Information Sciences 742 (2026) 123293

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

EZ-check: Explainable zero-shot knowledge extraction and fact checking using knowledge graphs

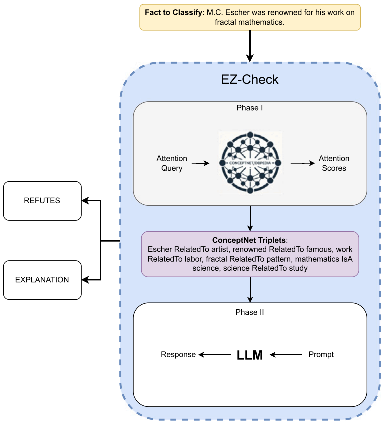
Akhil Chaudhary [a,*], Somayeh Kafaie [b], Enayat Rajabi [a]

[a] Shannon School of Business, Cape Breton University, Grand Lake Dr., Sydney, B1M 1A2, NS, Canada
[b] Department of Mathematics and Computing Science, Saint Mary's University, 912 Robie St, Halifax, B3H 3C3, NS, Canada

HIGHLIGHTS

- Novel vector-attention query enables fast and accurate knowledge graph extraction.
- EZ-Check achieves fact-checking without reliance on curated external evidence.
- Achieves state-of-the-art results on FEVER and UKP Snopes.

GRAPHICAL ABSTRACT


Fact to Classify: M.C. Escher was renowned for his work on fractal mathematics.
EZ-Check
Phase I
Attention Query
CONCEPTNET/DBPEDIA
Attention Scores
ConceptNet Triplets:
Escher RelatedTo artist, renowned RelatedTo famous, work RelatedTo labor, fractal RelatedTo pattern, mathematics IsA science, science RelatedTo study
Phase II
Response
LLM
Prompt
REFUTES
EXPLANATION

ARTICLE INFO

Keywords:
Knowledge graph
Language models
LLM
Fact-checking
Explainability
Vector attention
ConceptNet
Information extraction
Knowledge extraction
Information retrieval
Vector semantic calculation

ABSTRACT

The unchecked proliferation of online information underscores the need for transparent and efficient fact-checking systems. While some methods leverage structured representations like knowledge graphs, significant gaps persist in efficiently and accurately extracting relevant information. To address this, we introduce EZ-Check, a novel framework that combines explainability with a fast, reliable, and effective querying mechanism for knowledge graphs, leveraging resources such as ConceptNet.

At its core, EZ-Check integrates structured knowledge graph relationships with advanced language models to assess textual authenticity and extract relevant information precisely. EZ-Check achieves more than a 10% relative F1 improvement on FEVER and UKP Snopes compared to representative claim-only and evidence-based baselines, including BERT-as-KB, GPT-3.5 Turbo, KGAT, and TARSA, while reducing inference time by approximately 50%. On fact-checking datasets such

[*] Corresponding author.
Email addresses: akhil_chaudhary@cbu.ca (A. Chaudhary), somayeh.kafaie@smu.ca (S. Kafaie), enayat_rajabi@cbu.ca (E. Rajabi).

https://doi.org/10.1016/j.ins.2026.123293
Received 1 September 2025; Received in revised form 23 February 2026; Accepted 23 February 2026
Available online 26 February 2026

as FEVER and UKP Snopes, it achieves state-of-the-art results, surpassing baselines without relying on external evidence sentences. Moreover, EZ-Check delivers clear, interpretable justifications, setting a new standard for both performance and explainability in fact verification based on our testing against baselines.

1. Introduction

The spread of misinformation on the internet has become a critical issue. With the rapid growth in data generation and information retrieval techniques, vast amounts of data are constantly being produced and consumed. Among this data, a portion consists of false information and misinformation, such as fake news and propaganda, which can influence public opinion, disrupt financial markets, and even alter election outcomes [1]. Assessing the credibility of information is essential in various areas of natural language processing, including tasks like language comprehension, knowledge graph completion, and open-domain question answering. As recent analyses show, generative large language models introduce new opportunities and risks in automated fact-checking pipelines, particularly in zero-shot scenarios and hallucination control [2].

Due to the massive influx of data, there is an urgent need for automated methods to evaluate the scores of claims. Creating reliable automated fact-checking systems is still a significant hurdle despite its critical importance. Numerous methodologies have been crafted to address the complexities of automated fact-checking [3]. These methods usually involve document retrieval, evidence selection, and claim validation. However, most solutions only affirm or refute a claim without offering a rationale. Explanations are essential: (i) to help laypeople understand domain-specific claims, and (ii) to provide perspectives that improve fact-checking effectiveness. While sites like PolitiFact[1] and Snopes[2] provide such explanations, they rely on costly, labor-intensive human verification.

Most current fact-checking approaches rely on evidence sentences, often retrieved from sources like Wikipedia, Wikidata, Snopes, or PolitiFact. Typically framed as Recognizing Textual Entailment (RTE) [4] or Natural Language Inference (NLI) [5], they connect evidence via concatenation, selection, or re-ranking [6]. Such methods lack sufficient relational and logical context [7], limiting accuracy when explanatory information is missing. Recent work using graph neural networks [8] improves consolidation but remains shallow, modeling only at the word or sentence level [9], and failing to capture complex multi-granular relationships among words, facts, and sentences.

In explainable fact-checking, the task is to verify whether text supports or contradicts a claim [10]. Unlike prior methods, we generate concise explanations without relying on pre-identified evidence sentences. Evidence-based approaches [11] achieve strong results but often suffer from long inference times and dependence on stored information, which may not directly clarify a claim's factual basis.

Previous approaches to fact verification have drawn on broad ontologies like DBPedia [12], formal logic rules [13], and semi-structured data sources [14]. However, they face three main issues: (i) reliance on pre-stored evidence sentences, (ii) generation of overly detailed and cluttered explanations, and (iii) limited interpretability for non-experts.

Our work addresses these challenges by: (i) introducing a vector-based querying method to eliminate dependence on pre-stored evidence, (ii) employing attention-driven triplet selection to produce concise and coherent explanations, and (iii) designing EZ-Check to provide human-understandable outputs that balance accuracy with interpretability.

To address the above challenges, we turn to knowledge extraction from structured sources such as knowledge graphs. Resources like ConceptNet [15] and DBPedia [12] have advanced significantly in recent years and are critical for fact verification, as they reduce reliance on costly, manually curated information.

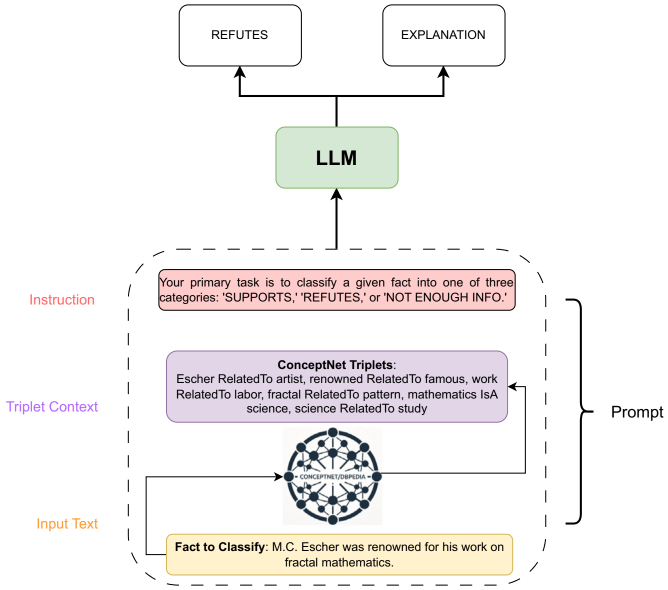
Within this direction, our focus is on approaches that enable querying knowledge graphs using natural language or directly processing text inputs without complex syntax. Such methods improve usability with large language models and make fact verification more accessible to non-technical users. A common strategy is to use seed terms (e.g., TF-IDF [16]) to extract subgraphs, which are then expanded by including nearby nodes within a fixed edge distance [17]. Other techniques restrict path lengths for efficiency and relevance, constructing graphs by linking triples through shared entities [18].

Alternatively, structured query languages (SQLs) such as Cypher and SPARQL allow precise querying but demand expertise in syntax. These methods face limitations: reliance on heuristics and thresholds can exclude relevant nodes, while fixed edge distances miss deeper relations [19]. Moreover, dependence on structured syntax discourages broader adoption. A comprehensive review of modern knowledge-graph reasoning methods underlines ongoing issues with scalability, semantic drift, and the challenge of integrating KG reasoning with downstream NLP tasks.

Our objective is to design a lightweight framework that leverages knowledge graphs such as ConceptNet [15] and DBPedia [12], in combination with language models, for fact verification. Prior work shows that appropriate prompting with relevant data can improve model performance while reducing the need for extensive fine-tuning [20]. Building on this insight, the proposed *EZ-Check* framework operates without fine-tuning or additional training, using datasets solely for evaluation, with ConceptNet as its only external resource.

[1] https://www.politifact.com.

[2] https://www.snopes.com.



Your primary task is to classify a given fact into one of three categories: 'SUPPORTS,' 'REFUTES,' or 'NOT ENOUGH INFO.'

ConceptNet Triplets: Escher RelatedTo artist, renowned RelatedTo famous, work RelatedTo labor, fractal RelatedTo pattern, mathematics IsA science, science RelatedTo study

Fact to Classify: M.C. Escher was renowned for his work on fractal mathematics.

Fig. 1. This figure overviews our approach and shows the workflow of EZ-Check. The language model outputs with a classification —either "SUPPORTS," "REFUTES," or "NOT ENOUGH INFO"—along with a detailed explanation generated to support the decision.

The proposed *EZ-Check* framework has two goals:

- Provide a fast and reliable natural language querying method for knowledge graphs.
- Enable zero-shot fact verification using structured graph knowledge without fine-tuning or additional training.

EZ-Check makes knowledge graphs accessible to non-technical users through intuitive querying and zero-shot capabilities, improving usability and efficiency. Unlike methods dependent on evidence sentences or large-scale traversals, it directly leverages structured graph knowledge to reduce complexity and misinterpretation. As shown in Fig. 1, the framework employs a vector-based querying method inspired by attention to retrieve relevant triplets for each claim, which a language model then interprets to verify the claim and generate concise natural language explanations. A custom prompting system structures this process through a chain-of-thought approach that integrates the claim's content, model knowledge, structured commonsense from the graph, and auxiliary signals such as sentiment and stance [21].

To the best of our knowledge, *EZ-Check* is the first framework to provide textual explanations for fact verification without relying on pre-supplied evidence sentences. It offers a novel method for clarifying the factual nature of claims through structured knowledge and contextual reasoning.

Our main contributions are:

- Proposing a novel vector-based semantic querying method for fast and natural language-driven knowledge extraction from Knowledge Graphs, eliminating the need for structured queries.
- Introducing EZ-Check, an explainable fact-checking framework that integrates structured knowledge from graphs and unstructured knowledge from language models, providing clear and human-comprehensible explanations via an attention-based query mechanism.
- Enabling zero-shot fact-checking without reliance on extractive evidence gathering or complex query formulation, by leveraging knowledge graphs with the proposed vector-attention-based querying technique.
- Demonstrating EZ-Check's effectiveness through extensive evaluations, showing competitive or superior performance compared to state-of-the-art methods.

2. Related work

Fact-checking is essential for verifying the accuracy of information. In its initial conceptualization, this task was framed within the context of Natural Language Inference (NLI) [5]. Natural Language Inference (NLI), a subfield of NLP, identifies logical relationships between sentences. In fact-checking, NLI evaluates whether a statement aligns with or contradicts known facts, enabling automated assessment of claim truthfulness and supporting accurate, reliable information. Early NLI-based approaches [7] primarily relied on straightforward techniques to combine evidence. For example, they might concatenate pieces of evidence or handle them individually. However, these approaches fall short of capturing the relationships among multiple interconnected pieces of evidence.

Subsequent works [19] moved toward using graph-based reasoning to capture the intricate logic involved in fact-checking. Models like GEAR [19] and KGAT [9] form graphs where each node represents a piece of evidence at the sentence level. These models employ deep graph attention networks to propagate information between connected nodes. TARSA [7] creates a fully connected evidence graph and incorporates topic-aware reasoning to improve fact verification.

Need for explainability in fact-checking. Fact-checking has shifted from merely verifying truthfulness to also providing rationales, which is critical in domains such as health, law, journalism, and public policy [11]. Approaches include NLI [22], reasoning from relational tables, textual entailment [23], and fake news detection [24]. The demand for interpretability is further underscored in the LLM era by work on factuality and fact-checking opportunities [25]. Recent work highlights that proper calibration and oversight are essential, as LLM-based fact-checkers may exhibit overconfidence or inconsistent judgment across similar claims.

