different linguistic representations. In hate speech detection (Table 4), we observe that combining all feature types yields the best performance (84% F1-score), representing a 3-point improvement over the best individual feature set (lemma-based). This suggests that the different linguistic levels capture complementary discriminative patterns for hate speech identification. The metrics values are all equal due to the balanced class distribution (approximately 50/50) as shown in Table 3.

Table 4. Experimental results of text features for hate speech detection.

Class/Metric	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Lemma	82	82	81	82
Root	78	79	78	79
Word	81	81	80	81
Combination	84	84	84	84

Notably, root-based features consistently underperform across all tasks, which can be attributed to two main factors:

1) The dialectal nature of the Levantine Arabic dataset, which deteriorates parser performance.
2) The high level of abstraction in root representations that may obscure task-specific discriminative features by linking tweets with fewer actual semantic similarities.

For fake news detection (Table 5), word-based features achieve optimal performance (95% across all metrics), while the combination approach (93%) does not provide additional benefits. This indicates that surface-level lexical patterns are sufficient for this task, and the additional abstraction from lemmas and roots introduces noise rather than complementary information. The marginal 2-point decrease when combining features suggests that fake news detection relies primarily on specific lexical cues that are best preserved at the word level.

Table 5. Experimental results of text features for fake news detection.

Class/Metric	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Lemma	93	93	93	93
Root	92	91	91	91
Word	95	95	95	95
Combination	93	93	93	93

In sarcasm detection (Table 6), lemma-based features demonstrate superior precision (84%), while the combination approach matches lemma performance in precision and recall but slightly underperforms in F1-score. This pattern suggests that morphological normalization provided by lemmatization effectively captures the linguistic patterns indicative of sarcasm, while additional feature types do not contribute meaningfully to the detection task.

Table 6. Experimental results of text features for sarcasm detection.

Class/Metric	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Lemma	84	86	82	86
Root	82	85	81	85
Word	82	85	80	85
Combination	83	86	82	86

The influence of feature combination varies significantly across tasks, revealing task-specific linguistic requirements. In hate speech detection, the combination leverages complementary strengths: lemma-based features provide morphological normalization, word-based features preserve contextual specificity, and even the underperforming root features contribute marginal discriminative information that collectively enhances performance. Conversely, for fake news and sarcasm detection, the combination approach either matches or slightly underperforms the best individual feature set, suggesting that these tasks benefit from focused linguistic representations rather than comprehensive feature aggregation. This task-dependent behavior indicates that the optimal feature strategy should be tailored to the specific linguistic phenomena underlying each classification problem.

4.3.2. Graph Features Results

In this section, we evaluate graph embedding-based features and their combination with text features (cf. tables 7, 8 and 9 for the three tasks). The purpose of using graph embedding is to demonstrate the graph's capacity to encapsulate information related to NLP-oriented tasks, which often depend on direct text features like word and sentence embeddings. Based on the results, graph embeddings consistently demonstrate competitive performance across various tasks. Notably, the best performance is attained in sarcasm detection. However, the sarcasm dataset exhibits a significant class imbalance, resulting in a skewed accuracy metric. Investigating the performance on this dataset using F1-score shows the graph embedding model only achieves 77%.

Table 7. Experimental results of graph features for hate speech detection.

Features Class/Metric	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Graph	70	67	68	72
Text features	84	84	84	84
Combination	81	82	81	82

Table 8. Experimental results of graph features for fake news detection.

Features Class/Metric	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Graph	82	82	82	82
Text features	95	95	95	95
Combination	90	90	90	90

Table 9. Experimental results of graph features for sarcasm detection.

