AUT Journal of Modeling and Simulation

AUT J. Model. Simul., 58(1) (2026) 23-32
DOI: 10.22060/miscj.2025.24744.5431

Neuro-Symbolic Claim Detection for Arabic: Integrating Graph Neural Networks with Efficiently Fine-Tuned Small Language Models

Masood Dina Falah[1], Heshaam Faili[2] *

[1] Department of Computer Engineering, College of Alborz, University of Tehran, Tehran, Iran.
[2] School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran, Iran.

ABSTRACT: The rise in online information has increased the spread of misinformation and disinformation, which makes claim detection important for the reliability of digital content. Claim detection refers to the problem of determining whether a factual statement is real or fake. While there has been great advancement in high-resource languages, Arabic is particularly challenging due to its special characteristics and the low availability of annotated resources. In this paper, we make use of the Arabic News Stance (ANS) corpus and develop a multi-stage framework to help tackle these problems. Our method effectively models the structural interplay between claims and words with a carefully constructed heterogeneous bipartite graph and the use of advanced Graph Neural Networks (GNNs). In parallel, we used two Small-scale Language Models (SLMs) in this domain, namely LLaMA-3.2-1B and Qwen2.5-3B, and demonstrated that together with QLoRA-based fine-tuning, these lightweight models can be as effective as or even better than heavier baselines. To this point, we suggested a novel neuro-symbolic fusion paradigm that unifies structural reasoning with linguistic prior information. By integrating the outputs from a Graph Convolutional Network (GCN) with the fine-tuned Qwen2.5-3B logits, we achieved state-of-the-art performance on the ANS dataset with an accuracy of 0.732 and an F1-score of 0.717. This framework shows that the combined results of structural graph reasoning and SLMs lead to accurate and computationally efficient claim detection. This work not only enhances the detection of Arabic misinformation but also provides a scalable solution for low-resource languages.

Review History:

Received: Aug. 16, 2025
Revised: Nov. 12, 2025
Accepted: Dev. 07, 2025
Available Online: Jul. 20, 2026

Keywords:

Claim detection
Arabic
Graph Neural Networks
Small-scale Language Models
Supervised fine-tuning

1- Introduction

A claim is a statement of fact declaring something to be the case, capable of verification through external sources such as statistics, examples, or events reported [1]. The explosion of information available online, in particular across social media and online news outlets, has led to a serious issue with spreading false or misleading claims. Automatically detecting if a claim is true or false is a crucial part in combating misinformation or disinformation, which can affect public opinion, political stability, or trust in society [2]. While there has been a great deal of work on the detection of claims in the English language and other high-resource languages, these methods fail when applied to Arabic due to its unique circumstances [3]. The Arabic language has its own inherent difficulties due to widely different linguistic properties and variations in regions. Moreover, the lack of readily available high-quality annotated datasets for Arabic claim detection causes problems for the production and training of reliable large-scale models. Existing methods are therefore incapable of effective generalization in the low-resource situation [4, 5].

Previous work on Arabic claim detection has mainly relied on transformer-based structures to try to improve feature extraction [6]. There has been some increase in performance in these hybrid systems, but they are still limited in the scope of their abilities. They rely on either slow-to-compute large-scale models or complete performance degradation due to the lack of explicit use of structural relationships in claims between claims and words. To address these limitations, we propose a methodology consisting of several stages, making use of graph-based learning using Small-scale Language Models (SLMs) and a neuro-symbolic fusion design. Each stage of the pipeline is intended to progressively improve the final performance through structural reasoning and semantic priors. The main contributions of the work can be summarized as follows:

Graph-based Representation Learning: We design novel graph construction methods in which the claims and words are nodes in a heterogeneous bipartite graph. Used with Graph Neural Networks (GNNs), the structural dependencies are better captured, allowing for greater feature extraction than previous transformer techniques.

Small Language Models: We show how compact Small Language Models (SLMs), specifically LLaMA-3.2-1B and

*Corresponding author's email: hfaili@ut.ac.ir

Copyrights for this article are retained by the author(s) with publishing rights granted to Amirkabir University Press. The content of this article is subject to the terms and conditions of the Creative Commons Attribution 4.0 International (CC-BY-NC 4.0) License. For more information, please visit https://www.creativecommons.org/licenses/by-nc/4.0/legalcode.

Qwen2.5-3B, can achieve performance comparable to Large Language Models (LLMs), while also using a considerably smaller set of computational resources. Via zero-shot evaluation and parameter-efficient fine-tuning through Quantized Low-Rank Adaptation (QLoRA), we show how SLMs can serve as both practical and scalable replacements for LLMs in low-resource settings.

Neuro-Symbolic Fusion: We offer a creative fusion of graph-based learning and SLMs, inventing a hybrid neo-symbolic model that utilizes the relational power of GCNs and combines it with the semantic impact of fine-tuned language models. This provides a strong mathematical foundation but also enables us to think of feature extraction from two-dimensional symbolic relations and semantic contexts.

The remainder of the paper is organized as follows. In Section 2, we outline related studies in the area of claim detection. In Section 3, the proposed method is introduced in detail. In Section 4, the experimental results are presented and discussed, comparing with baseline methods. Finally, Section 5 concludes the paper and gives some recommendations for future work.

