EA^{2}N: Evidence-based AMR Attention Network for Fake News Detection

Shubham Gupta, Abhishek Rajora, Suman Kundu, *Member, IEEE*

Abstract*—Proliferation of fake news has become a critical issue in today's information-driven society. Our study includes external knowledge from Wikidata which allows the model to cross-reference factual claims with established knowledge. This approach deviates from the reliance on social information to detect fake news that many state-of-the-art (SOTA) fact-checking models adopt. This paper introduces EA^{2}N, an Evidence-based AMR (abstract meaning representation) Attention Network for Fake News Detection. EA^{2}N utilizes the proposed Evidence based Abstract Meaning Representation (WikiAMR) which incorporates knowledge using a proposed evidence-linking algorithm, pushing the boundaries of fake news detection. The proposed framework encompasses a combination of a novel language encoder and a graph encoder to detect fake news. While the language encoder effectively combines transformer-encoded textual features with affective lexical features, the graph encoder encodes semantic relations with evidence through external knowledge, referred to as WikiAMR graph. A path-aware graph learning module is designed to capture crucial semantic relationships among entities over evidence. Extensive experiments support our model's superior performance, surpassing SOTA methodologies with a difference of 2-3% in F1-score and accuracy for Politifact and Gossipcop datasets. The improvement due to the introduction of WikiAMR is found to be statistically significant with t-value less than 0.01. *Code repository: https://github.com/brillard1/EA2N-Fake-News-Detection-Framework

***Index Terms*—misinformation detection, WikiAMR, evidence linking algorithm, external knowledge graph.**

I. Introduction

SOCIAL media has revolutionized the exchange of information by enabling people to obtain and share news online. However, with the growing popularity and convenience of social media, the dissemination of fake news has also escalated. The deliberate distortion and fabrication of facts in fake news have severe negative consequences for individuals and society [1]. Therefore, it is crucial and socially advantageous to detect and address fake news on social media.

Significant efforts have been made in the direction of fake news detection in the past decade. 'Veracity problem on Web' was first introduce by [2] where the authors proposed a solution named TURTHFINDER. It verifies the news based on different genuine websites. In contrast, Feng et al. [3] and Jing et al. [4] used manually crafted textual features to detect fake news. Later, many researchers used LSTM and RNN based methods [5], [6] for the purpose. Deep learning methods used therein is able to learn text features out of the article. Recently

S. Gupta, A. Rajora, and S. Kundu are with Department of Computer Science and Engineering, Indian Institute of Technology Jodhpur, India - 342030 .
E-mail: {gupta.37, rajora.1, suman}@iitj.ac.in

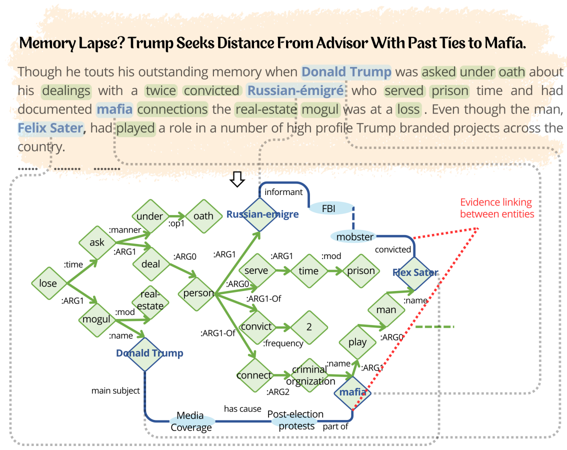

Fig. 1. An example of evidence linking in AMR graph constructed over Politifact news article.

[7]–[10], external knowledge is incorporated alongside the textual features to improve fake news detection models. In 2021, KAN [9] and CompareNet [11] were proposed which uses evidence from the external source Wikidata. On the contrary, FinerFact [10] and Dual-CAN [12], incorporates social information that supports authenticity of the news article.

Despite the significant achievements, these models struggle to maintain longer text dependencies and less effective to capture complex semantic relations such as events, locations, trigger words and so on. Additionally, the way of incorporating external knowledge into these models is not highly reliable and time-consuming. For example, KAN only considers single-entity contexts and CompareNet selects first paragraph of an entity from Wikipedia which fails to link context between two entities. On the other hand, FinerFact gathers supported claims from social platforms which is time-consuming. Although social authenticity produces good results, this information can be manipulated by social media users for personal gain. Also, these methods rely on social diffusion and engagement patterns that are sometimes challenging to capture accurately. In order to tackle these challenges, our study effectively uses complex semantic relations of news articles and evidence found in Wikidata5M [13] with the help of a novel graph representation.

In this paper, we present a novel model for detecting fake news that leverages a semantically enriched knowledge base to classify a news article as real or fake. Our model incorporates Abstract Meaning Representation (AMR) [14] to understand the logical structure of sentences. Further, the model establishes relations between entities found in AMR graph, through a new evidence-linking algorithm. The algorithm utilizes Wikidata to connect evidence, leading to the formation of a graph referred as WikiAMR. We utilize the Wikidata knowledge graph because it provides historical evidence knowledge related to claims and has less chances of impure knowledge as compare to social information. Additionally, the knowledge-based approach is inherently explainable, as it can directly reference specific pieces of validated information in response to doubtful claims, potentially increasing user trust in the fact-checking system. To the best of our knowledge, this is the first study to examine the evidence-based semantic features of AMR graphs for fake news detection. For the illustration, Fig. 1 shows an example of WikiAMR constructed over a news article using Wikidata. In this, a linkage is established between the entities '$Donald\ Trump$' and '$Mafia$', wherein the nodes '$Media\ Coverage$' and '$Post-election\ protest$' are connected by relations such as '$main\ subject$', '$has\ cause$', and '$part\ of$' in the path. Similarly, a connecting evidence path emerges between '$Russian-emigre$' and '$Flex\ Ster$'. These instances serve as valuable evidence to assess the credibility of news content. Next, to encode the WikiAMR graph, we employ a path-aware graph learning module. This module uses relation-enhanced global attention that focus on important relations over entities. It then computes the attention score considering the entities and their relations. By modifying the Graph Transformer [15] for entities, our model can effectively reason over the relation paths within the WikiAMR graph. Finally, the representations of language and WikiAMR graph are fed into a classification layer using transformer to predict the veracity of the news. We have experimented our model on two publicly available large datasets Politifact and Gossipcop and it is evident form the results that our model outpeforms stat-of-the-art models with a difference of 2-3% in F1-score and accuracy for both datasets. The key contributions of our research are as follows:

- Introduction of EA^{2}N, a novel Evidence-based AMR Attention Network for Fake News Detection, reasoning over evidence linked through external knowledge.
- Introduction of WikiAMR graph, a novel graph structure that includes undirected evidence paths, extracted form external knowledge graph, between entities of AMR constructed from text document.
- Evidence Linking Algorithm to generate WikiAMR, from entity-level and context-level filtering using proposed relatedness objective function to enhance model performance.
- Comprehensive evaluation of EA^{2}N against state-of-the-art techniques, demonstrating its superior performance and effectiveness.

This paper is organized in a following manner such as: In Section II, we discuss about the literature around the fake news detection and state problem formally in Section III. Proposed methodology is described in detail in Section IV, while Section V outlines the experimental framework used in the process. Section VI and Section VII reports the comparative study and evaluates the various findings. In last, Section VIII summarizes the work presented and explores potential avenues for further investigation.

II. Related Works

In this section, we delve into brief of the approaches employed in the detection of fake news. We have categorized the relevant studies into three components: textual-based methods, knowledge-aware methods, and AMR-based methods. A comprehensive explanation of chronological development is provided in the subsequent subsections.

a) Textual-based: These approaches primarily rely on the textual content extracted from articles to verify the authenticity of news. In the early times, the emphasis was primarily on developing a supplementary collection of manually created features rooted in linguistic characteristics [3]–[5], [16]. These early studies demanded extensive efforts to assess the efficacy of these manually crafted features. Nguyen et al. [17] proposed a spread interdiction problem of fake news that seek the most effective nodes for removal under Linear Threshold model to control the spread. Authors proposed a CPU-GPU method that scale to networks with billions of edges, yet possess rigorous theoretical guarantee on the solution quality. FakeFlow presented in [18] utilizes a text with lexical features to classify news as fake or real. It learns this flow by combining topic and affective information extracted from text. Early detection of fake news is facilitated in many work [19]–[21]. Recently, Fang et al. [21] proposed a unsupervised way to detect the rumors in which they proposed a tree based variational autoencoder that reconstructs the sentiment labels along the propagation tree of a factual tweet. However, their effectiveness is limited as they overlook auxiliary knowledge that could aid in news verification.