Features Class/Metric	Precision (%)	Recall (%)	F1-score (%)	Accuracy (%)
Graph	74	83	77	84
Text features	84	86	82	86
Combination	84	85	79	85

To ascertain the proposed approach performance, we compare it with a set of baselines detailed in Section 4.2.1. In every task, the best performing feature sets based on the previous results is selected. The proposed approach outperforms established models, AraBERT and AraGPT2 in fake news detection and demonstrates performance comparable to QARiB. The latter, a model pre-trained extensively on Arabic-language tweets, is known for its high performance in NLP tasks involving Twitter data. Hence, matching its performance is a substantial feat. In hate speech detection, our results are competitive but do not surpass the baseline. However, in sarcasm detection, we excel, surpassing the MARBERT-based approach by 5.7 percentage points in F1-score (cf. Table 10).

Table 10. Results comparison against baselines.

Task		Precision (%) (% Chg. baseline)	Recall (%) (% Chg. baseline)	F1-score (%) (% Chg. baseline)	Accuracy (%) (% Chg. baseline)
Hate speech detection	Baseline	90.5	89.0	89.0	90.3
	Proposed approach	84.0 (-6.5)	84 (-5.62)	84 (-5.62)	84.0 (-6.3)
Fake news detection	AraBERT	84.6	81.6	81.2	82.5
	AraGPT2	91.4	92.7	90.8	97.5
	QARiB (baseline)	95.6	95.6	95.3	97.5
	Proposed approach	95.0 (-0.6)	95.0 (-0.6)	95.0 (-0.3)	95.0 (-2.5)
Sarcasm detection	MARBERT (baseline)	-	-	76.3	-
	Proposed approach	84.0	86.0	82.0 (5.7)	86.0

5. Discussion

The proposed approach integrates two fundamental components to address Arabic text classification challenges. The ARKG serves as the foundation, systematically decomposing tweets into morpho-syntactic elements-words, lemmas, and roots-to create a heterogeneous graph structure that connects each tweet to its linguistic constituents. Complementing this structure, our weighting strategy quantifies the relevance of relationships between tweets and their components, ultimately producing a weighted heterogeneous graph that encodes distributional information across the dataset. Methodological Challenges and Limitations.

The ARKG's effectiveness is inherently tied to the accuracy of the CAMeL morphological analyzer. Processing the diverse and informal Arabic dialects prevalent in social media introduces what we term "morphological noise"-systematic errors in tokenization, lemmatization, and root extraction that propagate through the graph structure. These inaccuracies manifest as erroneous connections between tweets through incorrectly identified shared components, potentially misleading the random walk algorithms and causing the model to learn spurious semantic relationships. Given Arabic's morphological complexity, characterized by extensive word form variations and dialectal diversity, this challenge represents a fundamental limitation of our approach.

The multi-level abstraction of the ARKG-spanning normalized words, lemmas, and roots-creates a complex design space for meta-path selection. Different linguistic levels capture distinct semantic relationships, requiring task-specific optimization. Deep etymological connections accessed through Tweet → Root → Tweet paths prove effective for topic-centric tasks such as fake news detection, where broad thematic relationships are crucial. Conversely, tasks requiring sensitivity to specific lexical choices, such as sarcasm detection with its reliance on contextual wordplay, benefit from surface-level paths like Tweet → Word → Tweet. This heterogeneity necessitates careful meta-path engineering tailored to each application domain.

Experimental Findings and Performance Analysis

The experimental datasets exhibited distinct task-specific characteristics that illuminate the versatility of our approach. Hate speech detection requires recognition of specific discriminatory lexical patterns, while fake news detection demonstrates high topic sensitivity—performance typically degrades when datasets span diverse topics, motivating the development of domain-specific corpora (e.g., COVID-19, political news). Sarcasm detection presents unique challenges due to its dependence on contextual cues and severe class imbalance, yet our method maintained consistent effectiveness. Table 10 reports the overall comparison of the proposed approach against the baselines for the three tasks.

The sustained performance across varied tasks demonstrates the robustness of the ARKG framework. Unlike conventional language models that employ sub-word tokenization-potentially fragmenting meaningful linguistic units-our approach preserves semantic interpretability through human-readable graph components. This transparency facilitates both model interpretability and error analysis, while the multi-level representation captures complementary aspects of Arabic morphology that prove beneficial across diverse NLP applications.