FACE-KEG [11] builds a knowledge graph from a structured base for fact-checking, with nodes trained from scratch like TARSA. However, it is limited to phrase- or sentence-level semantics, restricting capture of complex, higher-order relationships. Both structure and semantics remain crucial for reasoning with knowledge graphs.

Limitations of existing models. While current models target explainability, they typically rely on pre-defined evidence sentences [6]. This dependence hinders the handling of real-time or sparsely documented information and makes evidence gathering labor-intensive. Moreover, with misinformation evolving through generative models, recent reviews highlight the need for adaptive systems that can track and explain such patterns in dynamic environments.

Advancements beyond pre-existing evidence. Recent work explores unstructured data and commonsense reasoning [26] to avoid predefined evidence sentences [27]. While BERT-based models can verify facts [28], they struggle in nuanced cases requiring richer context and often lack clear explanations. Progress has been made with models that jointly classify truthfulness and generate evidence, improving both accuracy and transparency.

Advances in knowledge graph extraction. Advancements in the field have led to innovative techniques for extracting domain-focused subgraphs from expansive knowledge graphs like ConceptNet, which are critical for tasks requiring domain-specific knowledge representation.

One technique identifies subgraphs using seed terms derived from TF-IDF [16] analysis of domain text or cleaned input. These top-ranked noun phrases are used to query ConceptNet, producing a subgraph that is expanded by including nearby nodes, typically within two edges for focus and relevance [17].

Another approach links question-related entities to answers within ConceptNet [18]. It restricts searches to paths under three steps for efficiency, then builds a graph where nodes represent triples and edges connect shared entities, capturing concept relationships.

These methods face limitations: reliance on heuristics and thresholds can exclude relevant nodes, and fixed edge or step constraints may miss deeper relationships, thereby reducing the context of extracted knowledge [29].

Given these challenges, there is a pressing need for new models in fact-checking that offer both explainability and independence from pre-existing evidence. To address these needs, we introduce an automated fact-checking system that is flexible and has real-world applicability. Our proposed system utilizes common-sense knowledge from ConceptNet. It uses a language model to understand facts and their relationships with common sense knowledge, and it verifies facts without needing evidence, sentences, or inference from external knowledge sources. Recent surveys have examined trends at the intersection of misinformation, generative AI, and societal impact, providing a timely perspective on the direction fact-checking research must take. Our approach aligns with these findings by focusing on both explainability and adaptability in the presence of rapidly evolving information ecosystems.

2.1. Distinction from graph-based attention models

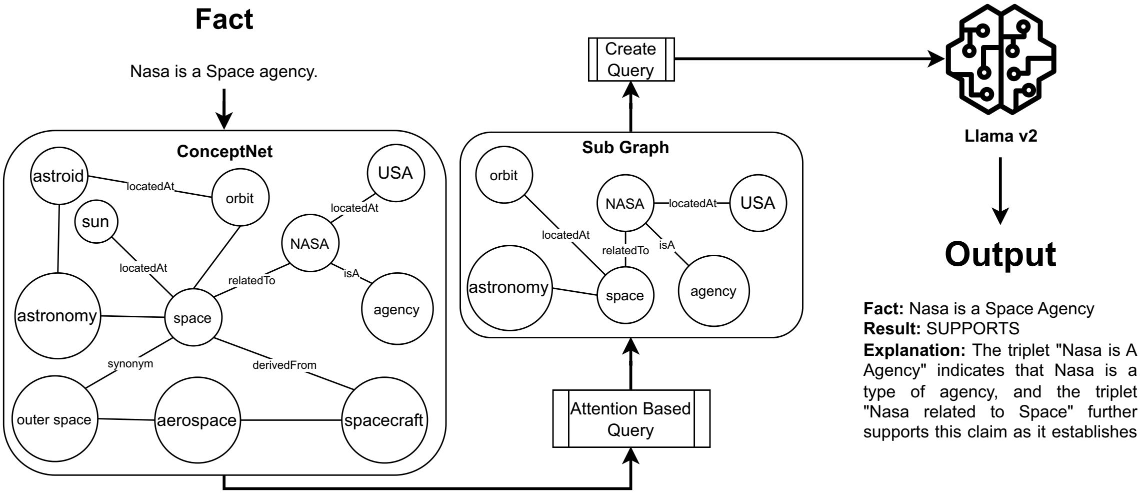
To clarify the contribution of our approach relative to prior graph-attention frameworks, we briefly contrast EZ-Check with models such as KGAT [9] and GEAR [19]. These methods learn graph attention weights during supervised training, propagating information across entities and relations to optimize node-level or statement-level predictions. Their attention mechanisms are parameterized, trained end-to-end, and rely on pre-constructed subgraphs.

2.2. Retrieval-augmented generation and hybrid LLM–KG fact-checking

Retrieval-Augmented Generation (RAG) has become a dominant paradigm in LLM-based fact-checking, where external textual evidence (e.g., Wikipedia passages) is retrieved and provided as context for claim verification. While RAG-based systems can improve factual grounding, they remain susceptible to evidence hallucination and opaque reasoning when retrieved text is incomplete or noisy. In contrast, EZ-Check does not retrieve or condition on textual evidence. Instead, it performs query-time reasoning directly over structured knowledge graph triplets using vector attention, enabling transparent, symbolically grounded explanations without relying on external text retrieval [30].

In fact, EZ-Check performs *query-time vector attention* using fixed ConceptNet embeddings. No parameters are learned, no graph propagation occurs, and the relevant subgraph is constructed dynamically for each claim. Instead of training a model to discover attention weights, EZ-Check computes semantic relevance directly by comparing token vectors with neighboring ConceptNet node vectors.

To illustrate this distinction, consider a simple example with three ConceptNet nodes. Given the claim token “doctor,” EZ-Check retrieves immediate neighbors such as *hospital* and *medical professional* and computes vector attention scores using static embeddings.



Fact: Nasa is a Space Agency
Result: SUPPORTS
Explanation: The triplet "Nasa is A Agency" indicates that Nasa is a type of agency, and the triplet "Nasa related to Space" further supports this claim as it establishes

Fig. 2. Overview of the EZ-Check fact-checking workflow. The system receives a natural-language claim, retrieves relevant ConceptNet triplets through the Attention Query mechanism, converts the selected triplets into readable sentences, and then feeds both the claim and the triplet-based context into the language model. The model outputs a final classification (SUPPORTS, REFUTES, or NOT ENOUGH INFO) along with an explanation derived from the retrieved knowledge. Each step is presented as a numbered stage to illustrate the sequential reasoning process.

The highest-scoring node is selected on the fly, forming the next hop in the query path. KGAT or GEAR, in contrast, would require a pre-trained attention module operating over a fixed subgraph, not a per-claim dynamic retrieval.

This design enables EZ-Check to perform reasoning without training, avoids task-specific fine-tuning, and supports efficient, claim-conditioned knowledge extraction.

3. Methodology

This section outlines our proposed framework to address the problem of explainable fact-checking.

An overview of the framework is presented in Fig. 2. Full pseudo code appears in Appendix A. Additionally, we introduce the fundamental notation and terminology that will be used consistently throughout the paper.

Our work introduces a two-phase approach for zero-shot fact-checking [31,32] using knowledge graphs such as ConceptNet. Zero-shot learning enables models to perform tasks they have not encountered during training.

In the first phase, our novel querying method extracts and prepares triplets relevant to the given fact from ConceptNet. Candidate paths are generated by querying the knowledge graph and creating neighbourhoods for each significant word in the factual statement, excluding stop words. An attention-based selection mechanism is then applied to identify the most relevant neighbourhood paths.

In the second phase, the extracted triplets are reformatted into a human-readable textual structure. These structured triplets, along with the original fact, are input into a language model fine-tuned using few-shot prompting techniques. The language model performs two tasks: classifying the facts as “SUPPORTS,” “REFUTES,” or “NOT ENOUGH INFORMATION,” and generating a corresponding explanation based on this classification.

This approach begins by determining the model’s stance on the fact, followed by generating a clear explanation that uses the extracted knowledge graph data and language model reasoning. This process provides both classification and an explanation to substantiate the decision, ensuring interpretability and reliability (Table 1).

A Toy Example Consider the claim “A penguin can fly.” The entities penguin and fly are extracted and mapped to ConceptNet nodes. From penguin, the system retrieves neighboring triplets including (penguin, CapableOf, swim) and (penguin, NotCapableOf, fly).

Each triplet embedding is compared with the claim embedding using cosine similarity. For this claim, the similarity score for (penguin, NotCapableOf, fly) is 0.82, while (penguin, CapableOf, swim) receives a lower score of 0.31. After normalization, the contradictory relation (NotCapableOf, fly) receives the highest attention weight and is selected for reasoning. Based on this high-attention contradiction, EZ-Check classifies the claim as REFUTES, and the selected triplet forms the explanation.

3.1. Attention query

We propose a method to query sentences in natural language into ConceptNet and extract salient triplets using an attention mechanism. ConceptNet Numberbatch embeddings [15] support querying, while Sentence-BERT [34] guides triplet selection, enabling context-aware extraction aligned with sentence semantics.

Table 1
Terminology clarification for EZ-Check.

Term	Definition and usage in This Work
Vector Attention	A modified multi-head attention mechanism used to compute semantic relevance between fact tokens and ConceptNet nodes. It drives the token-to-node scoring process.
Attention Query	The full querying procedure introduced in Section 3.1, which uses vector attention to extract relevant ConceptNet triplets for each input claim. This is the name of the entire querying method.
Multi-Head Attention	The classical attention architecture[33] that we adapt for ConceptNet embedding comparisons. Used in the internal calculations of vector attention and triplet scoring.
Triplet Selection	The step where extracted ConceptNet triplets are ranked using Sentence-BERT similarity and multi-head attention, selecting those most relevant for generating explanations.
Zero-Shot Fact-Checking	The overall task performed by EZ-Check: assessing claims without fine-tuning or relying on external evidence sentences, while integrating KG-derived knowledge.

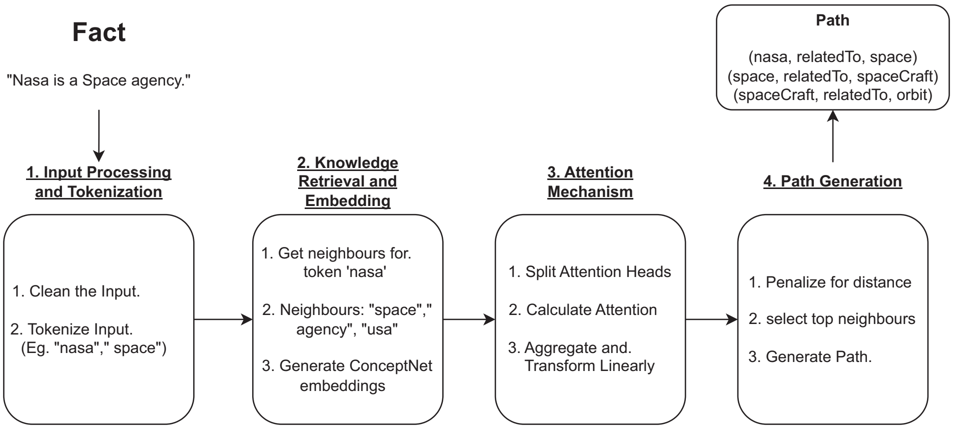

Fact
"Nasa is a Space agency."
1. Input Processing and Tokenization
1. Clean the Input.
2. Tokenize Input. (Eg. "nasa"," space")
2. Knowledge Retrieval and Embedding
1. Get neighbours for. token 'nasa'
2. Neighbours: "space"," agency", "usa"
3. Generate ConceptNet embeddings
3. Attention Mechanism
1. Split Attention Heads
2. Calculate Attention
3. Aggregate and. Transform Linearly
4. Path Generation
1. Penalize for distance
2. select top neighbours
3. Generate Path.
Path
(nasa, relatedTo, space)
(space, relatedTo, spaceCraft)
(spaceCraft, relatedTo, orbit)

Fig. 3. Step-by-step illustration of the Attention Query mechanism. (1) The claim is tokenized and stopwords are removed. (2) Each token is queried in ConceptNet to obtain neighboring nodes. (3) Vector attention scores are computed between the token and its neighbors using ConceptNet embeddings. (4) The highest-scoring neighbor is selected and expanded, forming a path that continues until a hop limit or score threshold is reached. (5) The resulting paths form the candidate triplets used for explanation. This schematic shows how EZ-Check extracts semantically relevant knowledge without performing full graph traversal.