2- Related Works

In this section, we survey the works in which GNNs and LLMs for claim detection were used. Majer and Šnajder [7] investigated an automated means of claim detection and claim check-worthiness detection as a means of fact-checking using LLMs with zero- and few-shot prompting. The datasets spanned many domains, including politics and health, with annotations of each claim made for factuality and for check-worthiness. The model considered differences in verbosity in the prompting, varying from simple to complex, and also the context given to the prompt, including co-text and metadata. The results of the study showed that LLMs performed better with a moderate amount of verbosity, with contextualization improving performance, especially for the simpler prompts.

Hüsünbeyi and Scheffler [8] tackled the problem of claim detection at a sentence level due to limited and common data by using the ClaimBuster and NewsClaims data sets. They expanded on statistical approaches using domain knowledge and advanced neural architectures, building an ontology from the ClaimsKG knowledge graph and including it with BERT embeddings. This ontology-based BERT model improved on the standard model.

Chen et al. [9] addressed biomedical claim detection to counteract misinformation in social media. They observed that the traditional supervised methods or fine-tuned pre-trained Language Models such as BERT and RoBERTa had drawbacks, because the pre-training tasks did not correspond with the classification tasks. To overcome this limitation, they suggested using a prompt learning method, which transformed claim detection into a masked language modeling problem using templates using masked tokens. They tested their method on the BioClaim data set, which consisted of 1,200 annotated Twitter posts, showing that prompt more methods outperformed the traditional baselines.

Lee et al. [10] took part in SemEval-2023 Task 8, concerned with the detection of medical causal claims and PIO extraction on Reddit posts. Utilising in their work chiefly transformer-based models of DeBERTa and DeBERTaV3. They treated claims detection as concerning the status of claims classification at the sentence level, and PIO extraction was regarded as a sequence labelling exercise, making use of weak supervision. Strong results were attained with their system, comparing the results of BERT, BioBERT, and RoBERTa.

Wührl and Klinger [11] dealt with respect to biomedical fact-checking models using the CoVERT dataset. In this instance, they put forward the idea of condensing tweets into structured claims in terms of entity-relation-entity triples, or in minimal token sequences. Their evaluation with MultiVerS demonstrated significant improvements in fact-checking accuracy.

Wührl and Klinger [12] developed a first data set for biomedical claim detection on Twitter that addressed misinformation in health discussions. They annotated and collected 1,200 Tweets related to COVID-19, Measles, Cystic Fibrosis, and depression, concluding that approximately 45% of them contained claims that were mostly explicit ones. Among the classifier types tested, logistic regression achieved the highest performance (F1 ≈ 0.70).

3- Methodology

For this purpose, we used the Arabic News Stance (ANS) Corpus. After a dedicated preprocessing pipeline for Arabic text normalization and cleaning, we built a claim-word graph, which constitutes one of the methodological novelties of this work (it better models structural as well as semantic relationships in the Arabic claims). Additionally, as the representation, we used a graph-based neural architecture (with a GCN backbone) for claim classification. Additionally, several SLMs were investigated and enhanced (employing zero-shot prompting + parameter-efficient fine-tuning with QLoRA) to better fit them into the Arabic claim detection task. The major contribution of this work is the inspiration for a neuro-symbolic integration framework, which integrates graph-based structure learning and SLM-based semantic reasoning into a unified model. As depicted in Figure 1, the proposed approach follows the pipeline of dataset pre-training, graph generation, GNN training, and SLM fine-tuning before neuro-symbolic integration. The specifications of each part are shown in the next subsections.

3- 1- Dataset

In this study, we utilized the ANS Corpus [13], which is derived from the Arabic News Texts (ANT) corpus. This dataset is collected from various reputable news sources, including BBC Arabic and CNN Arabic. To achieve a balanced distribution of the claims, the articles were paraphrased and modified crowdsourcely, so that they produced both factual and fabricated claims. This corpus can thus be used as a relevant benchmark for the fact-detection task in Arabic, where the aim is to distinguish between true and false statements. The detailed statistics of the dataset can be seen