The success of our weighting strategy suggests that distributional information encoded in component-tweet relationships provides a valuable signal that generalizes across tasks, supporting the broader applicability of graph-based approaches for morphologically rich languages. However, future work must address the dependency on morphological analysis accuracy and develop more robust methods for handling dialectal variation in social media contexts.

6. Conclusions and Future Works

The morpho-syntactic complexity of Arabic presents significant NLP challenges, particularly for social media texts. We address this by proposing the ARKG, which transforms tweets into multi-level semantic representations encompassing words, lemmas, and roots, offering an interpretable alternative to opaque sub-word tokenization.

Our hybrid methodology extracts text-centric features via TF-IDF weighted edges and captures structural topology through metapath2vec embeddings. We demonstrate that optimal linguistic representation is task-dependent: surface-level words excel for fake news detection, while normalized lemmas perform better for sarcasm identification. Our approach achieved state-of-the-art performance in Arabic sarcasm detection, surpassing MARBERT baselines while remaining competitive in fake news detection.

The ARKG framework provides both predictive power and interpretability, serving as a diagnostic instrument for task-specific linguistic phenomena. Future work will enrich the ARKG with syntactic dependencies and develop task-specific meta-path strategies.

References

[1] Abdul-Mageed M., Elmadany A., and Nagoudi E., "ARBERT and MARBERT: Deep Bidirectional Transformers for Arabic," *in Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Online, pp. 7088-7105, 2021. DOI:10.18653/v1/2021.acl-long.551

[2] Abu Farha I. and Magdy W., "From Arabic Sentiment Analysis to Sarcasm Detection: The Ar-Sarcasm Dataset," *in Proceedings of the 4th Workshop on Open-Source Arabic Corpora and Processing Tools, with a Shared Task on Offensive Language Detection*, Marseille, pp. 32-39, 2020. https://aclanthology.org/2020.osact-1.5/

[3] Al-Arfaj A. and Al-Salman A., "Arabic NLP Tools for Ontology Construction from Arabic Text: An Overview," *in Proceedings of the International Conference on Electrical and Information Technologies*, Marrakech, pp. 246-251, 2015. https://doi.org/10.1109/eitech.2015.7162980

[4] Al-Ayyoub M., Khamaiseh A., Jararweh Y., and Al-Kabi M., "A Comprehensive Survey of Arabic Sentiment Analysis," *Information Processing and Management*, vol. 56, no. 2, pp. 320-342, 2019. DOI:10.1016/j.ipm.2018.07.006

[5] Al-Ghadhban D., Alnkhilan E., Tatwany L., and Alrazgan M., "Arabic Sarcasm Detection in Twitter," *in Proceedings of the International Conference on Engineering and MIS*, Monastir, pp. 1-7, 2017. https://doi.org/10.1109/icemis.2017.8272990

[6] Al-Hassan A. and Al-Dossari H., "Detection of Hate Speech in Social Networks: A Survey on Multilingual Corpus," *In Proceedings of The 6th International Conference on Computer Science and Information Technology*, Sydney, pp. 1-18, 2019. https://doi.org/10.5121/csit.2019.90208

[7] Al-Kabi M., Al-Radaideh Q., and Akkawi K., "Benchmarking and Assessing the Performance of Arabic Stemmers," *Journal of Information Science*, vol. 37, no. 2, pp. 111-119, 2011. https://doi.org/10.1177/0165551510392305

[8] Al-Shawakfa E., Al-Badarneh A., Shatnawi S., Al-Rabab'ah K., and Bani-Ismail B., "A Comparison Study of Some Arabic Root Finding Algorithms," *Journal of the American society for information science and technology*, vol. 61, no. 5, pp. 1015-1024, 2010. https://doi.org/10.1002/asi.21301