Our proposed attention-based query method involves two primary stages: (1) Querying a sentence into ConceptNet to extract multiple paths representing possible interpretations and (2) Selecting the most important triplets from these paths to construct a coherent explanation. This approach enables extracting the proper triplets, emphasizing the most relevant aspects. This approach is applicable in generating concise and meaningful explanations from complex knowledge graphs.

3.1.1. Querying the ConceptNet

Given a sentence, we first preprocess it by converting all text to lowercase and removing stop words. This ensures a consistent pre-processing with that of older baselines, which use the classical approach, reduce the vocabulary size, and ensure uniformity. Then, we tokenize the text. For each token, we query it in ConceptNet and select its neighbours. We calculate the attention score between neighbours and tokens for all the neighbours and select the neighbour with the highest score. Then, for this neighbour, which we consider the most important neighbour based on the fact sentence, its neighbours are extracted, and the chain continues. The process will continue until the attention score falls below the threshold or we reach a fixed number of hops in the path. We also penalize the attention score if the current node is too far away from the original query token, and the score will be high only if the semantic similarity is very high. We use ConceptNet Numberbatch embedding, a graph-based embedding based on ConceptNet for encoding. This method is preferred as these embeddings encapsulate information related to ConceptNet, facilitating token-relatedness calculation in the graph. An overview of the querying can be seen in Fig. 3.

To obtain the neighbourhood nodes for a fact from ConceptNet, we first calculate the attention between the current token and the fact sentence. Then, we merge the weights with the token to ensure attention is calculated based on the fact sentence, not only with the current token during neighbour selection. Thus, in the second pass, we use these concatenated token weights with the fact sentence and then calculate its attention score with the neighbours of the current token, ensuring the best neighbour, in terms of the fact and the current token, is selected. We utilized the multi-head attention mechanism inspired by vector attention [33] and modified it to use without the need for a feed-forward neural network. Such hybrid attention approaches have shown growing promise in KG reasoning, enabling more adaptive retrieval of concept-level information, further supporting our design choice. It is also optimized to work on knowledge graphs, which allows the model to attend to information from different representation subspaces at different positions by calculating each neighbour's score separately in an attention head and then joining them. It is achieved through a series of linear projections for queries, keys, and values, followed by concatenation and another linear projection. Although the modified method does not continuously learn from the weights due to a lack of a feed-forward network, we address this limitation by using ConceptNet

Numberbatch Embeddings to capture the relationship between the nodes inside the graph and Sentence-Encoder Embeddings to model the relationship between the natural query and knowledge graph facts (nodes).

The number of attention heads is adjusted to ensure it evenly divides the dimension of the embeddings, denoted as d:

$$\text{num\_heads} = \max\left\{k \in \mathbb{N} \,\|\, 1 \le k \le \text{original\_num\_heads and } d \bmod k = 0\right\} \tag{1}$$

$$\text{head\_dim} = \frac{d}{\text{num\_heads}} \tag{2}$$

In simple terms, the Key (K) represents the tokens of the fact we input, the Query (Q) comes from a neighboring node, and the Value (V) refers to the vector of the original fact, which we update using attention weights. The input tensors Q, K, and V are reshaped and transposed to split into multiple heads:

In the equation:

$$\text{split\_heads}(t) = \text{t.view}(b, n, h, d).\text{transpose}(2, 1) \tag{3}$$

The following abbreviations are used for clarity:

- t represents the *tensor* being manipulated.
- b stands for *batch_size*, indicating the size of each batch.
- n is used for *neighbour*, representing the neighboring elements.
- h denotes *num_heads*, the number of attention heads.
- d symbolizes *head_dim*, the dimension of each attention head.

The function *split_heads* is designed to reshape and transpose the tensor t for multi-head attention processing, facilitating parallelized computations across multiple heads.

For each head, the attention mechanism is applied as follows:

$$\text{output} = \text{softmax}\left(\frac{Q \cdot K^T}{\sqrt{\text{head\_dim}}} \cdot distance, \text{dim} = -1\right) \cdot V \tag{4}$$

We penalize the attention score based on the distance between the start token and the current node, ensuring that a distant node is only considered if it is highly relevant.

Finally, the outputs of individual heads are concatenated, and the attention weights are combined: In the following equation, awc stands for the combined attention weights:

$$\text{output} = [\mathbf{H}_1 | \mathbf{H}_2 | \dots | \mathbf{H}_n] \tag{5}$$

$$\text{awc} = \begin{bmatrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \mathbf{W}_n \end{bmatrix} \tag{6}$$

The final output consists of the concatenated output and the combined attention weights, providing both the transformed values and the attention scores. The workflow can be seen in Algorithm 2 and Fig. 3.

3.1.2. Triplet selection

Upon obtaining multiple paths, one from each token, we select the most relevant triplets among them, which are most helpful in explaining the fact.

First, we convert all the triplets into a readable format, combining subject, predicate, and object and considering each as a sentence. For example, a triplet is “NASA RelatedTo Space” and can be converted to a readable format such that it becomes “NASA is related to Space.” Then, we embed the original query (the fact sentence) and triplet sentences using sentence-BERT and compute the attention score between the fact and triplet sentences.

We utilize a vanilla attention mechanism [33] for triplet selection and use multi-head attention. Given queries, keys, and values represented as matrices $Q, K, V \in \mathbf{R}^{\mathbf{d}_{\text{model}} \times \mathbf{d}_{\text{model}}}$, the attention calculation is performed as follows:

1. Apply linear transformations to the queries, keys, and values:

$$\hat{Q} = W_q \cdot Q \tag{7}$$

$$\hat{K} = W_k \cdot K \tag{8}$$

$$\hat{V} = W_v \cdot V \tag{9}$$

where $W_q, W_k, W_v \in \mathbf{R}^{\mathbf{d}_{\text{model}} \times \mathbf{d}_{\text{model}}}$ are the weight matrices.

2. Split the transformed matrices into multiple heads:

$$\text{head\_dim} = \frac{d}{\text{num\_heads}} \tag{10}$$

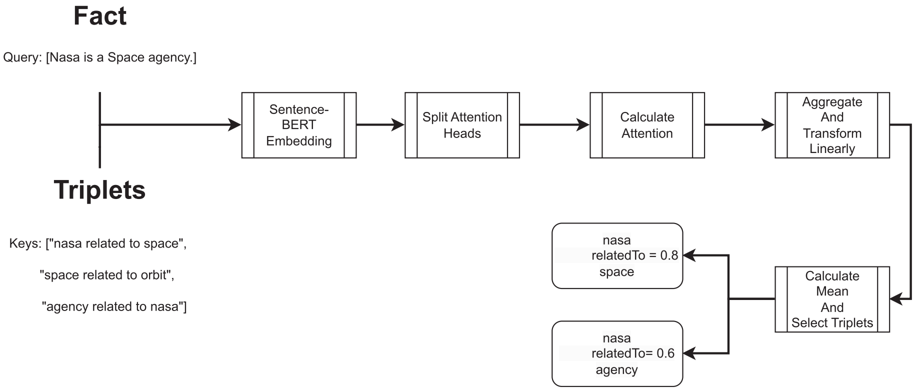

Fact
Query: [Nasa is a Space agency.]
Triplets
Keys: ["nasa related to space",
"space related to orbit",
"agency related to nasa"]
Sentence-BERT Embedding
Split Attention Heads
Calculate Attention
Aggregate And Transform Linearly
Calculate Mean And Select Triplets
nasa relatedTo = 0.8 space
nasa relatedTo= 0.6 agency

Fig. 4. Step-wise process for selecting the most relevant ConceptNet triplets. (1) Each extracted triplet is converted into a readable sentence. (2) Sentence-BERT embeddings of the claim and the triplet sentences are computed. (3) Multi-head attention determines the semantic relevance of each triplet to the claim. (4) Triplets scoring above the mean relevance threshold are selected. (5) These top triplets are used to construct the explanation prompt for the language model. This diagram demonstrates how EZ-Check filters ConceptNet knowledge to produce concise, claim-focused rationales.

3. Compute the attention scores for each head:

$$\text{scores}_h = \frac{\hat{Q}_h \cdot \hat{K}_h^T}{\sqrt{d_{\text{head}}}}, \quad d_{\text{head}} = \frac{d_{\text{model}}}{H}, \quad \forall h \in \{1, \dots, H\} \tag{11}$$

$$\text{attention\_weights} = \text{softmax}(\text{scores}, \text{dim} = -1) \tag{12}$$

$$\text{output} = \text{attention\_weights} \cdot \hat{V} \tag{13}$$

4. Concatenate the output and apply the final linear transformation:

$$\text{final\_output} = W_o \cdot \text{output} \tag{14}$$

where $W_o \in \mathbf{R}^{\mathbf{d}_{\text{model}} \times \mathbf{d}_{\text{model}}}$ is the weight matrix.

After calculating the attention score between the fact sentence and all the triplets inside the extracted paths, we select the most important triplets. An overview of the approach can be seen in Fig. 4 and Algorithm 3. To select the most important triplets from the paths, we take the attention score of all triplet sentences and aggregate them by taking the mean of the scores. Then, we select all the triplets that scored above the mean for further processing to generate an explanation.

3.2. Fact verification

This subsection outlines our method for fact verification. Our strategy integrates data from ConceptNet and the Llama-2 language model [32] to classify a given fact into one of three categories: “SUPPORTS,” “REFUTES,” or “NOT ENOUGH INFO.” Based on this classification, we generate a rationale to explain the model’s decision.

Our methodology uses the Llama-2 7-billion[3] parameter model [32]. While other generative language models could be adapted for this purpose, we chose Llama-2 due to its balance of performance, up-to-date knowledge, and resource efficiency, as it operates effectively on a 16GB VRAM GPU. This powerful and open-source model uses prompt-based inference [35], making it ideal for quickly generating transparent explanations.

Our method outputs two main components for each input fact: the *classification* and the *explanation*. The classification states are “SUPPORTS,” “REFUTES,” and “NOT ENOUGH INFO.” The explanation provides the reasoning behind the classification. This dual-output structure clarifies the model’s stance and lends itself to potential integration in systems requiring explainable AI.

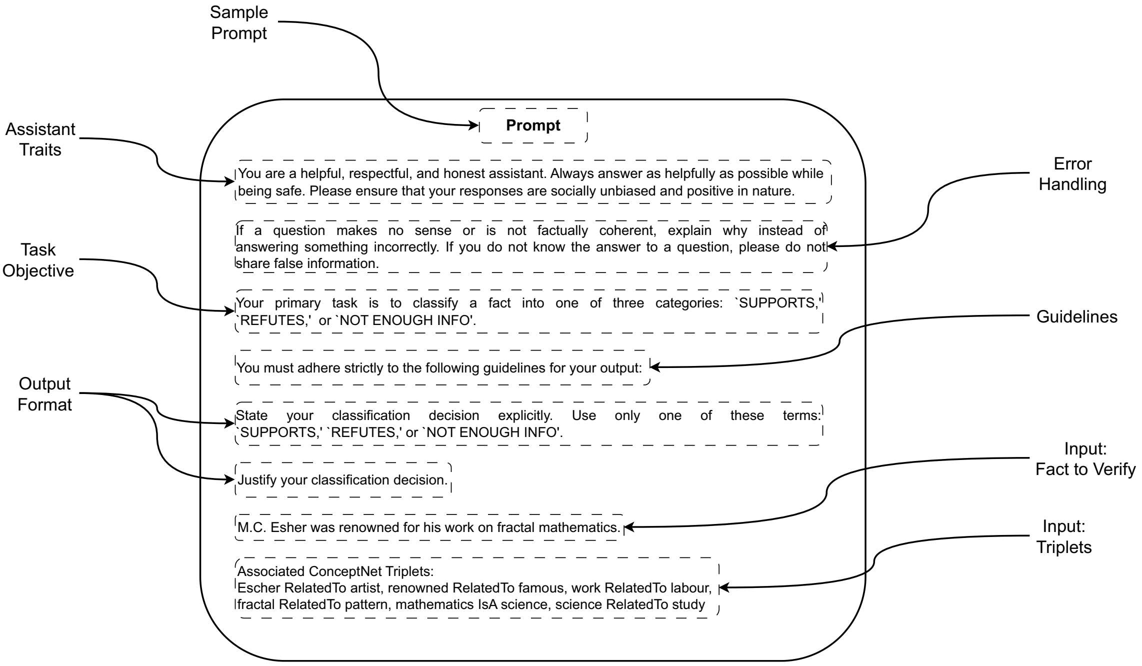
The model receives a structured prompt containing the fact, associated ConceptNet triplets, and a query requesting the model’s stance on the fact. We tuned Llama-2 using few-shot learning prompting [35] on a custom dataset covering various evaluation scenarios. This dataset includes 100 examples for each classification category.