b) Knowledge-aware: These methods utilize auxiliary knowledge to aid in the process of news verification. Firstly, Popat et al. [7] introduced a model that retrieves external articles in response to a claim and models the interactions between them. Furthermore, the DeClarE model [8], combines evidence-based embeddings with textual features to assess the truthfulness of news. However, these models do not adequately address the evaluation of evidence justifications related to a claim. Later, Monti et al. [22] extends classical CNNs to operate on graphs by analyzing the user activity, content, social graph etc., while [23] found user profile features to be useful in fake news detection. A co-attention model is employed in [24] that uses both news content and social context. Later, entity linking method is used to capture entity descriptions from Wikidata and integrate them into their models for the identification of fake news [9], [11]. In 2021, KAN [9] and CompareNet [11] considered external knowledge from Wikidata to expand domain knowledge and [25] proposed KGML a low-resource text classification model that bridges the gap between meta-training and meta-testing tasks by leveraging the external knowledge bases. The model GET [26] aligns with aspects of KAN in its utilization of graph structures for fake news detection, but it adopts 1-layer Graph Gated Neural Network (GGNN) instead of transformers. In [10], Jin et al. presented Finerfact, a fine-grained reasoning framework using social information to detect fake news. Dual-CAN [12] method takes news content and social media replies as external knowledge for the purpose. Recently fake news detection methods [27]–[29] have been proposed which proves that better detection requires external knowledge acquisition. Wu et al. [27] proposed GETRAL, a unified graph-based semantic structure mining framework with contrastive learning that captures the long-distance semantic dependency among dispersed relevant snippets via neighborhood propagation. A knowledge-guided dual-consistency network [28] is proposed to detect rumors with multimedia contents. It uses two consistency detection subnetworks to capture the inconsistency at the cross-modal level and the content-knowledge level simultaneously. In order to address biases of out-of-distribution news and evidence content, Liu et al. [29] proposed DAL that reversely optimizes news-aspect and evidence-aspect debiasing discriminators to mitigate the impact of news and evidence content biases. Inspired from these, our approach incorporates external knowledge to improve performance by leveraging information related to entities.

c) AMR-based: AMR as introduced by [14], represents relations between nodes using PropBank, frameset and sentence vocabularies. It utilizes over hundred semantic relations, including negation, conjunction, command, and wikification. It aims to represent different sentences with the same semantic meaning using the same AMR graph. Various NLP fields, such as summarization [30], event detection [31], question answering [32] etc., have effectively used AMR. Despite its wide range of applications in NLP, AMR has not been much investigated to capture complex semantic relations of entities along with evidences to support the claims made by entiities in documents for fake news detection. Recently, Zhang et al. [33] used AMR for the identification of out-of-context multimodal misinformation in detection of multimodal discrepancies between visual and textual data for out-of-context misinformation. Also, in one of our preliminary study [34], we encoded textual information using Abstract Meaning Representation (AMR) and analyzed how AMR's semantic relations affect the final veracity of news. However, this study does not provide sufficient justification or evidence support for the relationships among entities within the AMR graph. AMR 1.0 (LDC2014T12) and AMR 2.0 (LDC2017T10) datasets are commonly used for generating and evaluating these representations. Notably, [35] achieved high performance on these datasets using BERT. By incorporating AMR, we enhance the capability of detection model to identify and analyze the intricate semantic structures present in news documents.

III. Problem Statement

Our objective is to perform binary classification of news articles, as either real $(y = 0)$ or fake $(y = 1)$. Formally, given a news article $\mathcal{S} \in C$, where C is collection of news articles, the task is to learn a function f such that $f : f(C) \rightarrow y$, where $y \in \{0, 1\}$ represents the ground truth labels of the news articles.

IV. Evidence-based AMR Attention Network (EA^{2}N)

We present our proposed model, Evidence-based AMR Attention Network (EA^{2}N), depicted in Fig. 3 here in this section. This novel architecture comprises three key components: Graph Encoder, Language Encoder, and Classification Module. Before describing the components in the subsequent sections, let us first present the novel WikiAMR representation and its generation used in the Graph Encoder module.

A. External Knowledge enhanced Abstract Meaning Representation (WikiAMR)

In this study, we propose a new representation of AMR, named as **WikiAMR**, where external knowledge is incorporated to provide the justification over complex relationship between entities derived from the textual content. Proposed WikiAMR representation is denoted as $\mathcal{G}^{WikiAMR}$, comprises interconnected undirected paths between entities nodes in base AMR $\mathcal{G}^{amr}(\mathcal{V}^{amr}, \mathcal{E}^{amr})$ generated from the text. This structure facilitates reasoning about evidences present in external knowledge along with the directed acyclic structure extracted from the text document in $\mathcal{G}^{amr}$. Mathematically, it can be represented as follows:

$$\mathcal{G}^{WikiAMR} = \mathcal{G}^{amr} \cup \sum_{s,d} \mathcal{P}^{wiki}(v_s^{amr}, v_d^{amr}) \tag{1}$$

Here, $\mathcal{P}^{wiki}$ is the evidence path between v_s^{amr} and v_d^{amr} $(v_s^{amr}, v_d^{amr} \in \mathcal{V}^{amr})$ and $\sum$ denotes the graph generated from the set of evidence paths.

1) Evidence Integration in WikiAMR: We first extracted the base AMR graph [14] from an article into a network of nodes and edges, capturing the relationships between different entities. As stated before, for an article $\mathcal{S}$, we represent the AMR graph as $\mathcal{G}^{amr} = (\mathcal{V}^{amr}, \mathcal{E}^{amr})$.

We propose an evidence linking algorithm to extract evidence rich paths from external knowledge graph among entities in AMR. Given $\mathcal{G}^{wiki} = (\mathcal{V}^{wiki}, \mathcal{E}^{wiki})$, a Wikidata knowledge graph, the algorithm integrates $\mathcal{G}^{wiki}$ with AMR by entity-level filtering (ELF) and context-level filtering (CLF). ELF assesses the relevance between entities using a $Relatedness(\cdot)$ function. If the relevance exceeds the ELF threshold (γ), it initiates the CLF process to link evidence between entities.

Entity-Level Filtering (ELF): During ELF, pairs of entities within the AMR graph are examined for their corresponding representations and relevance in Wikidata. The relevance between source and destination entities is calculated as:

$$\mathcal{R}_{ELF}^{(s,d)} = Relatedness(v_s^{wiki}, v_d^{wiki}) = Tag(v_s^{wiki}, v_d^{wiki}) \tag{2}$$

where v_s^{wiki} and v_d^{wiki} are the entity representation of $(v_s^{amr}$ and $v_d^{amr})$ found in Wikidata. $Tag(i, j)$ is the tagme score [36] summed up from i to j, If $\mathcal{R}_{ELF}^{(s,d)}$ exceeds γ, the entities v_s^{amr} and v_d^{amr} are related in Wikidata. This implies the existence of potential evidence path to be attached between them.

Context-Level Filtering (CLF): The CLF algorithm determines the relevance between v_s^{amr} and v_d^{amr}. The CLF follows the search path in the knowledge graph from the starting entity v_s^{amr} to the destination entity v_d^{amr}.

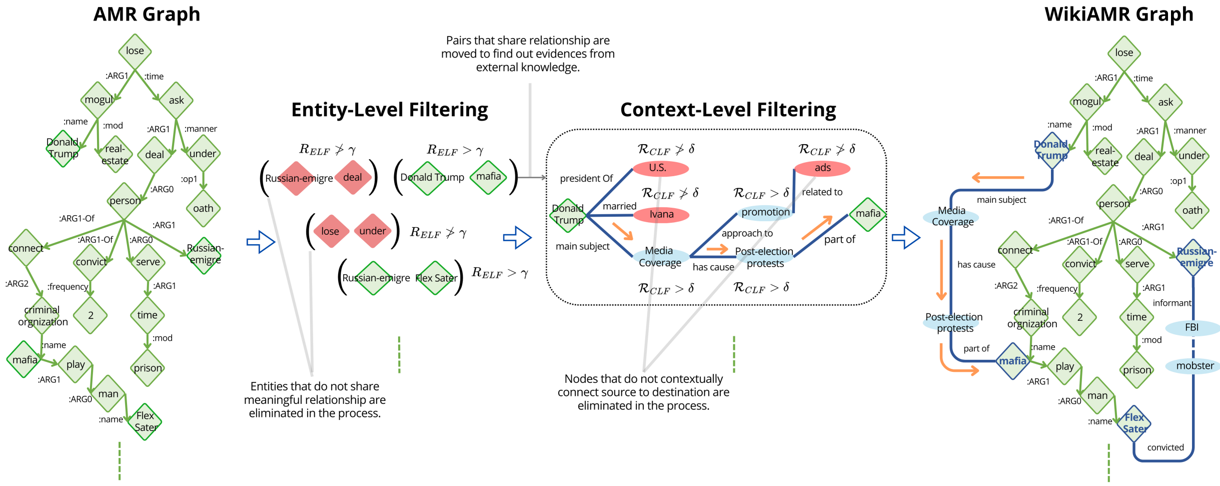

Fig. 2. Generation of WikiAMR using proposed Evidence Linking Algortihm.