[9] Al-Yahya M., Al-Khalifa H., Al-Baity H., AlSaeed D., and Essam A., "Arabic Fake News Detection: Comparative Study of Neural Networks and Transformer-Based Approaches," *Complexity*, vol. 2021, pp. 1-10, 2021. https://doi.org/10.1155/2021/5516945

[10] Albadi N., Kurdi M., and Mishra S., "Are They Our Brothers? Analysis and Detection of Religious Hate Speech in The Arabic Twittersphere," *in Proceedings of The IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Barcelona, pp. 69-76, 2018. DOI:10.1109/asonam.2018.8508247

[11] Aljlayl M. and Frieder O., "On Arabic Search: Improving the Retrieval Effectiveness Via a Light Stemming Approach," *In Proceedings of The 11th International Conference on Information and Knowledge Management*, McLean, pp. 340-347, 2002. https://doi.org/10.1145/584792.584848

[12] Alruily M., "Classification of Arabic Tweets: A Review," *Electronics*, vol. 10, no. 10, pp. 1-31, 2021. DOI:10.3390/electronics10101143

[13] Alzanin S., Azmi A., and Aboalsamh H., "Short Text Classification for Arabic Social Media Tweets," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 9, pp. 6595-6604, 2022. https://doi.org/10.1016/j.jksuci.2022.03.020

[14] Anezi F., "Arabic Hate Speech Detection Using Deep Recurrent Neural Networks," *Applied Sciences*, vol. 12, no. 12, pp. 1-15, 2022. https://doi.org/10.3390/app12126010

[15] Antoun W., Baly F., and Hajj H., "AraELECTRA: Pre-Training Text Discriminators for Arabic Language Understanding," *in Proceedings of the 6th Arabic Natural Language Processing Workshop*, Kyiv, pp. 191-195, 2020. https://aclanthology.org/2021.wanlp-1.20/

[16] Aoun Barakat K., Dabbous A., and Tarhini A., "An Empirical Approach to Understanding Users' Fake News Identification On Social Media," *Online Information Review*, vol. 45, no. 6, pp. 1080-1096, 2021. https://doi.org/10.1108/oir-08-2020-0333

[17] Aref A., Al Mahmoud R., Taha K., Al-Sharif M., and et al, "Hate Speech Detection of Arabic Shorttext"; *in Proceedings of The 9th International Conference on Information Technology Convergence and Services*, Copenhagen, pp. 81-94, 2020. https://doi.org/10.5121/csit.2020.100507

[18] Askool S., "The Use of Social Media in Arab Countries: A Case of Saudi Arabia," *in Proceedings of the Web Information Systems and Technologies: 8th International Conference*, Porto,

pp. 201-219, 2012. 2013. https://doi.org/10.1007/978-3-642-36608-6_13

[19] Atwan J., Mohd M., Rashaideh H., and Kanaan G., "Semantically Enhanced Pseudo Relevance Feedback for Arabic Information Retrieval," *Journal of Information Science*, vol. 42, no. 2, pp. 246-260, 2016. https://doi.org/10.1177/0165551515594722

[20] Ben Guirat S., Bounhas I., Slimani Y., "Meta-Search Based Approach for Arabic Information Retrieval," *Online Information Review*, vol. 46, no. 7, pp. 1257-1274, 2022. https://doi.org/10.1108/oir-11-2020-0515

[21] Boudchiche M., Mazroui A., Bebah M., Lakhouaja A., and Boudlal A., "AlKhalil Morpho Sys 2: A Robust Arabic Morpho-Syntactic Analyzer," *Journal of King Saud University-Computer and Information Sciences*, vol. 29, no. 2, pp. 141-146, 2017. https://doi.org/10.1016/j.jksuci.2016.05.002

[22] Bounhas I., Soudani N., and Slimani Y., "Building A Morpho-Semantic Knowledge Graph for Arabic Information Retrieval," *Information Processing and Management*, vol. 57, no. 6, pp. 102124, 2020. https://doi.org/10.1016/j.ipm.2019.102124