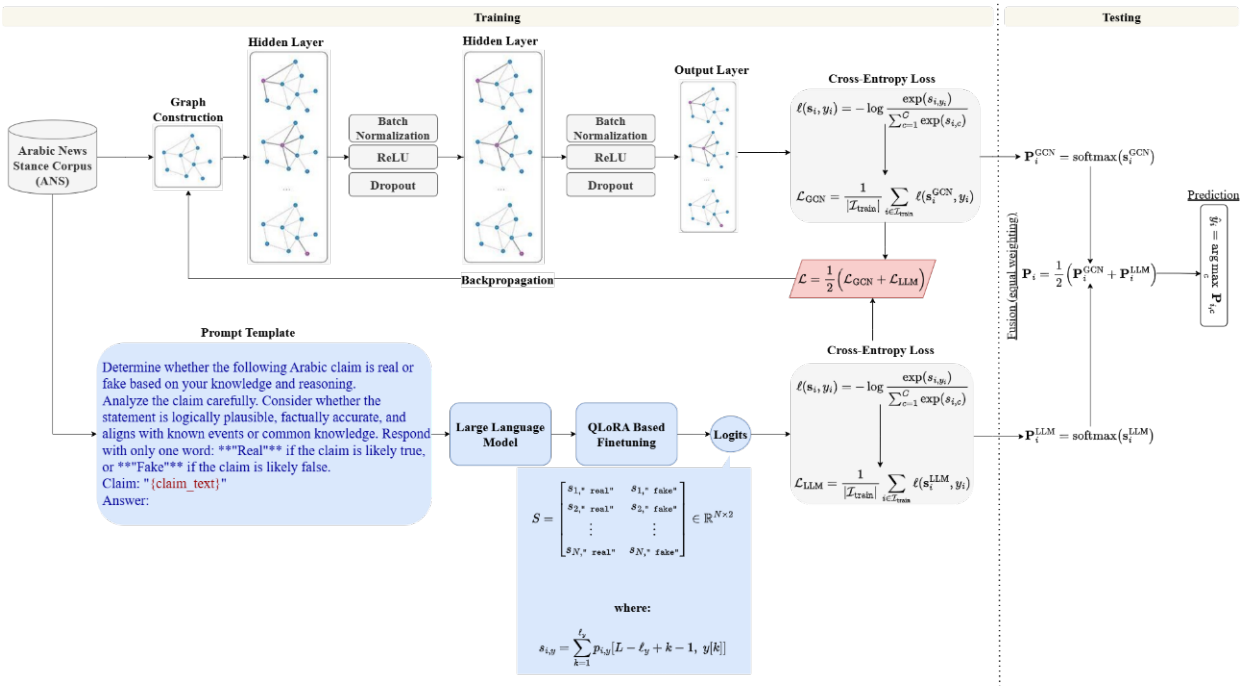

Fig. 1. The pipeline of our proposed methodology.

Table 1. Dataset statistics.

Split	False (Fake)	True (Real)	Total
Train	1035	2150	3185
Development	290	616	906
Test	306	150	456
Total	1631	2916	4547

in Table 1, which shows the distribution among various splits with respect to the claims. The dataset contains 4547 claims, out of which 1631 are labelled as False and 2916 as True, with the maximum number of samples in the training set at 3185, followed by the development set, which has 906 samples, followed by the test set, which has 456 samples.

3- 2- Date Preprocessing

To prepare the Arabic claims suitable for downstream analysis, we have established a dedicated preprocessing pipeline that is designed with the linguistic and orthographic characteristics of Modern Standard Arabic (MSA) in mind. This step is of vital importance because of the morphological richness and script variability of Arabic, to ensure the quality and consistency of the textual input for subsequent graph construction and model training. The preprocessing pipeline yielded high-quality token representations of Arabic claims in both normalized and semantically relevant terms. These normalized tokens then acted as the basis for the construction of the claim-word graph and for further classification purposes. The preprocessing consisted of the following steps:

Text Normalization: The writing of Arabic makes use of several shapes of writing, which are all the diacritical signs and multiple shapes of writing practices of certain characters. To normalize these last forms, we must first erase Tashkeel (the diacritical signs) and Tatweel (elongation signs). Different forms of the Alef character (e.g., “إ” ,“أ” ,“آ”) are thus passed in a normalized form (e.g., to a word with the written form “ا”), and the fixed Yeh (last form of Yeh “ى”) is converted into the normal written form “ي”. The fact of operating

these manipulations during the preprocessing allows for the production of tokens that are semantically equivalent to those regularly obtained in writing in the corpus.

Text Cleaning: All the non-alphanumeric characters of the text, including punctuation marks, signs, etc., are removed, and the white spaces existing in the text are also normalized. These last manipulations tend to diminish the noise existing in the corpus, thus permitting us to ensure the soundness of the tokenization for the next stages.

Tokenization: The preprocessed text was tokenized using the PyArabic library [14], an open-source Python library for Arabic language processing. PyArabic takes into consideration the common word boundaries and morphological features of Arabic, which produce linguistically coherent tokens.

Stop word Removal: A collection of Arabic stopwords was used to remove non-informative words, such as stop words or function words like conjunctions, particles, and articles. Tokens beyond the useful vocabulary of words those obscure, most-carry pieces of data that can help discard them.

Claim-wise Token Generation: Each claim was treated individually, and the tokenized value of it was saved into a dictionary indexed by claim identifiers. Simultaneously, a global vocabulary of unique tokens over all claims was built. The resulting vocabulary was then employed for graph construction and node embedding.

3- 3- Graph Construction

In order to describe the relational structure of the claims to their constituent words, we first construct a heterogeneous bipartite graph, where the words and claims are node types and their relational encoding are represented by edges that encode the degree of semantic relationship between individual words and claims. This graph representation allows GNNs to exploit the structural relationships as well as the semantic representations of the words jointly, thereby improving modelling for claim classification.

3- 3- 1- Node Construction

The set of claims is denoted as $X = \{X_1, X_2, \ldots, X_{N_c}\}$, and the set of unique words in the corpus is denoted as $W = \{W_1, W_2, \ldots, W_{N_w}\}$. Thus, the graph has $N = N_X + N_w$ nodes. The claim nodes and the word nodes are assigned different indices to avoid collisions.