For each entity $v_i^{wiki} \in \mathcal{V}_{wiki}$ lying in the evidence path between v_s^{amr} and v_d^{amr}, the relevancy is calculated between v_i^{wiki} and v_d^{amr} to predict the possibility of finding a rich evidence path between source and destination:

$$\begin{aligned}\mathcal{R}_{CLF}^{(s,d)} &= Relatedness(v_i^{wiki}, v_d^{wiki}) \\ &= \frac{Tag(v_s^{wiki}, v_i^{wiki})}{n_i + \epsilon} + \frac{Tag(v_i^{wiki}, v_d^{wiki})}{n_m - n_i + \epsilon}\end{aligned} \tag{3}$$

where n_i and n_m represent the current hops and maximum hops respectively, a very small value, ϵ is added to avoid division by zero. The first term averages the Tagme score from source (v_s^{wiki}) to current node (v_i^{wiki}) and the second term averages the heuristic value of Tagme score from current node (v_i^{wiki}) to destination (v_d^{wiki}). If $\mathcal{R}_{CLF}^{(s,d)}$ is above the CLF threshold (δ), the entity v_i^{wiki} is linked with the next entity in the Wikidata path until v_d^{amr} is reached. This process leads to the attachment of evidence information between entities, enriching the AMR graph with external knowledge. The detailed algorithm to generate WikiAMR using ELF and CLF is presented in Algo. 1.

$$\begin{aligned}
&(s \quad / \ seek-01 \\
&\qquad :arg0 \ (p \ / \ person \\
&\qquad\quad :name \ (n \ / \ name : op1 \text{ “}Donald \ Trumph\text{”})) \\
&\qquad :arg1 \ (d \ / \ distance-01 \\
&\qquad\quad :arg1 \ p \\
&\qquad\quad :arg2 \ (p2 \ / \ person \\
&\qquad\qquad :arg0-of \ (a \ / \ advise-01) \\
&\qquad\qquad :arg1-of \ (t \ / \ tie-01) \\
&\qquad\qquad\quad :name \ (n2 \ / \ name \ \ :op1 \text{ “}mafia\text{”}) \\
&\qquad\qquad\quad :time \ (p3 \ / \ past)))
\end{aligned}$$

The entire process of creating WikiAMR using ELF and CLF is illustrated with an example in Fig. 2. Let us consider the sentence: "*Donald Trumph seeks distance from advisor with past ties to mafia.*" The corresponding AMR graph is presented above. The AMR graph is a directed acyclic graph that represents a hierarchical structure with nodes denoting entities ($Donald\ Trumph$, $mafia$, $seek$, etc). Edges ($arg0, arg1, name$, etc) capture the relationships between

Algorithm 1 Evidence Linking Algorithm

Input: AMR graph $\mathcal{G}^{amr}$, Wikidata graph $\mathcal{G}^{wiki}$, γ, δ
Output: WikiAMR graph $\mathcal{G}^{WikiAMR}$

```
function CLF(start, goal, γ, δ):
    path ← [start], relation_path ← []
    while path is notempty do
        Pick n_cur, last indexed node from path
        if n_cur = goal then
            Extract relationships from G^wiki
            and append them in relation_path
            return relation_path
        end if
        for each neighbor n_adj of n_cur in G^wiki
                & n_adj ∉ path do
            Get R_CLF between n_adj, goal
            if (R_CLF > δ) then
                Append n_adj to path
            end if
        end for
    end while
end function

function ELF(G^amr, G^wiki, γ, δ):
    evidence ← []
    for each node pair u, v ∈ G^amr do
        Get relatedness R_ELF between
        start node u, and goal node v
        if (R_ELF > γ): then
            path = CLF(u, v, γ, δ);
            evidence.append(path);
        end if
    end for
    Integrate G^amr and Relation paths to get
    G^WikiAMR
    return G^WikiAMR
end function
```

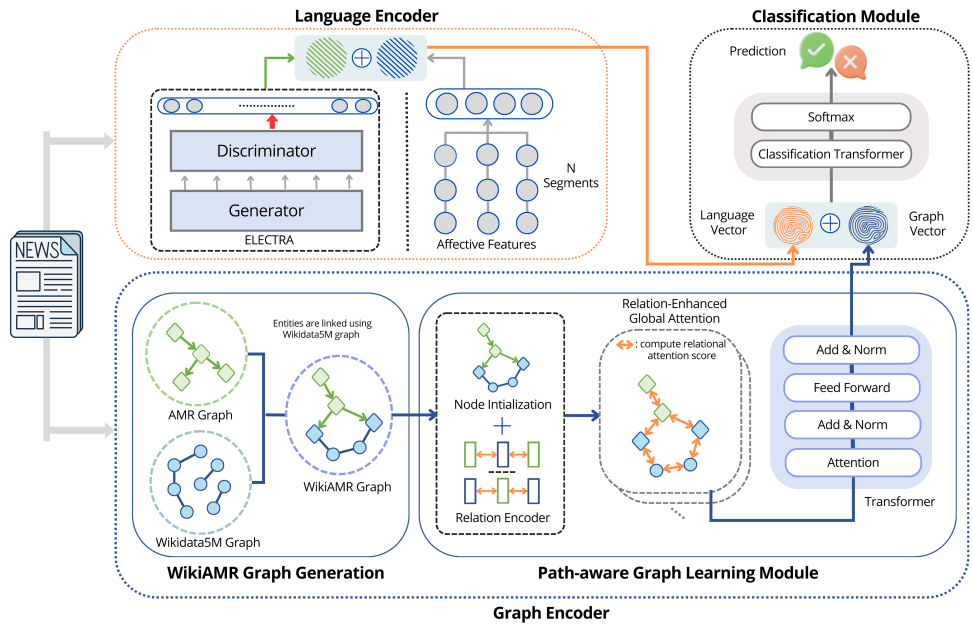

Fig. 3. EA^{2}N framework for fake news detection.

these entities, forming a semantically structured representation of $\mathcal{S}$.

Suppose we want to link the entities $Donald\ Trumph$ and $Mafia$, the relevance $\mathcal{R}_{ELF}$ between them is calculated using $Relatedness(\cdot)$ function. If $\mathcal{R}_{ELF} > \gamma$, the entities $Donald\ Trumph$ and $Mafia$ are identified as relevant, potentially holding evidence path between them. Upon identifying relevant entities, the $\mathcal{R}_{CLF}$ between the next entity v_i^{wiki}, $Media\ Coverage$ and the destination entity $Mafia$ is computed. If $\mathcal{R}_{CLF} > \delta$, entity v_i^{wiki} is linked with entity v_{i+1}^{wiki} in the AMR graph until the destination is reached. This results in the attachment of relevant evidence information. This is repeated over all the pair of entities in AMR and the final graph is represented as WikiAMR.

B. Graph Encoder

The Graph Encoder module plays a crucial role in transforming the language representation into a structured and abstract form. We use novel External Knowledge enhanced Abstract Meaning Representation (WikiAMR) described above to encode the meaning of the news article in a graph structure. The graph encoder comprises the path-aware graph learning module which processes WikiAMR graph and encode the evidence relations using Relation Path Encoder.

1) Path-aware Graph Learning Module: This module plays a crucial role in EA^{2}N by generating informative features from the enriched WikiAMR graph obtained. These features capture essential semantic relationships and enable the model to gain a deeper understanding of the information present in the news articles. The module involves a Graph Transformer, which leverages multi-head attention mechanisms to process the WikiAMR representation in a way that facilitates effective reasoning and representation learning.

Relation Path Encoder The obtained WikiAMR is passed to the node intialization and relation encoder to get the encoding of WikiAMR in $\mathbb{R}^{N\times L\times D}$, where N is the batch size, L is the maximum sequence length, and D is the dimension of the graph encoding.