[23] Bounhas I., "On the Usage of a Classical Arabic Corpus as a Language Resource: Related Research and Key Challenges," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 18, no. 3, pp. 1-45, 2019. https://doi.org/10.1145/3277591

[24] Castillo C., Mendoza M., and Poblete B., "Information Credibility on Twitter," *in Proceedings of the 20th International Conference on World Wide Web*, Hyderabad, pp. 675-684, 2011. https://doi.org/10.1145/1963405.1963500

[25] Chandra S., Mishra P., Yannakoudakis H., Nimishakavi M., and et al, "Graphbased Modeling of Online Communities for Fake News Detection," *arXiv Pre-prints*, vol. arXiv:2008.06274v4, pp. 1-16, 2020. https://doi.org/10.48550/arXiv.2008.06274

[26] Devlin J., Chang M., Lee K., and Toutanova K., "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding," *in Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Minneapolis, pp. 4171-4186, 2019. https://doi.org/10.18653/v1/n19-1423

[27] Dong Y., Chawla N., and Swami A., "Metapath2vec: Scalable Representation Learning for Heterogeneous Networks," *in Proceedings of the 23rd ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, Halifax, 2017, pp. 135-144, 2017. https://doi.org/10.1145/3097983.3098036

[28] El Mahdaouy A., Gaussier E., and Ouatik El Alaoui S., "Should One Use Term Proximity or Multi-Word Terms for Arabic Information Retrieval?," Computer Speech and Language, vol. 58, pp. 76-97, 2019. https://doi.org/10.1016/j.csl.2019.04.002

[29] El Mahdaouy A., El Alaoui S., and Gaussier E., "Word-Embedding-Based Pseudo-Relevance Feedback for Arabic Information Retrieval," *Journal of information science*, vol. 45, no. 4, pp. 429-442, 2019. https://doi.org/10.1177/0165551518792210

[30] Faraj D. and Abdullah M., "SarcasmDet at Sarcasm Detection Task 2021 in Arabic Using AraBERT Pretrained Model," *in Proceedings of the 6th Arabic Natural Language Processing Workshop*, Kyiv, pp. 345-350, 2021. https://aclanthology.org/2021.wanlp-1.44/

[31] Grover A. and Leskovec J., "Node2vec: Scalable Feature Learning for Networks," *in Proceedings of The 22nd ACM SIGKDD International Conference On Knowledge Discovery and Data Mining*, San Francisco, pp. 855-864, 2016. https://doi.org/10.1145/2939672.2939754

[32] Guirat S., Bounhas I., and Slimani Y., "Combining Indexing Units for Arabic Information Retrieval," *International Journal of Software Innovation*, vol. 4, no. 4, pp. 1-14, 2016. https://doi.org/10.4018/ijsi.2016100101

[33] Guirat S., Bounhas I., and Slimani Y., "Enhancing Hybrid Indexing for Arabic Information Retrieval," *in Proceedings of the 32nd International Symposium*, Poznan, pp. 247-254, 2018. https://doi.org/10.1007/978-3-030-00840-6_27

[34] Guirat S., Bounhas I., and Slimani, Y., "Pre-Indexing Techniques in Arabic Information Retrieval," *in Proceedings of the 11th International Conference on Agents and Artificial Intelligence*, Prague, pp. 237-246, 2019. https://doi.org/10.5220/0007393402370246

[35] Habash N., *Introduction to Arabic Natural Language Processing*, Springer Nature, 2022. https://doi.org/10.1007/978-3-031-02139-8

[36] Hadni M., Lachkar A., and Ouatik S., "A New and Efficient Stemming Technique for Arabic Text Categorization," *in Proceedings of the International Conference on Multimedia Computing and Systems*, Tangier, pp. 791-796, 2012. https://doi.org/10.1109/icmcs.2012.6320308

[37] Hamdi T., Slimi H., Bounhas I., and Slimani Y., "A Hybrid Approach for Fake News Detection in Twitter Based on User Features and Graph Embedding," *in Proceedings of the 16th International Conference on Distributed Computing and Internet Technology*, Bhubaneswar, pp. 266-280, 2020. https://doi.org/10.1007/978-3-030-36987-3_17