3- 3- 2- Feature Initialization

All the nodes are enriched with semantic embeddings. For a claim node X_i, we get its representation using a pre-trained Arabic embedding model (inspired by AraBERT). Likewise, for all the word nodes W_j, we get their embeddings in the same embedding space. The embedding function $E(\cdot)$ maps claims and words into dense vectors as in formula (1):

$$h_{x_i} = E(x_i) \quad , h_{w_j} = E(w_j) \tag{1}$$

3- 3- 3- Edge Construction with PMI

There are edges between the claim nodes and their corresponding word nodes. An edge is created between each pair (x_i, w_j) if w_j appears in x_i. The edge is weighted according to Pointwise Mutual Information (PMI), which measures the strength of association between a claim having and a word as measured by (2):

$$\mathrm{PMI}(x_i, w_j) = \log \frac{p(x_i, w_j)}{p(x_i) \cdot p(w_j)}. \tag{2}$$

This results in an adjacency matrix A, which is determined using formula (3):

$$A_{ij} = \begin{cases} \mathrm{PMI}(x_i, w_j), & \text{if } w_j \in x_i \\ 0, & \text{otherwise .} \end{cases} \tag{3}$$

3- 3- 4- Graph Representation

The bipartite graph is represented as $\mathcal{G} = (V, \mathcal{E}, \mathbf{X})$ with $V = X \cup W$ being the set of claim and word nodes, $\mathcal{E}$ referring to the set of claim-word edges weighted by PMI, $\mathbf{X} \in \mathbb{R}^{N \times d}$ containing node feature embeddings from Eq. (1).

3- 3- 5- Integration with SLM Logits

We use the outputs from SLMs as auxiliary features for readiness to feed neuro-symbolic fusion. The SLM generates logits of the labels “real” and “fake” for each claim node x_i. Let the combined label space be $Y = \{"real", "fake"\}$. Formula (4) produces a logarithmic distribution for the SLM:

$$s_{i,y} = \sum_{k=1}^{\ell_y} p_{i,y}[L - \ell_y + k, y[k]], \tag{4}$$

Where ℓ_y is the token length of label y, $p_{i,y}$ are log-probabilities from the SLM, and $s_{i,y}$ is the log-likelihood score of label y given claim x_i. Stacking these scores yields the SLM logit matrix as in formula (5):

$$S = \begin{bmatrix} s_{1,\,\text{real}} & s_{1,\,\text{fake}} \\ s_{2,\,\text{real}} & s_{2,\,\text{fake}} \\ \vdots & \vdots \\ s_{N,\,\text{real}} & s_{N,\,\text{fake}} \end{bmatrix} \in \mathbb{R}^{N \times 2} \tag{5}$$

This matrix is aligned with the graph structure, and GNNs can then exploit both node-level embeddings and SLM-derived priors in downstream classification.

3- 4- Graph Neural Networks

After obtaining the constructed heterogeneous claim-word graph, we utilize a family of GNNs for node-wise

classification to identify whether each claim node is real or fake. One of the key benefits of GNNs is that they can effectively propagate information in graph neighborhoods and hence have the potential to explore both semantic and structural dependencies in data.

3- 4- 1- GNN Architectures

We adopted and benchmarked six typical GNN models. Each model represents a diverse design philosophy, from spectral convolutions and attention-based to inductive neighbor sampling or propagation-based. This diversity enabled us to assess the extent to which different message-passing schemes capture the semantic and structural relationship dynamics in the claim-word graph.

Graph Convolutional Network: The Graph Convolutional Network (GCN) model exploits spectral graph theory (which involves manipulating the eigenvalues of the graph Laplacian) to effect convolution operations on graph-structured data. At each layer, a localized filter is applied so that nodes may aggregate information from their immediate neighbors. Our implementation stacks four GCN layers with batch normalization and ReLU nonlinearity after each layer, with dropout applied after each of the hidden layers to prevent overfitting. In the last layer, the node embeddings are projected into a 2D space corresponding to the output of the binary classification problem (real vs. fake). This architecture provides a powerful baseline because it allows for modelling local connectivity with a low computational burden.

Graph Attention Network: While GCN assumes a common weight across all neighbors, Graph Attention Network (GAT) considers weights learned by a masked self-attention mechanism assigned to different neighbors. This allows the model to dynamically weight more useful connections in the graph. Each intermediate layer of our implementation utilizes multi-head attention, and the outputs of the heads are concatenated together to increase expressiveness. The final layer utilizes a single head of attention to produce classification logits. By allowing for variable weighting of neighborhood nodes, GAT works particularly well in heterogeneous graphs in which not all words will be equally important in terms of the veracity of a claim.

GraphSAGE: GraphSAGE allows for inductive learning, which makes it better suited to unseen claims. This is achieved by creating node embeddings using node features aggregated from a sample of its neighboring nodes. We implement mean aggregation, which learns the average of the neighboring node embeddings and combines them with the node features of the target node. Furthermore, we stack multiple layers of GraphSAGE, each one having batch normalization, ReLU activations, and dropout. The result is that we are granted a parameter-efficient representation learning model with propagating localized features, making it well-suited to scaling to large dynamic graphs.