To facilitate the model's recognition of explicit graph paths from $\mathcal{G}^{WikiAMR}$, the relation encoder is applied to capture the shortest path between two entities. The sequence representing this path is transformed into a relation vector using a Gated Recurrent Unit (GRU) based RNN [37]. The mathematical representation of this relation encoding is given by:

$$\overrightarrow{p}_t = \mathrm{GRU}_f(\overrightarrow{p}_{t-1}, sp_t)\ \overleftarrow{p}_t = \mathrm{GRU}_g(\overleftarrow{p}_{t+1}, sp_t)$$

Here, sp_t signifies the shortest path of the relation between the two entities. As per [15]'s paper, in order to calculate the attention score, the final relational encoding r_{ij} is divided into two separate encodings: $r_{i\to j}$ and $r_{j\to i}$, which are obtained using a linear layer and the parameter matrix W_r.

$$r_{ij} = [\overrightarrow{p}_n; \overleftarrow{p}_0],\ \ [r_{i\to j}; r_{j\to i}] = W_r r_{ij}$$

The Graph Transformer processes the input $\mathcal{G}^{WikiAMR}$ using multi-head attention mechanism. Then the attention scores α_{ij} are computed based on both the entity representations and their relation representation.

$$\begin{aligned} \alpha_{ij} &= g(e_i, e_j, r_{ij}) \\ &= (e_i + r_{i\to j})W_q^T W_k(e_j + r_{j\to i}) \\ &= \underbrace{e_i W_q^T W_k e_j}_{(a)} + \underbrace{e_i W_q^T W_k r_{j\to i}}_{(b)} \\ &\quad + \underbrace{r_{i\to j} W_q^T W_k e_j}_{(c)} + \underbrace{r_{i\to j} W_q^T W_k r_{j\to i}}_{(d)} \end{aligned} \tag{4}$$

The attention weights here are computed to work over entities based on relations and each term in Eq. 4 holds an explanation. The term (a) signifies content-based addressing, (b) and (c) capture the source-dependent and target-dependent relation bias, and (d) embodies a universal relation bias, encompassing a broader perspective on relation interactions. Collectively, this equation explains a comprehensive mechanism for the model to reason and weigh entity-relation interactions.

Graph Transformer for WikiAMR Learning Graph Transformer applies self-attention to capture dependencies between different positions within each WikiAMR graph representation. The encoder consists of multiple identical blocks, of which the core is multi-head attention. The model computes attention weights for the encoded paths to learn the enhanced representations. Given a set of attention heads H, each head computes distinct Query (Q_i), Key (K_i), and Value (V_i) matrices, which are then linearly combined through learnable weight matrices (W_i) to produce the final attended representation:

$$A_i = Attn(Q_i, K_i, V_i) = softmax\left(\frac{Q_i K_i^T}{\sqrt{D'}}\right) V_i \quad (5)$$

$$\mathrm{A} = Concat([\mathrm{A}_1, \mathrm{A}_2, ..., \mathrm{A}_h]) W_\mathrm{H} \quad (6)$$

Here, h, A_i and W_H represent the number of attention heads, output of the i^{th} head, and learnable weight matrix. This dynamic method enhances intricate semantic extraction.

After computing the attention weights, the Graph Transformer (GT) encodes the integrated WikiAMR representations $\mathcal{G}^{WikiAMR}$ as follows:

$$\mathcal{Z}^g = \mathrm{GT}(\mathcal{G}^{WikiAMR}, A) \in \mathbb{R}^{N \times L \times D} \quad (7)$$

Where $\mathcal{Z}^g$ represents the final graph embedding obtained from the Graph Transformer.

C. Language Encoder

We utilized the ELECTRA [38] model to encode sequences of tokens in sentences. Let a news article S be denoted as tuple (T, B), where T represents the title and B represents the body text of the news article, both containing a set of words $\{w_1, w_2, \ldots, w_n\}$. In order to create the final text for input, we concatenate the title and body using a special "[SEP]" tag. After that, we initially tokenize these words and fed into the ELECTRA model to obtain the final layer embedding H^i in the following manner:

$$S = Concat([T, B]) \quad (8)$$

$$[\mathrm{H}_S^0, \mathrm{H}_S^1, \ldots, \mathrm{H}_S^{N_S}] = \mathrm{ELECTRA}(S), \mathcal{Z}^S = \mathrm{H}_S^0 \quad (9)$$

We extracted affective lexical features [18] such as emotions, sentiment, morality, hyperbolic, and imagebility from the n different segments $\{s_1, s_2, \ldots, s_n\}$ of article S to enhance the capabilities of the language encoder. This enhance the capabilities to differentiate documents more effectively by capturing the distribution of the features segment wise. Each feature vector, a_{s_j}, of j-th segment $s_j \in S$ is represented with a term frequency that takes into account the articles' length as a weighting factor. This approach allows us to effectively capture and represent the distinctive characteristics of each segment in the article, accounting for variations in segment lengths. The resulting language vector, denoted as $\mathcal{Z}^l$, is obtained by concatenating the vector embedding derived from BERT with the representation vector formed by integrating the affective features as:

$$\mathcal{Z}^{affect} = Concat([a_{s_1}, a_{s_2}, \ldots a_{s_n}]) \quad (10)$$

$$\mathcal{Z}^l = Concat([\mathcal{Z}^S, \mathcal{Z}^{affect}]) \in \mathbb{R}^{N \times L \times D'} \quad (11)$$

Here D' is the dimension of the language feature vector.

D. Classification Module

The final stage of EA^{2}N involves the Classification Module, which takes the semantically-informed WikiAMR representation and the enriched language features to produce the fake news predictions.

We concatenate the graph features $\mathcal{Z}^g$ and language features $\mathcal{Z}^l$ to create the final fused embedding:

$$\mathcal{Z} = Concat([\mathcal{Z}^g, \mathcal{Z}^l]) \in \mathbb{R}^{N \times L \times (D+D')} \quad (12)$$

Finally, we pass $\mathcal{Z}$ through a classification transformer (CT) followed by a softmax layer to obtain the final probabilites Y_pred over real and fake.

$$f(Z) = softmax(\mathrm{CT}(\mathcal{Z})) \in \mathbb{R}^{N \times 2}, \; Y_\text{pred} = argmax(f(Z)) \quad (13)$$

The comprehensive EA^{2}N model, leveraging WikiAMR using external knowledge integration, affective features, and attention mechanisms, offers a powerful and novel approach in the domain of Fake News Detection.

V. Experimental Setup

A. Dataset and Evaluation Metric

In order to assess the effectiveness of EA^{2}N, we perform experiments on two benchmark datasets, namely, PolitiFact and GossipCop [39]. These datasets consist of 815 and 7,612 news articles, respectively, along with labels assigned by domain experts. We evaluate our model using a set of metrics, including Precision (Pre), Recall (Rec), F1-score, Accuracy (Acc), and Area Under the ROC curve (AUC). We conduct 5-fold cross-validation and report the average results.

B. Baselines

In our evaluation, we contrast our EA^{2}N model with various state-of-the-art baselines, categorized into two groups. The first group utilizes only textual information (**SVM** [40], **DTC** [41], **RFC** [42], **GRU-2** [4], **FF** (FakeFlow) [18]), while the second incorporates auxiliary knowledge in addition to textual features (**B-TransE** [43], **KCNN** [44], **GCAN** [24], **KAN** [9], **FinerFact** [10]).

Below, we provide a description of each group:

- **Textual-based (T)** group comprises models that utilize textual information from news article for detecting fake news. For instance, **SVM** [40], which is a classifier that

TABLE I
COMPARATIVE STUDY OF OUR MODEL EA^{2}N W.R.T. DIFFERENT BASELINES.

	Method	PolitiFact					GossipCop				
		Pre	Rec	F1	Acc	AUC	Pre	Rec	F1	Acc	AUC
T	SVM	0.7460	0.6826	0.6466	0.6694	0.6826	0.7493	0.6254	0.5955	0.6643	0.6253
	RFC	0.7470	0.7361	0.7362	0.7406	0.8074	0.7015	0.6707	0.6691	0.6918	0.7389
	DTC	0.7476	0.7454	0.7450	0.7486	0.7454	0.6921	0.6922	0.6919	0.6959	0.6929
	GRU-2	0.7083	0.7048	0.7041	0.7109	0.7896	0.7176	0.7079	0.7079	0.7180	0.7516
	FF	0.8462	0.7923	0.8193	0.8574	0.8627	0.7263	0.7352	0.7307	0.7563	0.7616
T+K	B-TransE	0.7739	0.7658	0.7641	0.7694	0.8340	0.7369	0.7330	0.7340	0.7394	0.7995
	KCNN	0.7852	0.7824	0.7804	0.7827	0.8488	0.7483	0.7422	0.7433	0.7491	0.8125
	GCAN	0.7945	0.8417	0.8345	0.8083	0.7992	0.7506	0.7574	0.7709	0.7439	0.8031
	KAN	0.8687	0.8499	0.8539	0.8586	0.9197	0.7764	0.7696	0.7713	0.7766	0.8435
	FinerFact	0.9196	0.9037	0.9172	0.9092	0.9384	0.8615	0.8779	0.8685	0.8320	0.8637
Ours	EA^{2}N	**0.9333**	**0.9324**	**0.9328**	**0.9318**	**0.9523**	**0.8947**	**0.8865**	**0.8906**	**0.8713**	**0.9014**

uses extracted features, **DTC** [41], **RFC** [42] that employs identified characteristics and hand crafted features of news articles, and **GRU-2** [4], a model designed to capture higher-level feature interactions across different time steps. Recent state-of-the-art, **FF** (FakeFlow) [18] model identify fake news by combining topic and affective information extracted from text.