[38] Han Y., Silva A., Luo L., Karunasekera S., and

Leckie C., "Knowledge Enhanced Multi-Modal Fake News Detection," *arXiv Pre-prints*, vol. arXiv:2108.04418v1, pp. 1-10, 2021. https://doi.org/10.48550/arXiv.2108.04418

[39] Haouari F., Hasanain M., Suwaileh R., and Elsayed T., "ArCOV19-Rumors: An Arabic COVID-19 Twitter Dataset for Misinformation Detection," *in: Proceedings of the 6th Arabic Natural Language Processing Workshop*, Kyiv, pp. 72-81, 2021. https://aclanthology.org/2021.wanlp-1.8/

[40] Harrag F. and Djahli M., "Arabic Fake News Detection: A Fact Checking Based Deep Learning Approach," *Transactions on Asian and Low-Resource Language Information Processing*, vol. 21, no. 4, pp. 1-34, 2022. https://doi.org/10.1145/3501401

[41] Himdi H., Weir G., Assiri F., and Al-Barhamtoshy H., "Arabic Fake News Detection Based on Textual Analysis," *Arabian Journal for Science and Engineering*, vol. 47, no. 8, pp. 10453-10469, 2022. https://doi.org/10.1007/s13369-021-06449-y

[42] Himdi H., Alhayan F., and Shaalan K., "Neural Networks and Sentiment Features for Extremist Content Detection in Arabic Social Media," *The International Arab Journal of Information Technology*, vol. 22, no. 3, pp. 522-534, 2025. https://doi.org/10.34028/iajit/22/3/8

[43] Jollife I. and Cadima J., "Principal Component Analysis: A Review and Recent Developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, pp. 1-16, 2016. https://doi.org/10.1098/rsta.2015.0202

[44] Larkey L. and Connell M., "Arabic Information Retrieval at UMass in TREC-10," *in Proceedings of the Tenth Text REtrieval Conference*, Gaithersburg, pp. 1-9, 2001. https://doi.org/10.21236/ada456273

[45] Larkey L., Ballesteros L., and Connell M., *Arabic Computational Morphology: Knowledge-Based and Empirical Methods*, Springer, pp. 221-243, 2007. https://doi.org/10.1007/978-1-4020-6046-5_12

[46] Liu Y., Ott M., Goyal N., Du J., and et al, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," *arXiv preprint*, vol. arXiv:1907.11692, pp. 1-13, 2019. https://doi.org/10.48550/arXiv.1907.11692

[47] Matsumoto H., Yoshida S., and Muneyasu M., "Propagation-Based Fake News Detection Using Graph Neural Networks with Transformer," *in Proceedings of The IEEE 10th Global Conference on Consumer Electronics*, Kyoto, pp. 19-20, 2021. https://doi.org/10.1109/gcce53005.2021.9621803

[48] Mohammed A. and Kora R., "An Effective Ensemble Deep Learning Framework for Text Classification"; *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 10, pp. 8825-8837, 2022. https://doi.org/10.1016/j.jksuci.2021.11.001

[49] Mulki H., Haddad H., Bechikh Ali C., and Alshabani H., "L-HSAB: A Levantine Twitter Dataset for Hate Speech and Abusive Language," *in Proceedings of the 3rd Workshop on Abusive Language Online*, Florence, pp. 111-118, 2019. https://doi.org/10.18653/v1/w19-3512

[50] Nabil M., Aly M., and Atiya A., "ASTD: Arabic Sentiment Tweets Dataset," *in Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Lisbon, pp. 2515-2519, 2015. https://doi.org/10.18653/v1/d15-1299

[51] Obeid O., Zalmout N., Khalifa S., Taji D., and et al, "CAMeL Tools: An Open Source Python Toolkit for Arabic Natural Language Processing," *in Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, pp. 7022-7032, 2020. https://aclanthology.org/2020.lrec-1.868/