Topology Adaptive Graph Convolution: Topology Adaptive Graph Convolution (TAGNet) works to alter the receptive field so that it is not restricted to neighbors but tests polynomial filters of order K. This allows the model to aggregate signals from higher neighborhoods in a topology-adaptive manner. In our implementation, K was set to $K = 3$, allowing every node to aggregate information from neighbourhoods of size up to three hops away. This means that local contextualization is traded off with global awareness. This ability makes TAGNet particularly suitable for capturing claim-word relations that were not explicitly next to each other but possessed structural similarity across the corpus.

Approximate Personalized Propagation of Neural Predictions (APPNP): Approximate Personalized Propagation of Neural Predictions (APPNP) disentangles feature transformation from knowledge diffusion. It first conducts a Multi-Layer Perceptron (MLP) in the initial layer to transform the node features, and subsequently performs personalized PageRank-based propagation, thus enabling deeper and smoother signal diffusion without over-smoothing of node embeddings. We set up APPNP with two linear layers and ten propagation steps, using a teleport probability α=0.1. This provides a flexible trade-off between locality and global context, allowing claims to describe long-range semantic dependencies in the graph.

Graph Isomorphism Network: Graph Isomorphism Network (GIN) is theoretically motivated around maximizing the discriminatory capabilities of GNNs, approximating the Weisfeiler-Lehman graph isomorphism test. Each layer has an MLP-based aggregator to update the node embeddings, enabling the model to distinguish between node structures that would otherwise appear indistinguishable when viewed through the lens of a simpler aggregator. We implemented multiple stacked layers of GIN, each being a deep MLP and containing batch normalization. The architecture proves particularly effective for capturing structural patterning inherent in relatively sparse graphs.

All models were set to have a hidden dimension $d_h = 256$, dropout $p = 0.5$, included batch normalization, and were trained for up to 500 epochs with the Adam optimizer ($\eta = 0.01$, $weight\ decay = 5\times10^{-4}$). Supervision was applied only to the claim nodes, while evaluation was applied to the validation and test splits. Each claim node given by x_i has as its output a GNN model $f_{\text{GNN}}(\cdot)$ producing a logit vector as in formula (6), where $\mathcal{G}$ is the graph constructed, $C = 2$ is the number of classes (real vs. fake).

$$s_i^{\text{GNN}} = f_{\text{GNN}}(x_i, \mathcal{G}), s_i^{\text{GNN}} \in \mathbb{R}^C, \tag{6}$$

Given the training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i) | i \in I_{\text{train}}\}$, where $y_i \in \{1,\ldots,C\}$ is the true class label, the cross-entropy loss for node i is calculated by formula (7):

$$\ell(s_i, y_i) = -\log \frac{\exp(s_{i,y_i})}{\sum_{c=1}^{C} \exp(s_{i,c})}. \tag{7}$$

The average GNN loss across the training set is determined using equation (8):

$$L_{\text{GNN}} = \frac{1}{\left|I_{\text{train}}\right|} \sum_{i \in I_{\text{train}}} \ell\left(s_i^{\text{GNN}}, y_i\right). \tag{8}$$

Because the ANS dataset is imbalanced (more *real* than *fake* claims), weighted cross-entropy was applied, wherein weights specific to the class were set as being inversely proportional to the frequency of that class. This was to prevent the model from overfitting to the majority class.

3- 5- Zero-shot & Fine-Tune with QLoRA

In addition to graph-based models, we also wanted to explore the potential of LLMs for Arabic claim detection in small-scale contexts. More specifically, we tested two small instruction-tuned models, LLaMA-3.2-1B-Instruct [15] and Qwen2.5-3B-Instruct [16] to see if SLMs, when suitably tuned, can give reasonable predictions regarding veracity in low-resource contexts.

3- 5- 1- Zero-Shot Claim Classification

Both LLaMA-3.2-1B and Qwen2.5-3B were initially evaluated in a zero-shot setting where only instructions on the task in question were given in a unified prompting template, as shown in Figure 2. Let the set of claims be given by $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$, and the label itself by the label space $\mathcal{Y} = \{\text{"real"}, \text{"fake"}\}$. Thus, for each claim x_i and label $y \in \mathcal{Y}$ we form a prompt-label sequence as in (9):

$$z_{i,y} = \text{concat}(x_i, y). \tag{9}$$

The length of the sequence is described by the equation (10), where $T(\cdot)$ is the tokenizer, and $\ell_y = \left|T(y)\right|$ is the label token length:

$$z_{i,y} = T\left(z_{i,y}\right) = [t_1, t_2, \ldots, t_L], \tag{10}$$

The SLM produces logits over the vocabulary as in relation (11):

$$h_{i,y} = M\left(z_{i,y}\right) \in \mathbb{R}^{L \times |V|}. \tag{11}$$

These are normalized into log-probabilities as indicated in relation (12):

$$p_{i,y} = \log \text{softmax}\left(h_{i,y}, \text{dim} = -1\right). \tag{12}$$

The log-likelihood score for label y given claim x_i is computed as relation (13):

$$s_{i,y} = \sum_{k=1}^{\ell_5} p_{i,y}\left[L - \ell_y + k, y[k]\right]. \tag{13}$$

Linking together all the claim scores gives the logit matrix from the SLM output as in (5), the logit matrix reflecting the prior classification signal that was generated by the small LLMs as a result of zero-shot prompting.

```
Determine whether the following Arabic claim is real or fake based on your
knowledge and reasoning.

Analyze the claim carefully. Consider whether the statement is logically
plausible, factually accurate, and aligns with known events or common
knowledge. Respond with only one word: **"Real"** if the claim is likely
true, or **"Fake"** if the claim is likely false.

Claim: "{claim_text}"

Answer:
```

Fig. 2. The structured template for the zero-shot setting.