- **Textual+Knowledge-aware (T+K)** group consists model that leverages external knowledge such as knowledge graphs and social knowledge about the online posts. For instance, **B-TransE** [43] combines positive and negative single models to leverage knowledge graphs and textual information. **KCNN** [44] utilizes CNN to learn news representations enriched with external knowledge, and **GCAN** [24] is graph-aware co attention network that uses social information like tweet comment, retweet etc. Recent methods in this category include **KAN** [9], a knowledge-aware attention network that incorporates external knowledge, and **FinerFact** [10], a reinforcement-based approach that incorporates human knowledge gathered from social information as evidence.

VI. RESULTS

We used various transformer based model for textual encoding and reported the best results for EA^{2}N using ELECTRA in the table. Table I shows a comparative analysis of EA^{2}N against various models. The standard deviations for accuracy and F1-score metrics in Politifact are 2.17 and 1.82, respectively and in Gossipcop, these standard deviations stand at 2.35 and 2.08. The table clearly demonstrates that our model, EA^{2}N, outperforms the state-of-the-art model, FinerFact, by 1.6%, 2.3% in terms of F1-score and accuracy on the Politifact, and by 2.2%, 3.9% on the Gossipcop, respectively. Interestingly, our model achieves these superior results without integrating social information, which FinerFact utilizes. Furthermore, our model surpasses KAN's performance on both datasets with F1-score and accuracy improvement of 7.9%, 8.1% and 11.9%, 9.5%. This is attributed to our model's ability to consider contextual information across multiple entities in the WikiAMR and link evidence between them, unlike KAN, which only focuses on the contextual information of a single entity. This enables our model to learn the facts between entities, benefiting from external knowledge.

TABLE II
ANALYSIS ON NUMBER OF HOPS LINKED BETWEEN ENTITIES.

	Total Entities	# 1 hop	# 2 hops	# 3 hops	# 4 hops	# 5 hops
Politifact	5438	951	10	5	3	2
Gossipcop	66346	5482	23	8	5	3

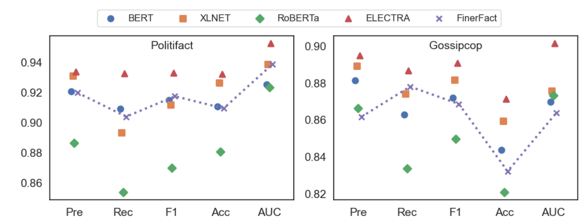

Fig. 4. Comparison on different language encoders.

VII. ABLATION STUDY

In this section, we have evaluated our methodology across a range of parameters and conducted experiments in various scenarios to understand the rationale behind our results. We have also carried out statistical tests to substantiate the validity of various measures. Additionally, we analyzed error cases within the model to highlight potential areas for improvement.

A. Comparison on different language encoders

We employ several transformer-based models to assess the effectiveness of EA^{2}N across various textual encodings. These include BERT-base-uncased [45], RoBERTa-base [46], XLNET-base [47], and ELECTRA-base [38]. The results of this evaluation, in conjunction with the baseline (FinerFact) result, are illustrated in Fig. 4. Notably, ELECTRA outperforms other models, exhibiting F1-score and accuracy of 0.9328, 0.9318 on the Politifact dataset and 0.8906, 0.8713 on the Gossipcop dataset. Comparing the remaining models, both XLNET and BERT demonstrate superior performance over RoBERTa. In comparison with baseline, this study concludes that by leveraging various textual encoders, EA^{2}N model surpasses other existing fake news detection models and shows comparable result with FinerFact. Although, ELECTRA language encoder surpasses all the other SOTA models, yielding substantial improvements in performance.

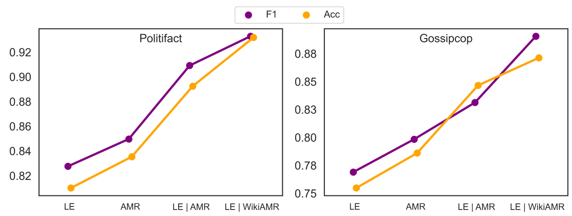

Fig. 5. Comparison on different EA^{2}N variants.

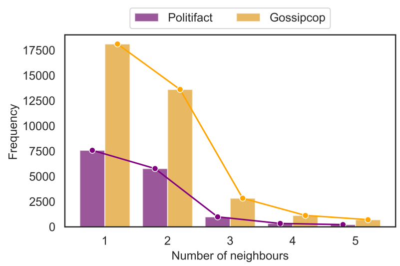

Fig. 6. Analysis on number of neighbours searched between entities.

B. Comparison on different EA^{2}N variants.

We conducted experiments with our model EA^{2}N (**LE**|**WikiAMR**) by incorporating different variations, including: 1) Only Language Encoder **(LE)** 2) Only AMR (**AMR**) and 3) Language Encoder with base AMR (**LE**|**AMR**). Our findings from Fig. 5 indicate that only AMR model performs better than only LE model. Moreover, when we combine both language encoder and AMR (**LE**|**AMR**), there is a significant improvement of 6-8% observed over the only LE and AMR models. Additionally, when we integrate evidence in AMR into our final model (**LE**|**WikiAMR**), there is a further enhancement of 3-4% over the **LE**|**AMR** model for both datasets. We conducted two-tailed t-tests and observed a significant difference between EA^{2}N variants, obtaining a significance score(p-value) < 0.01 thus rejecting the null hypothesis. The detailed analysis of t-tests is explained in Section VII-E.

C. Analysis on Evidence Linking Algorithm

In our proposed Evidence Linking Algorithm, we conducted an analysis of the number of hops linked and the number of neighbors searched between start and goal nodes. From Table II, It is observed that the majority of entities where we found connections from external knowledge in both datasets are linked using 1 hop, indicating direct relations between them. As the number of hops increased, the proportion of linked entities gradually decreased, with very few entities linked using 5 hops relations. Also, in Fig. 6, the results revealed that most entities are searched within the first neighbor, and the frequency gradually decreases for subsequent neighbors. Based on findings, we capped linked path at 5 nodes and explored up to 5 neighbors for each node between start and goal entities in Algo 1.

D. Effectiveness of WikiAMR

In order to investigate the influence of WikiAMR features on our model, we have conducted an examination of the attention

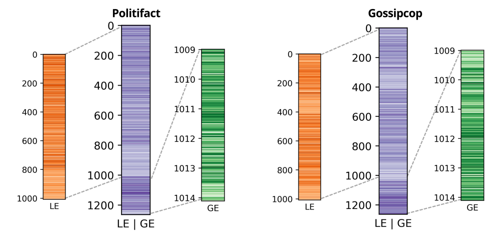

Fig. 7. Attention weight analysis over random samples from a) Politifact b) Gossicop.

weights from the final layer of the EA^{2}N. We delve into the attention weights of the Language Encoder (**LE**), the Graph Encoder (**GE**), as well as the combined Language Encoder with Graph Encoder (**LE**|**GE**). Analyzing the attention weights of **LE**|**GE** from Fig. 7, we deduce that the lower portion of the feature set (**GE** feature set) holds significant influence on the model's performance. This conclusion arises from the fact that our proposed WikiAMR encapsulates a comprehensive and intricate semantic structure of news articles. Furthermore, delving into the weights of individual encoders, we infer that within the **LE**, the initial feature subset strongly affects the model's behavior. This stems from the fact that the title of a news article, a concise summary of the news, is typically situated in the initial sentences. On the other hand, for **GE**, the entire feature set carries significance since WikiAMR emphasizes crucial semantic relationships among entities.

E. Two tailed t-tests on EA^{2}N variants

We conducted two-tailed t-tests to evaluate the significance of differences between the accuracy of EA^{2}N variants on randomly selected samples. The hypotheses were defined as follows:

- **Null Hypothesis ($\mathbf{H_0}$):** There is no significant difference between the accuracy of EA^{2}N variants on randomly selected samples.
- **Alternative Hypothesis ($\mathbf{H_1}$):** There is a significant difference between the accuracy of EA^{2}N variants on randomly selected samples.

The t-statistic was calculated using the formula:

$$t = \frac{\bar{x1} - \bar{x2}}{\sqrt{\frac{s1^2}{n1} + \frac{s2^2}{n2}}}$$

Here $s1$ and $s2$ are the standard deviation, $n1$ and $n2$ represent the samples considered, and $\bar{x1}$ and $\bar{x2}$ represent the mean accuracy for Model 1 and Model 2, respectively.