Llama-2’s pre-existing multitasking capabilities augment our fine-tuning process, streamlining tasks such as text classification and summarization. This eliminates the need for resource-intensive adjustments to the model’s deeper layers, resulting in a more efficient training phase and making our approach more practical for real-world applications.

[3] Experiments were finalized in 2023 using LLaMA-2 and GPT-3.5 Turbo. We preserved these settings to ensure consistency across submissions and comparisons. While newer models are available, our framework is designed to be model-agnostic and can adapt to future LLMs.

Table 2
Breakdown of the Prompt components for fact classification task.

Component	Description
Task Objective	Specifies the primary role of the assistant: to classify a given factual statement into one of "SUPPORTS," "REFUTES," or "NOT ENOUGH INFO" and generate an explanation.
Guidelines	The set of guidelines to which the assistant must strictly adhere. For example, we require an explicit classification decision and its justification to be provided.
Input: Fact	Fact to be classified. A claim or factual statement will be provided, which the assistant is expected to classify.
Input: Triplets	The assistant will be given ConceptNet triplets related to the claim. These triplets are intended to assist in making the classification decision.



You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible while being safe. Please ensure that your responses are socially unbiased and positive in nature.

If a question makes no sense or is not factually coherent, explain why instead of answering something incorrectly. If you do not know the answer to a question, please do not share false information.

Your primary task is to classify a fact into one of three categories: `SUPPORTS,' `REFUTES,' or `NOT ENOUGH INFO'.

You must adhere strictly to the following guidelines for your output:

State your classification decision explicitly. Use only one of these terms: `SUPPORTS,' `REFUTES,' or `NOT ENOUGH INFO'.

Justify your classification decision.

M.C. Esher was renowned for his work on fractal mathematics.

Associated ConceptNet Triplets:
Escher RelatedTo artist, renowned RelatedTo famous, work RelatedTo labour, fractal RelatedTo pattern, mathematics IsA science, science RelatedTo study

Fig. 5. This figure demonstrates a sample prompt from the training dataset. The same prompt is also used for testing. Details on different prompt components can be shown in Appendix Table D.19.

3.2.1. Query formulation for explanation generation

We employ a specialized query (prompt), formatted as an "info" prompt [35], to interact with the Llama-2 model, which utilizes chat-based inference. The query aims to guide the model through a two-step process for fact verification: initial classification and subsequent explanation.

The prompt consists of three crucial components:

1. Fact to Classify: This sets the context and subject matter for the model's task, establishing what needs to be verified, as can be seen in Table 2 and in Appendix Table D.18.
2. ConceptNet Triples: These triplets are transformed into a human-readable format to enhance the model's semantic understanding of the fact. They act as supplementary knowledge for making an informed classification. We map ConceptNet relations to their plain-text equivalents to enhance readability. For example, "RelatedTo" becomes "is related to," and "IsA" is transformed into "is a", as can be seen in Table 2 and in Appendix Table D.18.
3. Task Description: This part explicitly instructs the model to classify the fact as "SUPPORTS," "REFUTES," or "NOT ENOUGH INFO," and then justifies this classification by generating an explanation based on the triplets and its own knowledge, as can be seen in Table 2 and in Appendix Table D.18.

A sample prompt can be seen in Fig. 5, and details on each component are shown in the Appendix Table D.19.

To minimize the risk of the model providing false or extraneous information (often referred to as "hallucination"), our prompt is designed to be explicit in its requirements based on the guidelines published [32]. It asks the model to adhere strictly to guidelines, ensuring that the generated explanation is both coherent and directly relevant to the initial classification. This structured query formulation aims to secure reliable and interpretable outputs, enhancing the overall system's credibility, as can be seen in Section D, Appendix Table D.19.

3.2.2. Classification

The language model classifies a given fact in the first stage of our explanation generation pipeline. This classification is guided by a specialized query prompt, as detailed in Section 3.2.1. Leveraging ConceptNet triplets for external knowledge, the model assigns one of the following three categories to the fact:

- **SUPPORTS**: The model verifies the fact, establishing coherence with both the available internal knowledge of the language model and ConceptNet triplets.
- **REFUTES**: If the fact contradicts the ConceptNet triplets or the model's internal reasoning, the "Against" label is applied.
- **NOT ENOUGH INFO**: When the ConceptNet triplets or internal data are inadequate for conclusive judgment, the model opts for this label.

This structured classification serves as a prerequisite for generating reasoned explanations. It ensures more accurate and interpretable results, enhancing the system's reliability. The workflow can be seen in the Algorithm 5.

3.2.3. Explanation generation

Our explanation mechanism operates on a function $g : Y \times Z \rightarrow E$ to generate an explanation E after classification Y is obtained. Y is one of {"SUPPORTS," "REFUTES," "NOT ENOUGH INFO"}. Z comprises ConceptNet triplets T and internal knowledge of language model D, and E is the final explanation output.

- **Classification-Specific Rationalization**: g varies its approach based on Y. For "SUPPORTS," it utilizes triplets that directly affirm the fact. For "REFUTES," it identifies triplets in T that contradict or lack sufficient information, thereby refuting the fact. For "NOT ENOUGH INFO," it delineates what data elements are missing (M) to make a decision.
- **Coherence Check**: A coherence function $h : E \rightarrow [0, 1]$ assesses the logical flow within E.
- **Context Integration**: $E = g(Y, Z, C)$ integrates context C, utilizing conditional probabilities like $P(Y|C)$ to enrich the explanation.
- **Hallucination Mitigation**: g verifies E against T and D to detect and rectify any hallucinated information.

Exemplification using ConceptNet triplets. To illustrate, consider the fact that "M.C. Escher was renowned for his work on fractal mathematics." The associated triplets indicate that Escher is related to artistry, not fractal mathematics. Here, Y = "REFUTES". The justification would indicate that the triplets do not support the fact, and based on the triplets, the claim is wrong, thereby refuting it. See Table 7.

3.3. Knowledge expansion

Attention Query supports the expansion of knowledge based on the ConceptNet ontology. The missing context can be incorporated as additional data for the situations and datasets where further knowledge is necessary due to missing information about the work in ConceptNet. We can combine this additional knowledge into ConceptNet by expanding the ConceptNet API using facts extracted and added to ConceptNet using the same ontology paradigm that ConceptNet uses. This additional knowledge can be used similarly to ConceptNet and work as an extension of ConceptNet itself.

Zero-Shot Clarification. EZ-Check operates entirely in a zero-shot setting. Although Appendix B provides 300 illustrative examples, these were never used to fine-tune or adapt the language model. They serve only as format demonstrations, showing the pretrained model how to interpret subject–relation–object triples and convert them into natural-language reasoning. No model parameters were updated, no fact-checking labels were provided, and no domain-specific supervision occurred.

DBpedia Zero-Shot Evaluation. To further validate the zero-shot nature of the framework, we also tested EZ-Check on DBpedia using the GPT-OSS-120B model. These experiments were conducted in a strict zero-shot configuration: the model received *only* the claim and the retrieved DBpedia triples, with no prompt-format examples, template demonstrations, or tuning of any kind. This shows that EZ-Check generalizes across knowledge graphs and model architectures without relying on prompt engineering or example guidance, reinforcing its training-free design.

4. Experiments

This section describes datasets, evaluation metrics, and baselines. In our research, we assessed two distinct tasks:

1. Attention Query: We evaluated our novel vector-based querying method against a baseline, which performs natural language queries.
2. Fact-Checking: We evaluated the novel EZ-Check for fact verification. This is further divided into two parts:
    - Classification (Task I): Assessing EZ-Check's accuracy in fact verification by determining whether its classification of input claims correctly reflects their truthfulness (SUPPORTS or REFUTES).
    - Explanation (Task II): The evaluation of the effectiveness of the generated explanations for the claims.

Furthermore, we also conducted an ablation study to assess the impact of different parts of the proposed method on the outcome.

Table 3
Statistics of FEVER and UKP Snopes datasets. Units are word count.

Dataset	Claims	Avg. claim length	Content length	Explanation length
FEVER	16K	8	122	28
UKP Snopes	6K	14	156	73

4.1. Datasets

We evaluated the performance of our approach on two large, challenging and human-annotated fact-checking datasets: FEVER [36] and UKP Snopes [37].

FEVER[4] is a sizable shared task dataset comprising 5,416,537 Wikipedia pages from the June 2017 Wikipedia dump and 185,455 claims. UKP Snopes[5] consists of 16,508 Snopes pages and a mixed-domain dataset. Both datasets contain classification labels as "REFUTES," "SUPPORTS," and "NOT ENOUGH INFO." FEVER also contains human-annotated ground truth explanations supporting each claim's veracity. More details are provided in Table 3. For both datasets, we removed samples that had less than 60% of their content represented in the ConceptNet embeddings. This ensured we had enough embedding coverage to apply our method effectively. After cleaning, we retained 16,000 claims from the FEVER dataset and 6000 from the UKP Snopes dataset. To ensure a fair comparison, all approaches used the same cleaned data. We used the stock ConceptNet embeddings without any tuning to avoid giving our zero-shot method an unfair advantage, while ensuring the embeddings remained meaningful. The datasets used in this study show a clear class imbalance. In the UKP sample, REFUTES accounts for 58.2 percent, while NOT ENOUGH INFO accounts for 22.5 percent, and SUPPORTS accounts for 19.3 percent of the data. The FEVER sample is similarly imbalanced, with SUPPORTS comprising 72.9 percent, REFUTES 25.0 percent, and NOT ENOUGH INFO only 2.1 percent. Because we aimed to evaluate the models on the original distributions without altering the underlying task characteristics, we did not rebalance the data, as oversampling or undersampling would introduce artificial bias and reduce comparability across methods.

4.2. Pre-processing

For fairness, we apply the same pre-processing to every baseline that accepts raw text; Llama 2 and GPT-3.5 [38] receive original casing because their tokenisers are cased. For pre-processing, we lowercase the text, remove stopwords, and tokenize the text using whitespace splitting.

4.3. Attention query

4.3.1. Experiment setup and dataset

We used the FEVER dataset [36] to test the attention query, as it provides explanations for triplet verification with BERTScore [39]. We selected only entries where all tokens were represented in ConceptNet Numberbatch embeddings, ensuring complete vector coverage for a fair comparison. Post this filtering, a total of 1000 entries were earmarked for examination. These entries were employed across all methodologies, encompassing two baseline techniques and two distinct attention query variants. We compared the results against the following baselines:

- **Howard** [17]: This method aims to extract a domain-focused subgraph from ConceptNet.
- **Shangwen** [18]: This method aims to select and extract a relevant neighbourhood from the ConceptNet.

4.3.2. Evaluation

For evaluation, we utilize BERTScore[6] [39], a metric that calculates the similarity between predicted and ground-truth sequences using contextual embeddings. BERTScore has demonstrated a strong correlation with human evaluations as verified by the original BERTScore paper [39]. Unlike metrics that rely on exact word matching, BERTScore can recognize semantically relevant terms derived from ConceptNet that may differ from the exact wording of the ground truth explanation.

To calculate the BERTScore for a given explanation l and the corresponding extracted triples from ConceptNet $(triplet\_l_1, \ldots, triplet\_l_n)$, t is the element index of the word in the generated explanation, the following formula is applied and the variable T represents the total number of triplets being evaluated.

$$\text{score\_topic}_t = \max_{[1,\ldots,n]} \text{BERTScore}(l_t, triplet\_l_{ti}) \tag{15}$$

The overall evaluation score for the model is computed as the average BERTScore across all triplets:

$$\text{score\_model} = \frac{1}{T}\sum_{t=1}^{T} \text{score\_triplet}_t \tag{16}$$

[4] https://fever.ai/dataset/fever.html.

[5] https://tudatalib.ulb.tu-darmstadt.de/handle/tudatalib/2081.

[6] Results are computed using the official implementation: https://github.com/Tiiiger/bert_score.

Table 4
Baseline and comparative models used in fact verification for FEVER and UKP Snopes Datasets: Each model is selected for its unique features or contributions to the field, offering a comprehensive comparison to our proposed approach. The chosen models represent a range of complexity, from simple baselines to more sophisticated graph-based and topical coherence methodologies. Models marked with w/o ES indicate variants where evidence sentences (ES) are not used as input, and instead the claim text is fed directly (e.g., via Llama-2).