3- 5- 2- Fine-Tuning with QLoRA

While these zero-shot predictions make for a great baseline, the smaller LLMs are not able to perform any reasoning without an adaptation to a domain. To deal with this, we will apply it to both models with QLoRA. This approach enables efficient fine-tuning through the introduction of low-rank adapters in the quantized weight matrices and thus a much lower compute cost in the fit. Thus, by again fitting LLaMA-3.2-1B and Qwen2.5-3B on the Arabic News Stance dataset with QLoRA, it is possible to obtain reasoning patterns that are aware of veracity.

3- 6- Neuro-Symbolic Fusion

While both the GNNs (Section 3.4) and the SLMs (Section 3.5) yield useful signals towards claim classification, there are limitations in each of the two approaches. GNNs are great at exploiting structural dependencies, but have no grasp of general contextual knowledge; the SLMs enjoy more access to linguistic priors, but have a problem when fitting in noisy or domain-specific low-resource scenarios where there is little or no data. To exploit the qualities of both these approaches, we have constructed a neuro-symbolic fusion principle with SLM linguistics and GNN relational reasoning.

3- 6- 1- Training Objective

The model is defined where the training set $\mathcal{D}_{\text{train}} = \{(x_i, y_i) | i \in I_{\text{train}}\}$, then constitutes a training data set where x_i means a claim vector and $y_i \in \{1,\ldots,C\}$ its ground truth label, where $C = 2$ for the classes real and fake. To each of the claim nodes, the GNN yields logits s_i^{GNN}, and the SLM yields logits s_i^{SLM}. The cross-entropy loss for node i is given by equation (14):

$$\ell(s_i, y_i) = -\log \frac{\exp(s_{i,y_i})}{\sum_{c=1}^{C} \exp(s_{i,c})} \tag{14}$$

Using this definition, the average GNN-specific loss is calculated as in formula (15):

$$L_{\text{GNN}} = \frac{1}{|I_{\text{train}}|} \sum_{i \in I_{\text{train}}} \ell(s_i^{\text{GNN}}, y_i), \tag{15}$$

and the SLM-specific loss is as formula (16):

$$L_{\text{SLM}} = \frac{1}{|I_{\text{train}}|} \sum_{i \in I_{\text{train}}} \ell(s_i^{\text{SLM}}, y_i) \tag{16}$$

The final joint objective is the average of both distinct components, $L = \frac{1}{2}(L_{\text{GNN}} + L_{\text{SLM}})$. This objective ensures that GNN predictions are also consistent with the ground truth labels and the linguistic priors, which are given by the SLMs.

3- 6- 2- Inference and Fusion Strategy

At inference time, the outputs of the GNN and SLM are assembled to obtain the ultimate classification. It is to SoftMax as (17) constraints the logits into probability distributions.

$$\begin{aligned} \mathrm{P}_i^{\text{GNN}} &= \text{softmax}(s_i^{\text{GNN}}), \\ P_i^{\text{SLM}} &= \text{softmax}(s_i^{\text{SLM}}). \end{aligned} \tag{17}$$

A fused probability of claim x_i given by the average of the two distributions, calculated using formula (18):

$$P_i = \frac{1}{2}\left(P_i^{\text{GNN}} + P_i^{\text{SLM}}\right) \tag{18}$$

At last, the predicted label is decided according to the class of the maximum probability with equation (19):

$$\hat{y}_i = \arg \max_{c \in \{1,\ldots,C\}} P(i, c) \tag{19}$$

This fusion approach combines structural logic derived from GNN with semantic priors provided by SLM, resulting in better predictions than the single models.

4- Results and Discussion

Graph-based models provide the relational dependencies and symbolic structures relating the respective entities. We converted the dataset to a graph representation and applied six of the latest and optimally tuned GNN architectures. The results are displayed in Table 2, and of the models tested, the GAT network displayed the best performance, with an F1-score of 0.702, closely followed by a GCN of 0.697 and a GraphSAGE network of 0.688. Other diffusion-method-based algorithms, such as APPNP, also scored competitively (F1 = 0.684). All these performance gains corroborate that GNNs efficiently capture the relational and structural dependencies allied with the claims and the respective words that constitute the claims. Our dataset is in the main textual in representation; thus, language models were naturally efficient models to capture the semantic and linguistic features. Thus, LLaMA-3.2-1B and Qwen2.5-3B were naturally selected as the respective SLMs taken as best cases. Both of which were evaluated initially in a zero-shot scenario. The performance under a zero-shot evaluation of both models was limited (see Table 2), where the results of the F1-score were 0.463 for LLaMA-3.2-1B and 0.506 for Qwen2.5-3B. However, in all cases, Qwen2.5-3B outperformed LLaMA-3.2-1B, which led us to adaptively fine-tune these models further to fully evaluate their respective efficacy. Then Qwen2.5-3B was fine–tuned employing QLoRA, a parameter-efficient adaptation technique. This greatly improved the performance of the model substantially from 0.506 (zero-shot) to an F1-score of 0.673, and a score of 0.686 for accuracy (0.440 0 to

Table 1. Results of the proposed model and comparison with SATA approaches.