The t-statistic scores and p-values for all the models based on randomly selected 100 sample sets, grouped by dataset, are presented in Table III. Standard deviation for individual models such as LE, AMR, LE|AMR, LE|WikiAMR are 2.53, 3.21, 2.32, 2.71 for Politifact and 2.11, 3.07, 2.73, 2.35 for Gossipcop, respectively. It is evident from the table that the obtained significance values for the corresponding variants are less than 0.01, contradicting the null hypothesis. This statistical interpretation indicates a significant difference between the evaluated variants in both the Politifact and Gossipcop

TABLE III
T-TEST FOR EA^{2}N VARIANTS IN POLITIFACT AND GOSSIPCOP DATASETS.

Method	Politifact		Gossipcop	
	t-statistics	p-value	t-statistics	p-value
LE and AMR	6.19	3.39e-09	8.21	2.64e-14
AMR and LE|AMR	14.39	1.30e-32	14.46	7.96e-33
LE|AMR and LE|WikiAMR	12.40	1.63e-26	6.85	8.74e-11

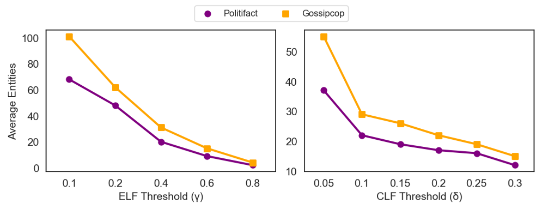

Fig. 8. Sensitivity analysis on γ and δ.

datasets. In other words, the improvement (Accuracy and F1-score) observed as LE $<$ AMR $<$ LE|AMR $<$ LE|WikiAMR shown in Fig. 5 is statistically significant.

F. Sensitivity analysis on γ and δ

We performed a sensitivity analysis to examine the influence of different ELF (Entity Level Filtering) and CLF (Context Level Filtering) thresholds on model's performance. The average entity values for ELF thresholds (γ) and CLF thresholds (δ), based on 100 samples, are displayed in Fig. 8.

The threshold values significantly impact system behavior. Higher thresholds lead to fewer entities, indicating stricter linking and classification criteria. When entities decrease, the graph tends to converge from WikiAMR to AMR, compromising accuracy as shown in statistical test. Conversely, a higher entity count results in a larger graph size, escalating training times. Optimizing threshold values involves striking a balance between accuracy and computational efficiency. The ELF threshold at 0.4 and the CLF threshold at 0.1 emerge as potential optimal values considering elbow method from Fig. 8. A γ of 0.4 maintains a balance between accurate and manageable linking of primary entities. Meanwhile, a δ of 0.1 helps control relevancy of the linked paths without compromising accuracy significantly.

G. Generalizability on different fake news domains

In order to evaluate the generalizability of our model, we conducted tests on the publicly available Snopes dataset [48], consisting of a total of 1703 fact-checking articles covering various topics science, health, politics, and urban legends sourced from the fact-checking website *snopes.com*. This dataset encompasses multiple classes, including false, true, mostly false, mostly true, scam, unknown, etc. Given that EA^{2}N is a fake/real classification model, we focused exclusively on the true and false classes. For the final assessment, we utilized 1106 articles classified as fake and 182 articles classified as true. To address class imbalance in the dataset, we employed standard NLP based data augmentation techniques. For this experiment, EA^{2}N is trained and tested on same dataset samples. The results presented in Table IV clearly demonstrate that our model EA^{2}N surpasses other models by a significant margin. We have compared our model EA^{2}N (**LE|WikiAMR**) with 1) GCAN 2) FakeFlow (**FF**) 3) Language Encoder (**LE**) 4) AMR (**AMR**) 5) Language Encoder with AMR (**LE|AMR**). It is evident form the results that when we integrate evidence in AMR into our final model (**LE|WikiAMR**), there is 2-5%, 1-6% gain in accuracy and F1- score respectively from all the other models.

TABLE IV
COMPARATIVE STUDY ON SNOPES DATASET.

Method	F1	Acc
GCAN	0.8015	0.7834
FF	0.8712	0.8542
LE	0.8620	0.8511
AMR	0.8961	0.8769
LE|AMR	0.9144	0.8869
LE|WikiAMR (EA^{2}N)	**0.9212**	**0.9045**

For cross-dataset generalization, we also compared our model with CompareNet [11], which incorporates external knowledge from Wikipedia through a directed heterogeneous document graph. For the comparison, we used two publicly available datasets, SLN [49] and LUN [50], and followed the 2-way classification setup described in the original paper [11]. In this setting, the model is first trained on LUN dataset samples and tested on SLN dataset samples. Our study shows that EA^{2}N (Micro F1: **0.9104**, Macro F1: **0.9088**) outperforms CompareNet (Micro F1: 0.8917, Macro F1: 0.8912) by a margin of 1-2% in both Micro and Macro F1 metrics. The rationale behind this is CompareNet extracts the first paragraph about an entity from Wikipedia and incorporates this into the final prediction. In contrast, our model selectively integrates only the most relevant evidence found on Wikipedia, which influences the final prediction.

VIII. CONCLUSION AND OUTLOOK

In this study, we introduce EA^{2}N, a novel Evidence-based AMR Attention Network designed to effectively identify fake news by encapsulating external knowledge within the proposed WikiAMR graph through an evidence linking algorithm. In addition to utilizing language features, our model shows the capablity of learning intricate semantic relations, including events, time, location, etc., by connecting evidences across entities. Effectiveness of our approach is demonstrated with extensive experiments including detailed ablation study, which showcased the state-of-the-art performance on two real-world datasets for fake news detection.

It is important to note that our study not only provide solution for fake news detection but it has the potential to pave the way for solving various other NLP applications. For example, the use of proposed WikiAMR is not limited to fake news detection only. One may use WikiAMR wherever external knowledge enhanced AMR is required, such as Law Enforcement, Insurance Validation, etc. In such cases, the choice of knowledge graph (in our case Wikipedia) should be based on the problem domain in hand. Further, the relatedness objective functions of evidence linking algorithm needs to modified accordingly.

It is evident from this study as well as from the other literature that external knowledge is imperative in veracity problems. Wikipedia is one of the way to incorporate external knowledge. Different other ways such as social information with semantic relations, live genuine channels can be explored for further improvement. Such information can also be part of WikiAMR graph. Further, one may use knowledge graph from live genuine channels to deal with the slow update issue with Wikipedia knowledge graph.

REFERENCES

[1] P. R. Brewer, D. G. Young, and M. Morreale, "The Impact of Real News about "Fake News": Intertextual Processes and Political Satire," *International Journal of Public Opinion Research*, vol. 25, no. 3, pp. 323–343, 09 2013.

[2] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[3] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in *ACL (Volume 2: Short Papers)*. Jeju Island, Korea: ACL, Jul. 2012, pp. 171–175.

[4] J. Ma, W. Gao, P. Mitra, S. Kwon, B. J. Jansen, K.-F. Wong, and M. Cha, "Detecting rumors from microblogs with recurrent neural networks," in *IJCAI*, ser. IJCAI'16. AAAI Press, 2016, p. 3818–3824.

[5] Y. Long, Q. Lu, R. Xiang, M. Li, and C.-R. Huang, "Fake news detection through multi-perspective speaker profiles," in *IJCNLP (Volume 2: Short Papers)*. Taipei, Taiwan: Asian Federation of Natural Language Processing, Nov. 2017, pp. 252–256.

[6] Y. Liu and Y.-F. Wu, "Early detection of fake news on social media through propagation path classification with recurrent and convolutional networks," *AAAI*, vol. 32, no. 1, Apr. 2018.

[7] K. Popat, S. Mukherjee, J. Strötgen, and G. Weikum, "Where the truth lies: Explaining the credibility of emerging claims on the web and social media," ser. WWW '17 Companion. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2017, p. 1003–1012.

[8] K. Popat, S. Mukherjee, A. Yates, and G. Weikum, "DeClarE: Debunking fake news and false claims using evidence-aware deep learning," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, E. Riloff, D. Chiang, J. Hockenmaier, and J. Tsujii, Eds. Brussels, Belgium: Association for Computational Linguistics, Oct.-Nov. 2018, pp. 22–32.

[9] Y. Dun, K. Tu, C. Chen, C. Hou, and X. Yuan, "Kan: Knowledge-aware attention network for fake news detection," *AAAI*, vol. 35, no. 1, pp. 81–89, May 2021.

[10] Y. Jin, X. Wang, R. Yang, Y. Sun, W. Wang, H. Liao, and X. Xie, "Towards fine-grained reasoning for fake news detection," *AAAI*, vol. 36, pp. 5746–5754, 06 2022.