Model	Rationale for Selection	Datasets
FEVER [40]	Foundational standing in fact verification, providing a primary benchmark.	FEVER
Athene [10]	Entity-linking in document retrieval.	FEVER
Athene w/o ES	Modified version of Athene, using Llama-2 for input instead of evidence sentences.	FEVER
GEAR [19] w/o ES	Modified version of GEAR, using Llama-2 for input instead of evidence sentences.	FEVER
KGAT [9] w/o ES	Modified version of KGAT, using Llama-2 for input instead of evidence sentences.	FEVER
BERT as KB [27]	Utilizes inherent knowledge within language models for fact-checking.	FEVER
BERT [41] Finetune	Fine-tuning capability with pre-trained language models.	FEVER
HiGIL [6]	Hierarchical graph learning structure.	FEVER
FACE-KEG [11]	Explainable fact verification through KG.	FEVER
Llama-2.7b [32] Quant	Quantized model parameters for comparison in size and efficiency.	FEVER
GPT-3.5 [38] Turbo	Advanced language modelling capabilities for natural language understanding.	FEVER
WikiCheck [42]	Focus on leveraging Wikipedia.	FEVER
KGAT [9]	Nuanced approach to fact verification with kernel-based attention in graph structures.	FEVER, UKP
GEAR [19]	Graph-based evidence aggregation and reasoning framework.	FEVER, UKP
TARSA [7]	Stance-aware aggregation and consistency.	FEVER, UKP
Random	Sets the lowest performance threshold.	UKP
Majority vote	Basic ensemble learning.	UKP

4.4. *Task I: classification*

In this task, we evaluate the model's ability to categorize claims into one of three classes: SUPPORTS, REFUTES, or NOT ENOUGH INFO.

4.4.1. *Experimental setup and dataset*

We employed FEVER and UKP Snopes datasets to assess the proficiency of our method in claim verification. For benchmarking, we integrated diverse baselines, covering an array of techniques, including Language Models [27], Neural Networks [10], Graph Networks [6], and knowledge graphs [9].

In the evaluation, we followed standard practices by using the F1 score as the metric, which is the official metric of the FEVER challenges.

All the selected baselines for FEVER and UKP Snopes datasets can be seen in Table 4.

4.5. *Task II: explanation*

This task focuses on generating natural language explanations for the claims based on the classification task.

4.5.1. *Experiment setup and dataset*

We utilized the FEVER dataset to evaluate the explanation ability of our approach. This resulted in 16,000 samples. Seven diverse baselines were selected for comparison, and their descriptions can be found in Table 5. Aligning with previous state-of-the-art studies, the following standard metrics were used for evaluation.

- **BLEU** [43]: The Bilingual Evaluation Understudy (BLEU) score is commonly used in machine translation but can also be adapted to compare generated explanations against ground truth explanations. BLEU assesses the quality of a generated text by comparing n-gram overlaps between the candidate and reference sentences.
- **ROUGE** [44]: The Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is another metric employed for assessing the quality of generated explanations against a ground truth. Unlike BLEU, ROUGE focuses on recall, measuring the amount of overlap between the n-grams found in the generated text and those in the reference text. This metric is particularly useful for ensuring that key information elements are present in the generated explanations.

5. Results and discussion

In this section, we present and discuss our findings for our proposed approach, "EZ-Check," and compare it with baseline models. Our analysis is organized into four main parts:

1. Attention Query evaluation
2. Classification on the FEVER and UKP Snopes datasets (Task I)
3. Explanation generation performance (Task II)
4. Insights from the Ablation Study

Table 5
Summary of baselines for explanation task on FEVER Dataset.

Model	Rationale for Selection
BHC	Graph-to-Sequence Learning using Gated Graph Neural Networks. Emphasizes addressing the parameter explosion problem in previous work [45].
KBLLH	Text Generation from knowledge graphs with Graph Transformers. Features a graph-transforming encoder without linearization or hierarchical constraints, particularly applied to the scientific text domain [46].
CL	Graph Transformer for Graph-to-Sequence Learning. Incorporates explicit relation encoding for efficient global graph structure modelling in text generation from AMR and syntax-based neural machine translation [29].
FACE-KEG	Fact Checking Explained using knowledge graphs. Uses automatic explainable fact-checking via knowledge graph construction and encoding, leveraging semantic contextual cues for veracity detection and explanation generation [11].
Llama-2.7b Quant	Chosen for its efficient model size and explanation capabilities through quantized parameters [32].
GPT-3.5 Turbo	Advanced language modelling for rich textual explanations, serving as a benchmark for evaluating explanation ability [38].
GEAR without ES	A modified version of GEAR without extracting evidence sentences [19].

5.1. Attention query mechanism: effectiveness and insights

This section evaluates the efficacy of the proposed attention query mechanism, introduced in Section 4.3, in extracting pertinent triplets from the ConceptNet knowledge graph. Our analysis spans two distinct triplet representations: the tokenized format, where triplets are treated as independent entities, and the aggregated format, where triplets are consolidated into a coherent, human-readable sentence. The results, summarized in Table 6, demonstrate the superior performance of our method over existing baselines, including Howard [17] and Shengwen [18].

The attention query mechanism achieves F1 scores of 39 (tokenized) and 44 (aggregated), outperforming Shengwen's 33 and 39 and Howard's 32 and 36, respectively, on the respective evaluation metrics. This improvement highlights the robustness of our approach in retrieving semantically meaningful triplets, even under sparse or incomplete knowledge graph conditions. In contrast, baseline models exhibit significant limitations: Howard's method struggles to prune irrelevant nodes effectively, while Shengwen's approach frequently fails to return meaningful results due to its dependence on fully connected graphs after pruning, with up to 80% of cases yielding no output.

5.1.1. Dynamic attention and node relevance

A defining feature of our method is the dynamic allocation of attention, which allows the model to prioritize relevant nodes without exhaustively traversing neighbourhood graphs. This capability is crucial for efficient and targeted extraction, particularly in complex graphs where irrelevant information abounds. Unlike traditional approaches, our method balances precision and computational efficiency, enabling more reliable retrieval of knowledge.

This advantage is evident in comparative evaluations such as *AQ vs GT Explanation* and *AQ vs Claim NRL*. These comparisons reveal that our attention query mechanism consistently aligns extracted triplets with ground truth explanations and original claims more effectively than the baselines. Importantly, the results suggest that explicit relationships between triplets are not always necessary for strong alignment, highlighting the flexibility and adaptability of our approach.

5.1.2. Comparative analysis with baselines

To further substantiate the performance of our model, we conducted detailed comparisons against baseline methods. Table 6 presents metrics such as *AQ vs. GT Explanation* and *AQ vs. Claim*, where our model demonstrates a significant performance edge. For instance, in sentence-level evaluations, our method achieves higher F1 scores even in scenarios where relational information between triplets is less critical (*AQ vs Claim NRL*). This finding underscores the versatility of our attention query mechanism in diverse evaluation contexts.

In contrast, Howard's model, which employs cosine similarity for node selection, often retrieves excessively irrelevant information due to insufficient pruning. Similarly, Shengwen's dependency on fully connected post-pruning graphs leads to significant output sparsity, particularly when the input graph lacks connectivity. These limitations emphasize the superiority of our approach in efficiently navigating sparse and heterogeneous data landscapes.

5.1.3. Impact of triplets on classification score

The inclusion of ConceptNet triplets in our methodology has a transformative impact on classification tasks. As illustrated in Table 7, the integration of these triplets facilitates transitions from *NOT ENOUGH INFO* to *REFUTES* or *SUPPORTS* classifications, resulting in marked improvements in scores and rationale generation.

For example, in analyzing claims related to M.C. Escher's work, the model initially categorizes the claim as *NOT ENOUGH INFO* due to insufficient contextual knowledge. However, with the inclusion of relevant triplets, the model confidently classifies the claim as *REFUTES*, demonstrating an enhanced understanding of the underlying semantics. Similarly, the classification of the *Fluffernox*

Table 6
Coverage evaluation of triplets using BERTScore F1 to assess their adequacy in representing facts. The assessment involves two methodologies: one with tokenized claims and triplets and another treating triplets and claims as individual sentences. The table contrasts performances based on different BERT variants and verification setups, distinguishing between our proposed method and baseline approaches.

Type	Comparison Type	BERTScore F1 Score %	
		Token	Sentence
Proposed Approach	AQ vs GT Explanation	33	37
	AQ vs GT Explanation NRL	30	40
	AQ vs Claim	**39**	41
	AQ vs Claim NRL	37	**44**
Baselines	Howard vs GT Explanation	26	**36**
	Howard Clean vs GT Explanation	26	36
	Howard vs Claim	**32**	34
	Howard Clean vs Claim	32	34
	Shengwen vs GT Explanation	26	34
	Shengwen vs Claim	**33**	**39**

Table 7
Comparison of classification outcomes with and without the aid of ConceptNet triplets. Using triplets from sources such as ConceptNet enhances the scores and justifications of the classification.

Claim	Without Triplets	With Triplets
M.C. Escher was renowned for his work on fractal mathematics.	NOT ENOUGH INFO	REFUTES
Albert Einstein formulated the Third Law of Thermodynamics and not by Walther Nernst.	SUPPORTS	REFUTES
The Fluffernox species primarily consumes the Drakolian fruit.	REFUTES	SUPPORTS

species claim shifts from *REFUTES* to *SUPPORTS*, further showcasing the pivotal role of triplets in fine-tuning decision-making processes. These examples, detailed in Table 7, highlight the critical contribution of the attention query mechanism to both score and interpretability.

5.2. Task I: classification

Note on Baseline Comparability. To ensure a fair evaluation, we separate baselines into two groups: (1) *claim-only* models that operate without external evidence, and (2) *claim + evidence* models that rely on retrieved sentences. Because evidence sentences often contain the explicit ground-truth answer, evidence-based models may achieve higher accuracy for reasons unrelated to model reasoning. EZ-Check does not use any external evidence sentences; it relies solely on knowledge-graph retrieval through the Attention Query mechanism. Therefore, comparisons should be interpreted in terms of design objectives rather than direct data parity.

ConceptNet Coverage Threshold. We applied a 60% ConceptNet coverage threshold to ensure that each claim had sufficient KG support to produce meaningful triplets. Empirically, lower thresholds introduced substantial noise, as claims with sparse or semantically distant neighbors led to unstable attention scores and reduced explanation quality. Higher thresholds, however, removed too many samples and biased the evaluation toward easier cases. The 60% value provided a balanced trade-off, retaining most claims while ensuring adequate KG grounding. In total, $20K$ FEVER claims and $8K$ UKP claims were excluded due to insufficient ConceptNet coverage.

5.2.1. Performance on FEVER dataset

On the FEVER dataset, *EZ-Check* achieved an F1 score of 60 using only the claim for classification, as shown in Table 8. This result demonstrates a notable improvement over claim-only baselines such as BERT Finetune (59) and GPT-3.5 Turbo (55). While models such as GEAR and TARSA, which rely on additional evidence sentences, achieved slightly higher scores, *EZ-Check*'s efficiency in requiring minimal input highlights its practicality and adaptability. Key observations include:

- **Efficiency with Minimal Input:** *EZ-Check* outperformed other claim-only models, leveraging its ability to extract meaningful semantic information from claims alone. Its performance demonstrates the potential for efficient verification without dependence on external evidence.

Table 8
Evaluation of models on the FEVER dataset for the classification task of determining SUPPORTS, REFUTES, or NOT ENOUGH INFO, using the F1 score as the metric. Models are grouped into baselines and our proposed approach, with further details provided on the input requirements for each model.

Performance Analysis on the FEVER Dataset			
Category	Input Type	Model	F1 %
Baselines	Claim & Evidence Sentences	FEVER Baseline	48.92
		Athene	69.80
		GEAR	73.90
		KGAT	**74.20**
		TARSA	70.70
		HIGIL	73.60
		FACE-KEG	**73.90**
		WikiCheck	57.50
	Claim Only	Athene Modified	52
		GEAR Modified	54
		KGAT Modified	53.80
		BERT as KB	44
		BERT Finetune	**59**
		Llama-2 7b Quant	53
		GPT-3.5 Turbo	55
Proposed Method	Claim Only	EZ-Check	**60**

- **Comparison with Evidence-Dependent Models:** While models such as TARSA and GEAR surpassed *EZ-Check* in absolute scores, their reliance on additional evidence underscores the unique efficiency of *EZ-Check*, which achieves competitive performance using significantly less input.

These results validate *EZ-Check*'s capability to adapt to scenarios where evidence acquisition is constrained, making it highly suitable for practical applications. An overview can be seen in Table 11.