Models	Accuracy	Precision	Recall	F1-Score
State-of-the-art Approaches				
AraBERT + 2D-CNN [6]	**0.714**	**0.756**	0.523	0.618
CNN and LSTM [17]	0.670	0.690	**0.670**	**0.680**
Graph Neural Network Techniques				
GCNs [18]	0.721	0.688	0.697	0.697
GATs [19]	**0.730**	**0.698**	**0.708**	**0.702**
GraphSAGE [20]	0.723	0.687	0.688	0.688
TAGNet [21]	0.701	0.682	0.703	0.684
GINNet [22]	0.706	0.671	0.679	0.674
APPNPNet [23]	0.710	0.680	0.692	0.684
Zero-Shot Analysis Using SLMs				
Llama-3.2-1B [15]	0.416	0.464	0.463	0.463
Qwen2.5-3B [16]	0.440	0.507	0.506	0.506
Fine-Tuning SLMs				
Qwen2.5-3B (QLoRA)	**0.686**	**0.665**	**0.683**	**0.673**
Neuro-Symbolic Approach				
GCNs + Qwen2.5-3B (QLoRA)	**0.732**	**0.708**	**0.727**	**0.717**

0.686). This result indicates a relative improvement of more than 50% in the resultant F1-scores, clearly showing that compact fine-tuned SLMs have significant uses for the claim detection model in low-resource settings.

We intended to complement the semantic perspective afforded by SLMs with the symbolic and relational information present in graphs. This way, we adopted a double-perspective approach such that both semantic and relational features were considered at once, resulting in a new neuro-symbolic fusion approach in which the outputs from the GCN model were integrated with the tuned Qwen2.5-3B model. GCN was chosen as the basis model for the neuro-symbolic fusion system because it provides a very strong balance between accuracy, stability, and efficiency concerning computation while providing outputs that can be easily integrated with the features derived from an LLM. This approach led to overall better performance, achieving an accuracy of 0.732 and an F1-score of 0.717, compared to both independent GNN models and independent SLMs. This improvement can be attributed to the complementary nature of the two feature types considered, whereby the GCN has captured the symbolic structure of claim-word dependencies while the Qwen2.5-3B model has provided the semantic priors, together enhanced by tuning. This two-dimensional perspective provided robust predictions even when one of the modalities was less assured.

To provide context for our findings, we compared the presented neuro-symbolic approach against the hybrid AraBERT + 2D-CNN model in [6] reported an F1-score of 0.6188 and accuracy of 0.714 on the ANS dataset. Although this architecture has the advantage of using a powerful pretrained transformer (AraBERT) and CNN feature reduction, its performance is far less than our neuro-symbolic fusion model (9.9 improvement in F1-score for the proposed model). The improvement stems from GNNs' ability to explicitly model structural relations between claims and words, which CNN-based architectures do not possess. Moreover, the union with QLoRA-tuned SLMs ensures that semantic reasoning is a complement to graph structure, leading to greater generalization.

The CNN-LSTM model [17] attained an accuracy of 67%, demonstrating competitive results. However, the sequential nature of the architecture does not fully utilize the graph-structured claim–word interactions. Our GCN + QLoRA model exceeds this baseline by >6% in accuracy and by a substantial margin in F1-score. This speaks to the advantage of combining symbolic relational reasoning (via GNNs) with the knowledge supplied by pretrained language modeling rather than only relying on sequence modeling.

The comparison shows that though transformer-CNN hybrids (e.g., AraBERT + 2D-CNN) and sequential CNN-LSTM models have solid baselines, the neuro-symbolic approach proposed here achieves superior performance through a harmonic combination of structural and contextual signals. By aligning GNN-reasoning with SLM-derived priors (and full representations of word sense), the model achieves the best reported performances for the ANS dataset while exploring a scalable way forward for low-resource languages, where relying solely on either structure or semantics alone may be deficient.

5- Conclusion

In this research, we have addressed the task of Arabic claim detection. This domain poses difficult challenges due to the limited resources available and linguistic complexity. Therefore, we proposed a multi-step approach to deal with these issues, incorporating graph-based learning and SLMs (fine-tuned using QLoRA) and a final neuro-symbolic fusion policy. The main contribution of our work consists of the combination of the GCN with the Qwen2.5-3B (QLoRA) model, which produced the best accuracy (0.732) and F1-score (0.717), outperforming previous Arabic detection models. The main advantages of this approach are that it uses structural reasoning and linguistic priors, leveraging the complementary strengths of graph learning and pretrained language models. At the same time, reliance on small, efficiently fine-tuned models makes this model computationally affordable, which is especially valuable for low-resource languages such as Arabic. The proposed neuro-symbolic approach of this work was compared to the AraBERT + 2D-CNN hybrid model. An improvement of 1.84% in accuracy on the ANS dataset was obtained. In the future, several promising approaches can be taken. One is the expansion of the dataset, which will integrate claims from several Arabic dialects, increasing robustness across regions. Another is the multimodal fusion of these (text with images, metadata) for increased detection while incorporating real-world aspects. Finally, the proposed neuro-symbolic framework can be extended with more complex integration strategies, including reinforcement learning or dynamic weighting of modalities, for even higher accuracy and better generalization.