[11] L. Hu, T. Yang, L. Zhang, W. Zhong, D. Tang, C. Shi, N. Duan, and M. Zhou, "Compare to the knowledge: Graph neural fake news detection with external knowledge," in *ACL-IJCNLP (Volume 1: Long Papers)*. Online: ACL, Aug. 2021, pp. 754–763.

[12] S.-h. Yang, C.-c. Chen, H.-H. Huang, and H.-H. Chen, "Entity-aware dual co-attention network for fake news detection," in *EACL*. Dubrovnik, Croatia: ACL, May 2023, pp. 106–113.

[13] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, and J. Tang, "Kepler: A unified model for knowledge embedding and pre-trained language representation," *Transactions of the ACL*, vol. 9, pp. 176–194, 2021.

[14] L. Banarescu, C. Bonial, S. Cai, M. Georgescu, K. Griffitt, U. Hermjakob, K. Knight, P. Koehn, M. Palmer, and N. Schneider, "Abstract Meaning Representation for sembanking," in *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, Sofia, Bulgaria, Aug. 2013, pp. 178–186.

[15] D. Cai and W. Lam, "Graph transformer for graph-to-sequence learning," in *AAAI*. AAAI Press, 2020, pp. 7464–7471.

[16] F. Qian, C. Gong, K. Sharma, and Y. Liu, "Neural user response generator: Fake news detection with collective user intelligence," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 3834–3840.

[17] H. T. Nguyen, A. Cano, T. Vu, and T. N. Dinh, "Blocking self-avoiding walks stops cyber-epidemics: A scalable gpu-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 7, pp. 1263–1275, 2020.

[18] B. Ghanem, S. P. Ponzetto, P. Rosso, and F. Rangel, "Fakeflow: Fake news detection by modeling the flow of affective information," in *16th EACL*, 2021.

[19] L. Wei, D. Hu, W. Zhou, Z. Yue, and S. Hu, "Towards propagation uncertainty: Edge-enhanced Bayesian graph convolutional networks for rumor detection," in *ACL-IJCNLP (Volume 1: Long Papers)*. Online: ACL, Aug. 2021, pp. 3845–3854.

[20] L. Azevedo, M. d'Aquin, B. Davis, and M. Zarrouk, "LUX (linguistic aspects under eXamination): Discourse analysis for automatic fake news classification," in *ACL-IJCNLP*. Online: ACL, Aug. 2021, pp. 41–56.

[21] L. Fang, K. Feng, K. Zhao, A. Hu, and T. Li, "Unsupervised rumor detection based on propagation tree vae," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 10, pp. 10 309–10 323, 2023.

[22] F. Monti, F. Frasca, D. Eynard, D. Mannion, and M. M. Bronstein, "Fake news detection on social media using geometric deep learning." ICLR2019, 2019.

[23] K. Shu, X. Zhou, S. Wang, R. Zafarani, and H. Liu, "The role of user profiles for fake news detection," in *IEEE/ACM ASONAM*, ser. ASONAM '19. New York, NY, USA: Association for Computing Machinery, 2020, p. 436–439.

[24] Y.-J. Lu and C.-T. Li, "GCAN: Graph-aware co-attention networks for explainable fake news detection on social media," in *ACL*. Online: ACL, Jul. 2020, pp. 505–514.

[25] H. Yao, Y.-x. Wu, M. Al-Shedivat, and E. Xing, "Knowledge-aware meta-learning for low-resource text classification," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 1814–1821.

[26] W. Xu, J. Wu, Q. Liu, S. Wu, and L. Wang, "Evidence-aware fake news detection with graph neural networks," in *ACM Web Conference 2022*. New York: ACM, 2022, p. 2501–2510.

[27] J. Wu, W. Xu, Q. Liu, S. Wu, and L. Wang, "Adversarial contrastive learning for evidence-aware fake news detection with graph neural networks," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2023.

[28] M. Sun, X. Zhang, J. Ma, S. Xie, Y. Liu, and P. S. Yu, "Inconsistent matters: A knowledge-guided dual-consistency network for multi-modal rumor detection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 736–12 749, 2023.

[29] Q. Liu, J. Wu, S. Wu, and L. Wang, "Out-of-distribution evidence-aware fake news detection via dual adversarial debiasing," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–13, 2024.

[30] P. Kouris, G. Alexandridis, and A. Stafylopatis, "Text summarization based on semantic graphs: An abstract meaning representation graph-to-text deep learning approach," 2022.

[31] Z. Wang, X. Wang, X. Han, Y. Lin, L. Hou, Z. Liu, P. Li, J. Li, and J. Zhou, "CLEVE: Contrastive Pre-training for Event Extraction," in *ACL-IJCNLP (Volume 1: Long Papers)*. Online: ACL, Aug. 2021, pp. 6283–6297.

[32] J. Lim, D. Oh, Y. Jang, K. Yang, and H. Lim, "I know what you asked: Graph path learning using AMR for commonsense reasoning," in *ICCL*. Barcelona, Spa(Online): International Committee on Computational Linguistics, Dec. 2020, pp. 2459–2471.

[33] Y. Zhang, L. Trinh, D. Cao, Z. Cui, and Y. Liu, "Detecting out-of-context multimodal misinformation with interpretable neural-symbolic model," 2023.

[34] S. Gupta, N. Yadav, S. Kundu, and S. Sankepally, "Fakedamr: Fake news detection using abstract meaning representation network," in *International Conference on Complex Networks and Their Applications*. Springer, 2023, pp. 308–319.

[35] S. Zhang, X. Ma, K. Duh, and B. Van Durme, "AMR parsing as sequence-to-graph transduction," in *ACL*. Florence, Italy: ACL, Jul. 2019, pp. 80–94.

[36] P. Ferragina and U. Scaiella, "Tagme: On-the-fly annotation of short text fragments (by wikipedia entities)," in *ICIKM*, ser. CIKM '10. New York, NY, USA: ACM, 2010, p. 1625–1628.

[37] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *EMNLP*. Doha, Qatar: ACL, Oct. 2014, pp. 1724–1734.

[38] K. Clark, M. Luong, Q. V. Le, and C. D. Manning, "ELECTRA: pre-training text encoders as discriminators rather than generators," in *ICLR*. OpenReview.net, 2020.

[39] K. Shu, D. Mahudeswaran, S. Wang, D. Lee, and H. Liu, "Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media," *Big Data*, vol. 8, no. 3, pp. 171–188, 2020.
[40] F. Yang, Y. Liu, X. Yu, and M. Yang, "Automatic detection of rumor on sina weibo," in *SIGKDD*, ser. MDS '12. New York, NY, USA: ACM, 2012.
[41] C. Castillo, M. Mendoza, and B. Poblete, "Information credibility on twitter," in *WWW*, ser. WWW '11. New York, NY, USA: ACM, 2011, p. 675–684.
[42] S. Kwon, M. Cha, W. Jung, Kyomand Chen, and Y. Wang, "Prominent features of rumor propagation online social media," in *2013 IEEE ICDM*, 2013, pp. 1103–1108.
[43] J. Z. Pan, S. Pavlova, C. Li, N. Li, Y. Li, and J. Liu, "Content based fake news detection using knowledge graphs," in *ISWC*. Cham: Springer International Publishing, 2018, pp. 669–683.
[44] H. Wang, F. Zhang, X. Xie, and M. Guo, "Dkn: Deep knowledge-aware network for news recommendation," in *WWW*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1835–1844.
[45] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: ACL, Jun. 2019, pp. 4171–4186.
[46] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.
[47] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le, *XLNet: Generalized Autoregressive Pretraining for Language Understanding*. Red Hook, NY, USA: Curran Associates Inc., 2019.
[48] N. Vo and K. Lee, "Where are the facts? searching for fact-checked information to alleviate the spread of fake news," in *In EMNLP 2020*. Online: ACL, Nov. 2020, pp. 7717–7731.
[49] V. Rubin, N. Conroy, Y. Chen, and S. Cornwell, "Fake news or truth? using satirical cues to detect potentially misleading news," in *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. San Diego, California: ACL, Jun. 2016, pp. 7–17.
[50] H. Rashkin, E. Choi, J. Jang, S. Volkova, and Y. Choi, "Truth of varying shades: Analyzing language fake news and political fact-checking," in *EMNLP*. Copenhagen, Denmark: ACL, Sep. 2017, pp. 2931–2937.
[51] I. Loshchilov and F. Hutter, "SGDR: Stochastic gradient descent with warm restarts," in *ICLR*, 2017.
[52] L. N. Smith and N. Topin, "Super-convergence: Very fast training of residual networks using large learning rates," 2018.

APPENDIX

A. Experimental Setup

1) Dataset: Details of the datasets are provided in Table V. Basic prepossessing steps such as removal of '@', '#' symbols, removal of website URLs etc., are performed on the news content.

TABLE V
DISTRIBUTION OF DATA FOR A) POLITIFACT, B) GOSSIPCOP.