5.2.2. Performance on UKP snopes dataset

On the UKP Snopes dataset, *EZ-Check* achieved an F1 score of 64, outperforming all evaluated baselines, as summarized in Table 9. Notably, this performance was achieved using only the claim, while competing models like TARSA required both claims and evidence sentences. Key insights include:

- **Superior Performance with Minimal Data:** *EZ-Check* surpassed all baselines, including TARSA (57), while operating with minimal input. This efficiency highlights its advantage in scenarios where extensive evidence collection is impractical.
- **Robustness Across Baselines:** The model's performance against both simple methods (e.g., Majority vote, Random baselines) and advanced graph-based approaches demonstrates its adaptability and effectiveness.

These findings reinforce *EZ-Check*'s versatility and robustness, showcasing its ability to deliver accurate fact verification even under resource-constrained conditions.

5.2.3. Discussion and implications

The classification results on FEVER and UKP Snopes highlight *EZ-Check*'s strengths and real-world potential:

- **Efficiency and Practicality:** Achieves high accuracy with minimal input, addressing challenges where external evidence is costly or infeasible.
- **Zero-Shot Readiness:** Leverages ConceptNet and language models for deployment without fine-tuning or customization.
- **Robustness Across Datasets:** Consistent performance across diverse datasets demonstrates adaptability to varying fact verification tasks.
- **Explainability Foundation:** Strong classification results support downstream explanation generation, aligning with explainable AI goals.

Overall, *EZ-Check*'s efficiency, zero-shot performance, and robustness position it as a practical solution for fact verification in resource-constrained environments, while its explainability focus ensures transparent and interpretable decision-making.

5.2.4. Additional evaluation using binary-class analysis and DBpedia

Binary-class statistical evaluation (Merged REFUTES + NOT ENOUGH INFO). To further analyze the behavior of EZ-Check under alternative label groupings, we conducted a binary-class evaluation by merging the *REFUTES* and *NOT ENOUGH INFO* labels into a single

Table 9
Assessment of different models on the UKP Snopes dataset for classifying into SUPPORTS, REFUTES, and NOT ENOUGH INFO categories. The table showcases F1 scores and each model's input prerequisites. The superior performance of our proposed method, "EZ-Check," is emphasized by the bolded scores.

Performance of models on UKP Snopes dataset			
Category	Required Input	Model	F1 %
Baselines	Claim & Evidence Sentences	GEAR	35.0
		KGAT	46.0
		TARSA	57.0
	Claim Only	Random baseline	33.0
		Majority vote	24.0
Proposed Method	Claim Only	EZ-Check	**64.0**

Table 10
DBpedia-Based Merged-Class evaluation (REFUTES + NEI = AB).

Dataset	F1 %	AB Precision	AB Recall	SUPPORTS Recall
FEVER	0.366	0.276	0.817	0.203
UKP Snopes	0.645	0.803	0.699	0.279

class (denoted **AB**). This setting is commonly used in fact-checking scenarios where the goal is to distinguish supported claims from all non-supported ones.

FEVER. After applying the merge, the classifier exhibited strong negative-class behavior, achieving high recall for AB (**0.817**) but low precision (**0.276**). SUPPORTS remained challenging, with recall only **0.203**. The resulting accuracy was **0.369**, which is substantially lower than the SUPPORTS-majority baseline of 0.729. This indicates that FEVER's label imbalance makes positive verification more difficult for a KG-based system. Results are shown in Table 10.

UKP snopes (N = 1900). In contrast, the UKP dataset showed significantly higher accuracy under the merge, reaching **0.618**. Precision and recall for AB were **0.803** and **0.699**, respectively, while SUPPORTS remained the more difficult class (precision 0.181, recall 0.279). Because UKP is AB-heavy, the merged structure aligns more closely with EZ-Check's reasoning patterns. Results are shown in Table 10.

Stability across increasing sample sizes. Across FEVER evaluations from 1k to 7k samples, accuracy converged between **0.36–0.37** and macro-F1 around **0.36**, suggesting stable behavior at scale. For UKP, accuracy rose from **0.560** (1k) to **0.618** (full dataset) with tightening confidence intervals. These patterns confirm that the merged-class results are statistically reliable. Results can be found in Table 10.

DBpedia. To further examine the robustness and generalizability of EZ-Check beyond ConceptNet, we conducted an additional evaluation using **DBpedia [12]** as an alternative knowledge graph. This aligns with our Knowledge Expansion strategy, where EZ-Check is designed to operate in a zero-shot setting across different structured sources. After retrieving DBpedia triples using the same vector-attention querying mechanism, we evaluated the model on the FEVER and UKP Snopes predictions using a merged negative class, where *REFUTES* and *NOT ENOUGH INFO* were combined into a single category (AB). This binary-class view provides insight into how EZ-Check behaves when supported by a different KG structure and when the fact-checking task is reframed around distinguishing supported claims from all non-supported ones. The results reveal consistent behavior across ConceptNet and DBpedia: strong recall for AB, lower recall for SUPPORTS, and dataset-dependent variation driven by label imbalance. These findings reinforce that EZ-Check's reasoning generalizes across knowledge graphs and highlight the persistent challenge of positive-evidence attribution when operating solely on commonsense or ontology-based resources.

5.2.5. Cross-dataset performance divergences

A direct quantitative comparison between FEVER and UKP Snopes reveals a significant performance gap. Under the merged-class setting, UKP achieves **+25 percentage points** higher accuracy than FEVER. A two-proportion z-test confirms this difference is statistically significant (<0.001 p<0.001).

The divergence arises from dataset structure: UKP's AB-majority distribution pairs naturally with EZ-Check's commonsense-based extraction, which excels at identifying contradictions or missing information. In contrast, FEVER requires richer positive evidence for SUPPORTS, which is more difficult to extract from ConceptNet alone, leading to lower recall on FEVER.

Table 11
Overall comparison of baseline and EZ-Check, highlighting evidence requirements, fine-tuning dependency, dataset compatibility, and classification performance (F1).

Model	KB / Graph	Evidence	Tune	Compat.	F1
BASELINES – FEVER					
FEVER Baseline	Wiki	Yes	Yes	No	48.92
Athene	Wiki	Yes	Yes	No	69.80
GEAR	Wiki	Yes	Yes	No	73.90
KGAT	Wiki + KG	Yes	Yes	No	74.20
TARSA	Wiki	Yes	Yes	No	70.70
HIGIL	Wiki	Yes	Yes	No	73.60
FACE-KEG	Wiki + KG	Yes	Yes	No	73.90
WikiCheck	Wiki	Yes	Yes	No	57.50
Athene (Modified)	None	No	Yes	No	52.00
GEAR (Modified)	None	No	Yes	No	54.00
KGAT (Modified)	KG	No	Yes	No	53.80
BERT as KB	None	No	Yes	No	44.00
BERT Fine-tuned	None	No	Yes	No	59.00
Llama-2 7B (Quant)	None	No	No	Partial	53.00
GPT-3.5 Turbo	None	No	No	Partial	55.00
BASELINES – UKP SNOPES					
GEAR	Wiki	Yes	Yes	No	35.00
KGAT	Wiki + KG	Yes	Yes	No	46.00
TARSA	Wiki	Yes	Yes	No	57.00
Random Baseline	None	No	No	Partial	33.00
Majority Vote	None	No	No	Partial	24.00
EZ-CHECK (ConceptNet)					
EZ-Check [FEVER]	**ConceptNet**	**No**	**No**	**Yes**	**60.00**
EZ-Check [UKP]	**ConceptNet**	**No**	**No**	**Yes**	**64.00**
EZ-CHECK (DBpedia, merged REFUTES + NEI = AB)					
EZ-Check [FEVER]	**DBpedia**	**No**	**No**	**Yes**	0.366
EZ-Check [UKP]	**DBpedia**	**No**	**No**	**Yes**	0.645

Legend: Tune = fine-tuning; Compat. = multi-datasets compatibility out of the box.

5.2.6. Limitations: difficulty in positive evidence attribution

Across both datasets, EZ-Check consistently shows strong capability in identifying unsupported or contradictory claims but lower recall for SUPPORTS. This reflects a structural limitation of commonsense-KG reasoning: ConceptNet contains richer relational and definitional contradictions than explicit confirmatory evidence paths. As a result, the system more easily detects missing or incompatible information than affirmative support. Future extensions may incorporate additional structured resources or relation-weighted attention mechanisms to strengthen SUPPORTS grounding while preserving the zero-shot design.

5.3. Task II: explanation

Beyond classification, the ability of *EZ-Check* to provide explanations for its decisions is a critical step toward achieving explainable AI. This section evaluates the model's performance in generating meaningful and coherent explanations using the FEVER dataset, with a focus on balancing input simplicity and explanation quality.

Qualitative Evaluation. In addition to ROUGE and BLEU, which we retain for reproducibility and comparison with existing work, we conducted a qualitative rationale analysis to better assess explanation quality. For each claim, we verified whether the generated explanation is factually grounded by examining its alignment with the ConceptNet triplets extracted during the Attention Query stage. We further included several annotated success and failure cases to illustrate the clarity, correctness, and usefulness of the explanations. Finally, we performed an internal author-based review of 80 samples from FEVER and 90 samples from UKP, randomly selected, enabling qualitative assessment without requiring new human annotation. This combined analysis provides a deeper understanding of factual consistency and explanation reliability beyond n-gram overlap metrics.

5.3.1. Interpretability analysis

To assess whether EZ-Check's explanations genuinely aid human understanding rather than simply paraphrasing ConceptNet triplets, we conducted a qualitative interpretability analysis using the examples in Table 7 and user study. We observed that some explanations closely rephrase the underlying triplets, particularly for claims with limited ConceptNet coverage. These outputs are factually aligned but offer limited additional insight beyond the retrieved knowledge.

In contrast, other explanations demonstrate meaningful reasoning behavior. For several claims, EZ-Check combines multiple triplets into a coherent causal or definitional chain, improving clarity and helping users understand *why* the model reached a particular

Table 12
Evaluation on the FEVER dataset for the explanation task. Models are grouped based on input type requirements.

Performance analysis on the FEVER Dataset				
Category	Input Type	Model	Rogue	Bleu
Baselines	Claim & Evidence	BHC	27.60	24.10
		KBLLH	38.20	34.20
		CL	32.90	29.90
		FACE-KEG	41.20	37.00
Baselines	Claim Only	Llama-2 7b Quant	24.80	19.20
		GPT-3.5 Turbo	27.10	22.00
		GEAR Modified	26.00	22.10
Our Approach	Claim Only	EZ-Check	29.10	24.50

label. These multi-triplet explanations illustrate the system's ability to integrate contextual knowledge rather than relying on shallow paraphrasing.

Across cases, we find that interpretability depends on the richness of available ConceptNet nodes and the strength of the vector-attention selection process. When relevant triplets are diverse and semantically connected, EZ-Check produces explanations that improve user comprehension; when triplets are sparse or generic, explanations become shorter and more paraphrastic.

5.3.2. Performance on the FEVER dataset

The FEVER dataset serves as the benchmark to assess *EZ-Check*'s explanation generation capabilities. Table 12 provides a detailed comparison of the model's results against a range of baselines, including graph-based models such as FACE-KEG, language models such as GPT-3.5 Turbo (SOTA at the time of testing), and hybrid approaches.

EZ-Check achieves a Rouge score of 29.10 and a Bleu score of 24.50, significantly outperforming claim-only baselines such as Llama-2 7b Quant and GPT-3.5 Turbo. While models like FACE-KEG, which incorporate evidence sentences, achieve marginally higher scores, *EZ-Check*'s ability to produce competitive explanations from only the claim highlights its efficiency and practicality. Key observations include:

- **Efficiency with Minimal Input:** Unlike evidence-dependent models, *EZ-Check* generates meaningful explanations using only the claim. For instance, it outperforms BHC, which requires additional evidence sentences, achieving higher Rouge and Bleu scores with fewer input requirements.
- **Comparison with Advanced Models:** *EZ-Check* surpasses Llama-2 7b Quant and GPT-3.5 Turbo, which also operate on claims alone, demonstrating its superior semantic understanding and explanation quality.

5.3.3. Insights and implications for explanation quality

EZ-Check demonstrates a strong balance between simplicity in input and richness in output. While evidence-intensive models like FACE-KEG achieve higher Rouge and Bleu scores by leveraging external sources, *EZ-Check*'s ability to generate coherent explanations from minimal input highlights its practicality in resource-constrained environments. Its robust zero-shot capabilities further support deployment without fine-tuning or reliance on external evidence, aligning well with goals in explainable AI.

Future improvements could enhance explanation quality through more advanced semantic modeling and refined integration of knowledge graph triplets, enriching the coherence and detail of outputs. Overall, *EZ-Check*'s performance underscores its readiness for real-world fact verification while laying a strong foundation for advancing explanation quality in interpretable and efficient AI systems.