[52] Pasha A., Al-Badrashiny M., Diab M., El Kholy A., and et al, "Madamira: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic," *in Proceedings of the 9th International Conference on Language Resources and Evaluation*, Reykjavik, pp. 1094-1101, 2014. http://www.lrec-conf.org/proceedings/lrec2014/pdf/593_Paper.pdf

[53] Ren Y. and Zhang J., "Fake News Detection on News-Oriented Heterogeneous Information Networks Through Hierarchical Graph Attention," *in Proceedings of the International Joint Conference on Neural Networks*, Shenzhen, pp. 1-8, 2021. DOI:10.1109/ijcnn52387.2021.9534362

[54] Rosenthal S., Farra, N., and Nakov P., "SemEval-2017 Task 4: Sentiment Analysis in Twitter," *in Proceedings of the 11th International Workshop on Semantic Evaluation*, Vancouver, pp. 502-518, 2017. https://doi.org/10.18653/v1/S17-2088

[55] Scarselli F., Gori M., Tsoi A., Hagenbuchner M., and Monfardini G., "The Graph Neural Network Model," *IEEE Transactions On Neural Networks*, vol. 20, no. 1, pp. 61-80, 2008. https://doi.org/10.1109/tnn.2008.2005605

[56] Shoukry A. and Rafea A., "Sentence-Level Arabic Sentiment Analysis," *in Proceedings of the International Conference on Collaboration Technologies and Systems*, Denver, pp. 546-550, 2012. DOI:10.1109/cts.2012.6261103

[57] Slimi H., Bounhas I., and Slimani Y., "Twitter Users Crediblity Evaluation Based on Social Graph Impression," *in Proceedings of the 16th International Conference on Applied Computing*, Cagliari, pp. 1-8, 2019. https://doi.org/10.33965/ac2019_201912l002

[58] Slimi H., Bounhas I., Slimani Y., "URL-Based Tweet Credibility Evaluation," *in Proceedings of*

the IEEE/ACS 16th International Conference on Computer Systems and Applications, Abu Dhabi, pp. 1-6, 2019. https://doi.org/10.1109/aiccsa47632.2019.9035255

[59] Slimi H., Bounhas I., and Slimani Y., "Adapting Pre-Trained Language Models to Rumor Detection on Twitter," *Journal of Universal Computer Science*, vol. 27, no. 10, pp. 1128-1148, 2021. https://doi.org/10.3897/jucs.65918

[60] Yamaguchi S., "Why Are There So Many Extreme Opinions Online?: An Empirical, Comparative Analysis of Japan, Korea and the USA," *Online information review*, vol. 47, no. 1, pp. 1-19, 2023. https://doi.org/10.1108/oir-07-2020-0310

[61] Yang Z., Dai Z., Yang Y., Carbonell J., and et al, "XLNet: Generalized Autoregressive Pretraining for Language Understanding," *in Proceedings of the 33rd International Conference on Neural Information Processing Systems*, NewYork, pp. 1-18, 2019. https://doi.org/10.48550/arXiv.1906.08237

Hamda Slimi is an Assistant Professor at the University of Echahid Cheikh Larbi Tebessi, Algeria, and a member of the Laboratory of Mathematics, Informatics and Systems (LAMIS). Slimi's research focuses on the detection of fake news and hate speech on Twitter, in both English and Arabic, and on the use of large language models to assess the viability of synthetic data. Broader research interests Span Natural Language Processing, with particular emphasis on these two application areas.

Ibrahim Bounhas is an Associate Professor at the University of Manouba, Tunisia, and a member of the SILAB Research Laboratory on Information Science. Bounhas' research lies at the intersection of natural language processing, data mining and information retrieval, applying machine learning and deep learning to textual and tabular data. Recent work exploits Large Language Models and Deep Neural Networks for tasks such as fake news and hate speech detection on social networks.