References

[1] E.C. Choi, E. Ferrara, Fact-gpt: Fact-checking augmentation via claim matching with LLMs, in: Companion Proceedings of the ACM Web Conference 2024, 2024, pp. 883-886.

[2] W.-Y. Kao, A.-Z. Yen, How we refute claims: Automatic fact-checking through flaw identification and explanation, in: Companion Proceedings of the ACM Web Conference 2024, 2024, pp. 758-761.

[3] A. Gupta, A. Singhal, T. Law, V. Rao, E. Duan, R.L. Li, Can LLMs Verify Arabic Claims? Evaluating the Arabic Fact-Checking Abilities of Multilingual LLMs, in: Proceedings of the 1st Workshop on NLP for Languages Using Arabic Script, 2025, pp. 104-113.

[4] A. Farghaly, The Arabic language, Arabic linguistics and Arabic computational linguistics, Arabic computational linguistics, (2010) 43-81.

[5] A. Alshehri, A. AlShabeb, Exploring attitudes, identity, and linguistic variation among Arabic speakers: Insights from acoustic landscapes, International Journal of Arabic-English Studies, 24(2) (2023) 1-16.

[6] N.A. Othman, D.S. Elzanfaly, M.M.M. Elhawary, Arabic Fake News Detection Using Deep Learning, IEEE Access, 12 (2024) 122363-122376.

[7] L. Majer, J. Šnajder, Claim Check-Worthiness Detection: How Well Do LLMs Grasp Annotation Guidelines?, in: Proceedings of the Seventh Fact Extraction and Verification Workshop (FEVER), 2024, pp. 245-263.

[8] Z.M. Hüsünbeyi, T. Scheffler, Ontology Enhanced Claim Detection, CoRR, (2024).

[9] T. Chen, A. Stefanidis, Z. Jiang, J. Su, Improving biomedical claim detection using prompt learning approaches, in: 2023 IEEE 4th International Conference on Pattern Recognition and Machine Learning (PRML), IEEE, 2023, pp. 369-376.

[10] L.-H. Lee, Y.-H. Cheng, J.-H. Yang, K.-Y. Tien, NCUEE-NLP at SemEval-2023 Task 8: Identifying Medical Causal Claims and Extracting PIO Frames Using the Transformer Models, in: Proceedings of the 17th International Workshop on Semantic Evaluation (SemEval-2023), 2023, pp. 312-317.

[11] A. Wührl, R. Klinger, Entity-based Claim Representation Improves Fact-Checking of Medical Content in Tweets, in: Proceedings of the 9th Workshop on Argument Mining, 2022, pp. 187-198.

[12] A. Wührl, R. Klinger, Claim Detection in Biomedical Twitter Posts, in: Proceedings of the 20th Workshop on Biomedical Language Processing, 2021, pp. 131-142.

[13] J. Khouja, Stance Prediction and Claim Verification: An Arabic Perspective, in: Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER), 2020, pp. 8-17.

[14] T. Zerrouki, pyarabic, An Arabic language library for Python, (2010).

[15] Llama-3.2-1B-Instruct, (Sept 25, 2024).

[16] Q. Team, Qwen2.5: A Party of Foundation Models, (2024).

[17] S.E. Sorour, H.E. Abdelkader, AFND: Arabic fake news detection with an ensemble deep CNN-LSTM model, Journal of Theoretical and Applied Information Technology, 100(14) (2022) 5072-5086.

[18] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in: Proceedings of the AAAI conference on Artificial Intelligence, 2019, pp. 7370-7377.

[19] A.G. Vrahatis, K. Lazaros, S. Kotsiantis, Graph attention networks: a comprehensive review of methods and applications, Future Internet, 16(9) (2024) 318.

[20] W. Hamilton, Z. Ying, J. Leskovec, Inductive representation learning on large graphs, Advances in neural information processing systems, 30 (2017).

[21] J. Du, S. Zhang, G. Wu, J.M. Moura, S. Kar, Topology Adaptive Graph Convolutional Networks, (2018).

[22] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How Powerful are Graph Neural Networks?, in: International Conference on Learning Representations, (2018).

[23] J. Gasteiger, A. Bojchevski, S. Günnemann, Predict then Propagate: Graph Neural Networks meet Personalized

PageRank, in: International Conference on Learning Representations (2018).

HOW TO CITE THIS ARTICLE
M. Dina Falah1, H. Faili, Neuro-Symbolic Claim Detection for Arabic: Integrating Graph Neural Networks with Efficiently Fine-Tuned Small Language Models, AUT J. Model. Simul., 58(1) (2026) 23-32.

DOI: 10.22060/miscj.2025.24744.5431