	# True	# Fake	# Total
PolitiFact	443	372	815
GossipCop	4219	3393	7612

2) Implementation Details: To perform Entity and Context Level Filtering (ELF and CLF) for linking external knowledge, we utilize the $TagMe$ API [36]. In order to generate the AMR graph, we have used pretrained STOG model [35]. For filtering at the entity and context levels, we set the values of γ and δ to 0.4 and 0.1, respectively. These values assume that primary entities in the AMR should exhibit high relatedness, while their neighbors will have comparatively lower relatedness. During the evidence linking process, we limit the linked path to a maximum of 5 hops (n_m) including start and goal nodes and explored up to 5 neighbors for each node between the primary entities. This limitation is kept to enforce time and memory constraint on our algorithm because of large number of entities present in the dataset. Further, the ablation study supports our choice of 5 for both the cases. We employ transformer encoding with an embedding size of 512. Additionally, the affective lexical features are set to an embedding size of 240 using 10 segments in a news article. To facilitate the learning rate annealing, we adopt the cosine learning rate technique [51]. The parameters used for the graph path learning model are identical to those described in Cai's model [15]. The training of our model was conducted on an NVIDIA A30 GPU with 24 GB of memory.

B. Supplementary Experiments

1) Effects of different learning rates and batch sizes on EA^2N: This section investigates the effects of different learning rates (LR) and batch sizes on the EA^2N model's performance, specifically focusing on accuracy and F1 score across the Politifact and Gossipcop datasets. For the learning rate evaluation, we compare three scheduling strategies: **Linear**, **OneCycleLR** [52], and **Cosine** [51]. In parallel, we analyze batch sizes of 1, 2, 4, and 8 to understand how batch variation impacts model outcomes.

TABLE VI
ANALYSIS ON DIFFERENT LEARNING RATE STRATEGIES.

Learning rate	Poltifact		Gossipcop	
	F1	Acc	F1	Acc
Linear	0.8869	0.8748	0.8571	0.8549
OneCycleLR	0.9112	0.9087	0.8726	0.8636
Cosine	0.9328	0.9318	0.8906	0.8713

Table VI reveals distinct performance trends associated with each LR schedule and batch size. The **Cosine LR** schedule consistently outperforms other strategies, achieving an F1 score and accuracy of 0.9328 and 0.9318 on the Politifact dataset, and 0.8906 and 0.8713 on the Gossipcop dataset. Despite the superior performance of **Cosine LR**, the **OneCycleLR** strategy exhibits the ability to reach saturation levels more swiftly in fewer epochs due to its dynamic LR scheduling. This introduces a delicate trade-off between the time efficiency of **OneCycleLR** and the enhanced performance of **Cosine LR**.

TABLE VII
ANALYSIS ON DIFFERENT BATCH SIZES.

Batch size	Poltifact		Gossipcop	
	F1	Acc	F1	Acc
1	0.8807	0.8591	0.8532	0.8471
2	0.8974	0.8711	0.8609	0.8581
4	0.9178	0.9228	0.8743	0.8685
8	0.9328	0.9318	0.8906	0.8713

Regarding batch size in Table VII, larger batch sizes yield better results, with the performance metrics higher at a batch size of 8. However, the incremental improvement diminishes beyond batch size 4, indicating a nuanced balance between batch size and model training efficacy.

TABLE VIII
COMPARATIVE STUDY OF EA^{2}N W.R.T. SOTA MODELS USING DIFFERENT TRANSFORMER.

Encoder	Method	PolitiFact					GossipCop				
		Pre	Rec	F1	Acc	AUC	Pre	Rec	F1	Acc	AUC
ELECTRA	FinerFact	0.9211	0.9148	0.9179	0.9124	0.9433	0.8687	0.8793	0.8747	0.8416	0.8723
	EA^{2}N	**0.9333**	**0.9324**	**0.9328**	**0.9318**	**0.9523**	**0.8947**	**0.8865**	**0.8906**	**0.8713**	**0.9014**
BERT	FinerFact (cased)	0.9196	0.9037	0.9172	0.9092	0.9384	0.8615	0.8779	0.8685	0.8320	0.8637
	EA^{2}N (uncased)	0.9203	0.9089	0.9167	0.9104	0.9250	0.8813	0.8694	0.8719	0.8435	0.8695
	EA^{2}N (cased)	**0.9295**	**0.9174**	**0.9211**	**0.9192**	**0.9397**	**0.8892**	**0.8794**	**0.8804**	**0.8798**	**0.8857**
1-Layer Transformer	FinerFact	0.9091	0.9182	0.8810	0.8864	0.9234	0.8579	**0.8874**	0.8602	0.8310	0.8600
	EA^{2}N	**0.9117**	**0.9203**	**0.9159**	**0.8973**	**0.9346**	**0.8638**	0.8831	**0.8733**	**0.8435**	**0.8719**

2) Assessment on Language Encoder Bias: In order to assess whether our model exhibits bias towards the language encoder, we conducted an ablation study focusing on the language encoder. We have compared our model with FinerFact on ELECTRA, BERT, and 1-Layer Transformer encoders. When both models are configured under identical settings, EA^{2}N surpasses FinerFact in benchmark performance. Table VIII shows less than a 1% improvement in FinerFact when switching to ELECTRA from BERT. A similar pattern is also evident in EA^{2}N when comparing the cased BERT and ELECTRA model. With this study, we can conclude that EA^{2}N outperforms FinerFact in all the cases.

C. Error Case Analysis

```
(snt1 (l / lead-03
      :ARG0 (o / order-01
         :mod (e / executive))
      :ARG2 (c / suspect-01
         :ARG1 (p / person
            :name (n / name
               :op1 "Rasheed"
               :op2 "Muhammad"))
            :ARG0-of (l2 / lead-02
               :ARG1 (c2 / criminal-organization
                  :name (n2 / name
                     :op1 "Islamic"
                     :op2 "State"))))))
   ...
   ...
   ...
      :op5 (p9 / person
         :name (n9 / name
            :op1 "Rasheed"
            :op2 "Muhammad"))
         :ARG2-of (s2 / suspect-01
            :ARG1 (t4 / terror))))
```

Case Study: *"Executive Order Leads to Capture of ISIS Leader Rasheed Muhammad Search tags .... apologizes rasheed muhammad .... **terror suspect rasheed .... was captured**"*.

The sentence in focus is: "Executive Order Leads to Capture of ISIS Leader Rasheed Muhammad..." Corresponding to this, an Abstract Meaning Representation (AMR) graph is constructed to represent the semantic structure. However, this particular AMR graph exhibits some limitations in its representation.

1) Limitation in AMR Graph Construction: The limitation arises from an incomplete AMR graph construction. Specifically, in the highlighted part of the article and towards the end of the graph, only the 'suspect' part is reflected, omitting the 'captured' aspect. This incompleteness affects the salient features of the graph and consequently impacts the accuracy of our model. It hinders a holistic understanding of the event, overlooking crucial actions like 'capture', which are vital for accurate predictions. To overcome this limitation, the model can be trained on domain data or subgraphs of segments can be created in an article which can be merged later in one graph.

2) Inconsistent relevance scores in $TagMe$: Furthermore, another aspect of concern is the inconsistent relevance scores in $TagMe$, an entity linking tool. These scores play a vital role in the Entity-Level Filtering (ELF) and Context-Level Filtering (CLF) stages of our model. When the relevance scores are inconsistent, they affect the path determination process in both the stages. For instance, relevance scores for certain prominent entities are not considered if they are below the thresholds, which were initially tuned for them. This alteration in the path can significantly impact the accuracy and reliability of the model's predictions.

Shubham Gupta is currently working toward the PhD degree with the Department of Computer Science at Indian Institute of Technology Jodhpur, India. He is also an active IEEE student member. His research interests include natural language processing, social network analysis, deep learning.

Abhishek Rajora is a Software Developer and an active researcher at SoNAA (Social Network Analysis and Application) in Indian Institute of Technology Jodhpur, contributing to academia. His research interests include Deep Learning, Computer Vision, Natural Language Processing and Multimodal Learning.

Suman Kundu is an Assistant Professor in the Department of Computer Science and Engineering at Indian Institute of Technology Jodhpur. He leads the research group SoNAA: Social Network Analysis and Application in the department. Dr. Kundu received his Ph.D. degree from Jadavpur University in 2017. He visited Engine group at Wroclaw University of Science and Technology in 2018-2019 for his postdoctoral research. Dr. Kundu has more than 6 years of industrial software development experience with Zinfi Software Systems Pvt. Ltd., Kolkata. He published several articles in the area of social network analysis, network science, granular computing, soft computing and streaming graph algorithms. His research interests includes social network analysis, network science, streaming algorithms, big data, fuzzy and rough set, and granular computing.