5.3.4. Human evaluation of explanation quality

To supplement automatic metrics such as ROUGE and BLEU, we conducted an author-based user study following a three-point scoring rubric:

- Score (1): *Incorrect* the reasoning fails to justify the label or contradicts it
- Score (2): *Neutral* partially aligned but unclear or incomplete
- Score (3): *Satisfied* the reasoning clearly and correctly supports the assigned label

Eighty FEVER and ninety UKP samples were independently rated by three authors. Table 13 presents four representative FEVER examples: two high-quality cases where all annotators assigned the maximum score (3,3,3) and two low-quality cases where all assigned (1,1,1). These illustrate when model explanations succeed in grounding predictions in knowledge-graph evidence and when they fail due to misalignment, missing context, or unjustified reasoning.

Overall scoring results. Across 80 FEVER samples, the total assigned score was **489/720**, where 720 is the maximum possible score (80 samples × 9 points). For the UKP Snopes set, the final score was **469/810** across 90 samples (maximum 810 = 90 × 9 points).

Table 13
Representative FEVER examples from the human evaluation study. Two high-quality and two low-quality samples are shown to illustrate the scoring rubric.

Claim	Gold	Pred.	Rationale Summary
Good (3,3,3) There is a wealthy place called Brentwood.	SUPPORTS	SUPPORTS	Correctly identifies Brentwood as a real location supported by KG facts, while noting that "wealthy" is unconfirmed. Reasoning is precise and label-aligned.
Good (3,3,3) The *Underworld* film series has a combined budget under $2M.	REFUTES	REFUTES	Explanation correctly states that each film has a budget far exceeding $20M, making the claim impossible. Clear justification matching the gold label.
Poor (1,1,1) George W. Bush did not pass multiple economic programs intended to preserve the financial system.	SUPPORTS	SUPPORTS	The explanation incorrectly claims "lack of contradictory data" as support, even though the triplets contain no relevant evidence. Misaligned justification leads to incorrect reasoning score.
Poor (1,1,1) Event management includes coordinating with vendors.	SUPPORTS	SUPPORTS	Reasoning incorrectly infers vendor coordination despite no KG evidence supporting that detail. The explanation overgeneralizes and does not justify the label appropriately.

These results indicate that, while most explanations are well-aligned with KG evidence, certain cases reveal systematic weaknesses such as overgeneralization or insufficient grounding due to lack of information in KGs.

5.4. Ablation study

We conducted an ablation study to evaluate the contributions of key components within our model. The study focused on classification accuracy, measured by F1 scores on the FEVER and UKP Snopes datasets, and explanation quality, assessed using Rouge and Bleu scores on the FEVER dataset.

5.4.1. Results and observations

Tables 14, 16, and 17 present the results of the ablation study, highlighting the importance of the attention mechanism, ConceptNet Numberbatch Embeddings, and contextual triplet selection. Key findings include:

- **Impact of Removing the Vector-Attention:** The absence of the attention mechanism caused a marked decline in classification performance, reducing F1 scores from 60 to 50 on the FEVER dataset and from 64 to 54 on UKP Snopes. Explanation quality was similarly affected, with Rouge and Bleu scores dropping from 29.1 to 16.8 and from 24.5 to 12.5, respectively. These results emphasize the critical role of the attention mechanism in identifying and focusing on salient nodes and triplets.
- **Effect of Embedding Choice:** Replacing ConceptNet Numberbatch Embeddings with GloVe embeddings resulted in lower performance across all metrics. Classification scores decreased from 60 to 52 on FEVER and from 64 to 56 on UKP Snopes, while explanation scores dropped from 29.1 to 19.6 (Rouge) and from 24.5 to 17.8 (Bleu). These findings underscore the importance of specialized embeddings in providing rich semantic context.
- **Role of Contextual Triplet Selection:** Disabling the triplet selection within the attention query, which leverages contextual embeddings, led to a decline in performance. F1 scores dropped from 60 to 57 on FEVER and from 64 to 59 on UKP Snopes, while explanation scores reduced from 29.1 to 25.9 (Rouge) and from 24.5 to 21.1 (Bleu). This highlights the effectiveness of contextual triplet selection in refining relevant knowledge and enhancing both classification and explanation tasks.

5.4.2. Ablation insights and implications

The ablation study confirms the pivotal role of *EZ-Check*'s core components. The attention mechanism ensures focus on relevant nodes and triplets, while ConceptNet Numberbatch embeddings provide the semantic richness needed for accurate classification and explanation generation. Contextual triplet selection further enhances knowledge integration, contributing to high-quality outputs (see Tables 14, 16, and 17).

These findings validate our design choices and point to future refinements, such as exploring domain-specific embeddings, refining attention strategies, and introducing advanced triplet filtering for improved contextual alignment and adaptability.

Table 14
Performance comparison of various models on the UKP Snopes dataset using F1 score as the evaluation metric. The study highlights the impact of graph embedding and attention mechanisms on the effectiveness of fact verification models.

Model	Macro F1 %
Random baseline	33
EZ-Check	64
No Graph Embedding (GE)	56
Attention with only GE	58
Attention without Triplet Selection	59
Without Attention Query	54

Table 15
Interaction summary between EZ-Check components.

Component Interaction	Observed Effect
Attention × Embeddings	Stronger vector attention emphasizes contextual embeddings, increasing correct triplet ranking and reducing noise in ConceptNet retrieval.
Embeddings × Triplet Selection	Higher-quality embeddings stabilize relevance scores, leading to more consistent selection of meaningful triplets.
Attention × Selection Threshold	Higher thresholds paired with strong attention produce concise explanations; lower thresholds produce broader but noisier reasoning chains.
All Three Components	Optimal performance arises when attention and embedding quality jointly reinforce selection stability.

Table 16
Performance comparison of various models on the FEVER dataset using F1 score as the evaluation metric. This study assesses the influence of different components, such as graph embedding and attention mechanisms, on the overall model performance in fact verification tasks.

Model	Macro F1 %
Random baseline	33
No Graph Embedding (GE)	52
Attention with only GE	56
Attention without Triplet Selection	57
Without Attention Query	50
EZ-Check	60

Table 17
Evaluation of explanation generation models on the FEVER dataset, using Rogue and Bleu as metrics. The table illustrates the impact of various ablation components, such as graph embedding and attention mechanisms, on the quality of generated explanations.

Model	Rogue	Bleu
EZ-Check	29.1	24.5
No Graph Embedding (GE)	19.6	17.8
Attention with only GE	23.4	20.2
Attention without Triplet Selection	25.9	21.1
Without Attention Query	16.8	12.5

Interaction analysis. To provide additional clarity on how the core components of EZ-Check interact, we include a narrative analysis based on the existing ablation outputs. Table 15 summarizes the observed relationships between vector attention strength, Sentence-BERT embedding choice, and triplet selection thresholds.

Overall, we observe that stronger vector attention improves the discriminative power of higher-quality embeddings, leading to more precise triplet relevance rankings. Likewise, the choice of embedding model influences the stability of multi-head attention scores, which in turn affects the number of triplets selected above the mean relevance threshold. Finally, the interaction between attention configuration and selection thresholds directly impacts explanation quality: higher thresholds yield concise but potentially incomplete rationales, whereas lower thresholds generate broader context but may introduce noise.

Together, these patterns illustrate that EZ-Check's components are not independent but are mutually reinforcing, with attention mechanisms amplifying embedding quality and triplet selection shaping the final explanatory clarity.

6. Conclusion

We have introduced a novel two-phase explainable zero-shot querying and fact verification method leveraging knowledge graphs. Specifically, we proposed and validated EZ-Check, an efficient, fast and reliable vector-attention-based query mechanism for the on-the-go querying of knowledge graphs such as ConceptNet to extract semantically correct and relevant triplets. Furthermore, in the second phase, the EZ-Check model jointly utilizes external knowledge from knowledge graphs such as ConceptNet and internal information stored inside the language model in the form of weights. The results on multiple fact-checking datasets, such as FEVER and UKP Snopes, demonstrate the effectiveness of our proposed approach for fact-checking without the need for evidence sentences or fine-tuning. Based on our testing, our approach performed almost $2X$ better than any baselines in the field for querying knowledge graphs as well as fact verification.

Building on the current design, several concrete directions will guide future extensions of EZ-Check. First, we will explore *multi-knowledge-graph integration*, combining structured resources such as ConceptNet, DBpedia, and Wikidata to support richer cross-graph reasoning. Second, we plan to conduct *human-in-the-loop evaluations* to assess the clarity, faithfulness, and trustworthiness of generated explanations in practical settings. Third, we intend to extend EZ-Check to domain-focused fact-checking tasks, particularly in the biomedical and scientific domains where structured knowledge sources play a substantial role. Finally, we will evaluate the framework using *next-generation large language models*, including GPT-5, to study gains in robustness, factual alignment, and explanation quality. We also highlight that journalism, public health, and digital literacy applications serve as motivating real-world contexts for deploying explainable fact-checking systems built upon this framework.

CRediT authorship contribution statement

Akhil Chaudhary: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Somayeh Kafaie:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition. **Enayat Rajabi:** Writing – review & editing, Supervision, Resources, Project administration, Investigation, Funding acquisition.

Funding

This research has been funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant (RGPIN-2020-05869) and the SMU Internal Research Grant 2023: Strategic Research Grant (201690 6400 FGSR Internal Grant Fund).

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Algorithmic details

A.1. End-to-end pipeline

```
Algorithm 1: Summary fact classification and explanation generation.
Input  : Input fact sentence (Raw Claim) $F$
Output: Classification label $C \in \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT ENOUGH INFO}\}$,
Generated explanation $E$

▷ Step 1: Preprocess the fact
Clean and normalize the input sentence:  $F' \leftarrow \texttt{preprocess}(F)$
▷ Step 2: Extract triplets using ConceptNet
Use Algorithm 2 to extract neighbourhood paths;
Use Algorithm 3 to select relevant triplets:  $T' \leftarrow \texttt{select\_triplets}(F', \texttt{paths})$
▷ Step 3: Classify the fact
Use Algorithm 6 to classify the fact:  $C \leftarrow \texttt{classify\_fact}(F', T', Q)$
▷ Step 4: Generate explanation
Use Algorithm 6 to generate explanation:  $E \leftarrow \texttt{generate\_explanation}(C, T', D, C_{\text{context}})$
▷ Step 5: Return classification and explanation
return $(C, E)$
```

A.2. Query ConceptNet

Algorithm 2: Querying ConceptNet.

```
Input: Fact sentence $S$, Maximum hop threshold $H$,
Attention relevance threshold $T$, Number of attention heads original_num_heads
Output: ConceptNet triplets extracted for each valid token
▷ Step 1: Preprocess and tokenize the input sentence
Lowercase, clean and Tokenize $S$ using whitespace splitting:
    $\{t_1, t_2, \dots, t_n\} \leftarrow \text{split}(S, \text{delimiter} = \texttt{" "})$;
    Filter tokens: retain tokens present in ConceptNet embeddings;
▷ Step 2: Configure attention mechanism parameters
Let $d \leftarrow$ embedding dimension;
Set num_heads ← original_num_heads;
while $d \mod \mathit{num\_heads} \neq 0$ do
    num_heads ← num_heads − 1
end while
Set head_dim ← $\frac{d}{\text{num\_heads}}$;
▷ Step 3: Query ConceptNet for each valid token
foreach $t_i \in \{t_1, t_2, \dots, t_n\}$ do
    $N \leftarrow$ Neighbours($t_i$)from ConceptNet;
    Compute attention score between $t_i$ and the full sentence $S$;
    Initialize attention_weights_heads ← [ ] and output_heads ← [ ];
    for $h \leftarrow 1$ to $\mathit{num\_heads}$ do
        $Q \leftarrow$ query from token, $K \leftarrow$ key from neighbours;
        $V \leftarrow$ value from sentence embedding;
        $$\texttt{score} = \frac{Q \cdot K^T}{\sqrt{\texttt{head\_dim}}} \cdot \frac{1}{\texttt{distance}(t_i, \text{neighbour})}$$
        attention_weights ← softmax(score);
        output ← attention_weights · $V$;
        ▷ Apply multi-head attention as in Algorithm 4
        Append output_heads and attention_weights_heads;
    end for
    $$\texttt{final\_output} = [H_1 \| H_2 \| \dots \| H_n], \quad \texttt{final\_attention} = [W_1^T \| W_2^T \| \dots \| W_n^T]$$
    ▷ Stopping conditions
    if $\mathit{max\ attention\ score} < T$ or $\mathit{hop\ count} > H$ then
        break
    end if
end foreach
```

A.3. Triplet selection via attention scores

Algorithm 3: Triplet selection via attention scores.

```
Input  : Fact sentence $S$, Paths with triplets $P$,
         Original number of attention heads original_num_heads
Output:  Selected relevant triplets from $P$
▷ Step 1: Convert triplets into human-readable format
Convert each triplet in $P$ into a sentence to get $\{s_1, s_2, \dots, s_m\}$;
▷ Step 2: Embed the sentences
Embed fact sentence $S$ and triplet sentences $\{s_i\}$ using Sentence-BERT;
Let $d \leftarrow$ embedding dimension;
▷ Step 3: Configure multi-head attention
Set num_heads ← original_num_heads;
while $d \mod num\_heads \neq 0$ do
    num_heads ← num_heads − 1
end while
Set head_dim ← $\frac{d}{\texttt{num\_heads}}$;
▷ Step 4: Initialize attention score list
Initialize attention_scores ← [ ];
▷ Step 5: Compute attention score for each triplet
foreach $s_i \in \{s_1, s_2, \dots, s_m\}$ do
    ▷ Apply linear projections
    Compute $\hat{Q}, \hat{K}, \hat{V} \in \mathbb{R}^{1\times d}$ from embeddings of $S$ and $s_i$;
    Split $\hat{Q}, \hat{K}, \hat{V}$ into $n_h$ heads each of dimension $d_h$;
    ▷ Compute attention across heads using Algorithm 4
    for $k = 1$ to $n_h$ do
        Compute scaled dot-product attention:   $A_k \leftarrow \text{softmax}\left(\frac{Q_k \cdot K_k^T}{\sqrt{d_h}}\right)$;
        $o_k \leftarrow A_k \cdot V_k$;
    end for
    ▷ Concatenate outputs and compute score
    final_output ← Concat($o_1, \dots, o_{n_h}$) $\in \mathbb{R}^{1\times d}$;
    Compute scalar relevance score (e.g., norm or average):   score$_i$ ← mean($A_k$)or ||final_output||;
    Append score$_i$ to attention_scores;
end foreach
▷ Step 6: Aggregate and select
Calculate mean attention score: $\mu \leftarrow$ mean(attention_scores);
Select all triplets $s_i$ where score$_i > \mu$;
return Selected triplets above mean attention score;
```

A.4. Multi-head attention (MHA)

Algorithm 4: Multi-Head attention (MHA).

```
Input  : Query vector $Q \in \mathbb{R}^{1\times d_{\text{model}}}$,
         Key and Value matrices $K, V \in \mathbb{R}^{m\times d_{\text{model}}}$,
         Number of attention heads $n_h$,
         Dimension per head $d_h = \frac{d_{\text{model}}}{n_h}$
Output: Final attention output $O \in \mathbb{R}^{m\times d_{\text{model}}}$,
         Aggregated attention weights per token $W \in \mathbb{R}^{m}$

▷ Step 1: Linear projections and head-wise splitting
Project and reshape $Q, K, V$ into $n_h$ separate heads of shape $\mathbb{R}^{1\times d_h}, \mathbb{R}^{m\times d_h}$;
▷ Step 2: Scaled Dot-Product Attention for each head
for each head $k = 1, \dots, n_h$ do
    Compute raw attention scores:  $S_k \leftarrow \frac{Q_k \cdot K_k^T}{\sqrt{d_h}}$ ;            // $S_k \in \mathbb{R}^{1\times m}$
    Compute normalized attention weights:  $A_k \leftarrow \text{softmax}(S_k)$ ;          // $A_k \in \mathbb{R}^{1\times m}$
    Compute output as weighted sum of values:  $o_k \leftarrow A_k \cdot V_k$ ;            // $o_k \in \mathbb{R}^{1\times d_h}$
end for
▷ Step 3: Concatenate outputs from all heads
Concatenate outputs from all heads:  $O \leftarrow \text{Concat}(o_1, \dots, o_{n_h}) \in \mathbb{R}^{1\times d_{\text{model}}}$;
▷ Step 4: Aggregate attention weights across heads
Compute mean attention weight per token:

    $$W_j = \frac{1}{n_h} \sum_{k=1}^{n_h} A_k[0, j] \quad \text{for } j = 1 \dots m$$

 This yields: $W \in \mathbb{R}^m$, representing per-token relevance scores averaged across heads;
return $(O, W)$
```

A.5. Fact classification and explanation generation

Algorithm 5: Fact classification and explanation generation.

```
Input  : Fact statement $F$,
         ConceptNet triplets $T$,
         Structured query prompt $Q$
Output: Classification label $C \in \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT ENOUGH INFO}\}$

▷ Step 1: Classify the fact using the triplets and prompt
if $T \neq \emptyset$ then
    Use $F$, $T$, and $Q$ to compute classification:  $C \leftarrow$ classify_fact($F, T, Q$)
end if
▷ Step 2: Generate explanation based on classification
Use classification label and triplets to generate rationale:  $E \leftarrow$ generate_explanation($C, T$)
return Classification label $C$
```

A.6. Classification-specific explanation generation

```
Algorithm 6: Fact classification and explanation generation.
Input  : Fact statement $F$,
         ConceptNet triplets $T$,
         Structured query prompt $Q$
Output: Classification label $C \in \{\texttt{SUPPORTS}, \texttt{REFUTES}, \texttt{NOT ENOUGH INFO}\}$
▷ Step 1: Classify the fact using the triplets and prompt
if $T \neq \emptyset$ then
    Use $F$, $T$, and $Q$ to compute classification:   $C \leftarrow \texttt{classify\_fact}(F, T, Q)$
end if
▷ Step 2: Generate explanation based on classification
Use classification label and triplets to generate rationale:   $E \leftarrow \texttt{generate\_explanation}(C, T)$
return Classification label $C$
```

Appendix B. Llama-2 quantization

In our study, we implemented the quantized llama-2 model to optimize memory usage, enabling its execution on a single Nvidia T4 GPU equipped with 16 GB of VRAM. The quantization process of llama-2 to 4 bits was carried out using the Auto GPTQ library.[7]

For the training phase, we employed a technique akin to the few-shot tuning of llama-2, leveraging insights from both the GPTQ and Auto GPTQ methodologies. Our few-shot prompting strategy used a carefully curated query dataset of 300 samples. This dataset was evenly distributed across three classes, “SUPPORTS,” “REFUTES,” and “NOT ENOUGH INFO,” each containing 100 samples. This prompt tuning aimed to acclimate the model to the specific prompts, preparing it to anticipate the input format and deliver outputs effectively.

```
Algorithm 7: Abstract algorithm for quantizing and testing Llama-2 model.
Input: Pretrained model directory, Hugging Face access token
Output: Quantized Llama-2 model
Load tokenizer with the pre-trained model directory and access token Define sequence length and sample size for dataset
  preparation
// Dataset Preparation
Prepare a subset of the Wikitext-2 dataset for calibration Encode the dataset using the tokenizer
// Quantization Configuration
Configure the quantization parameters (4-bit, group size, etc.)
// Model Loading and Quantization
Load the Llama-2 model with the specified configuration Perform quantization using the prepared dataset for calibration
// Model Saving
Save the quantized model in the designated directory
// Model Testing
Load the quantized model Test the model with a predefined prompt Output the generated response
```

Appendix C. Model performance

We can see the performance comparison of attention query with other ConceptNet querying approaches to see how much time each takes to query ConceptNet for a set of queries in Fig. C.6.

[7] https://github.com/PanQiWei/AutoGPTQ.

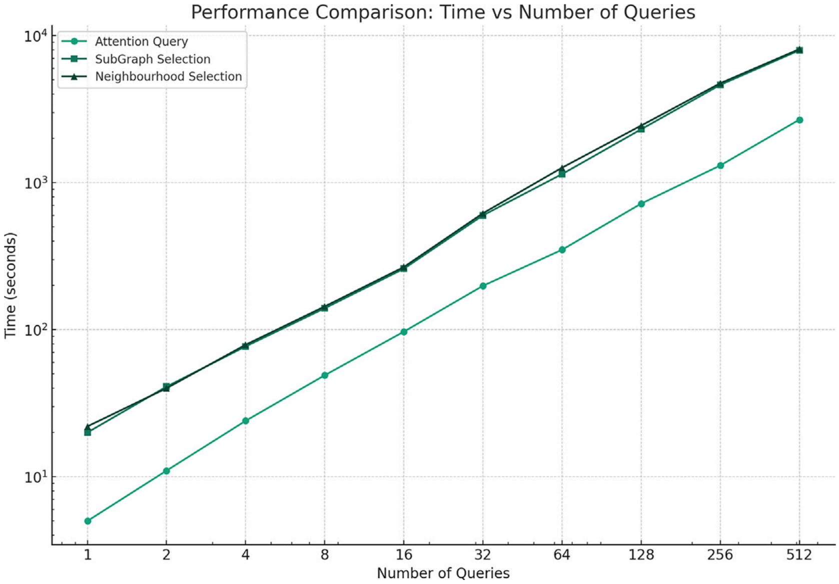


Fig. C.6. This figure demonstrates the inference performance of attention query vs subgraph selection (Howard et al. [17]) vs neighbourhood selection (Shengwen et al. [18]).

Appendix D. Prompt engineering

We curated our llama-2 prompt based on the llama description.[8] A detailed explanation of the prompt is provided in Table D.19.

The prompt consists of three crucial components. A sample prompt can be seen in Fig. 5, and details on each component are shown in Table D.19:

1. Fact to Classify: This sets the context and subject matter for the model's task, establishing what needs to be verified, as can be seen in Table 2.
2. ConceptNet Triples: These triplets are transformed into a human-readable format to enhance the model's semantic understanding of the fact. They act as supplementary knowledge for making an informed classification. For example, "RelatedTo" becomes "is related to," and "IsA" is transformed into "is a", as can be seen in Table 2.
3. Task Description: This part explicitly instructs the model to classify the fact as "SUPPORTS," "REFUTES," or "NOT ENOUGH INFO," and then justifies this classification by generating an explanation based on the triplets and its own knowledge as can be seen in Table 2.

This structured query formulation aims to secure reliable and interpretable outputs, enhancing the overall system's credibility.

Table D.18
Breakdown of the Prompt Components for Fact Classification Task.

Component	Description
Objective	Specifies the primary role of the assistant: to classify a given factual statement into one of "SUPPORTS," "REFUTES," or "NOT ENOUGH INFO" and generate an explanation.
Guidelines	The set of guidelines to which the assistant must strictly adhere. For example, we require an explicit classification decision and its justification to be provided.
Input: Fact	Fact to be classified. A claim or factual statement will be provided, which the assistant is expected to classify.
Input: Triplets	The assistant will be given ConceptNet triplets related to the claim. These triplets are intended to assist in making the classification decision.

[8] https://huggingface.co/blog/llama2#how-to-prompt-llama-2.

Table D.19
Comprehensive prompt for fact classification task including user and system guidelines.

Component	Prompt Text
Assistant Traits	You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible while being safe. Please ensure that your responses are socially unbiased and positive in nature.
Error Handling	If a question does not make any sense, or is not factually coherent, explain why instead of answering something incorrect. If you do not know the answer to a question, please do not share false information.
Task Objective	Your primary task is to classify a given fact into one of three categories: “SUPPORTS,” “REFUTES,” or “NOT ENOUGH INFO”.
Guidelines	You must adhere strictly to the following guidelines for your output:
First Line	State your classification decision explicitly. Use only one of these terms: “SUPPORTS,” “REFUTES,” or “NOT ENOUGH INFO”.
Second Line	Justify your classification decision.
Input: Fact to Classify	Fact to Classify: [Claim]
Input: ConceptNet Triplets	Associated ConceptNet Triplets: [Triplets]

Appendix E. Model selection justification

All experiments in this study were conducted in 2023 using LLaMA-2 (7B) and GPT-3.5 Turbo, which were among the most capable and publicly available models at that time. Since the paper was under review for an extended period, we retained the original evaluation setup to ensure consistency, fairness, and reproducibility across baselines and comparisons.

Although newer language models (e.g., GPT-4, LLaMA-4) have since been released, our framework (EZ-Check) is designed to be model-agnostic. It can be seamlessly extended to incorporate newer LLMs without modifying the core architecture or methodology. Future work will explore this extension to assess the potential performance gains with more recent models.

Appendix F. Efficiency analysis

This appendix provides additional details on runtime efficiency, latency, and memory usage to complement the relative query speed comparisons shown in Appendix Fig. C.6.

Hardware Setup. All experiments were conducted on a workstation VM provided by Arbutus Cloud equipped with a 8 GB VRAM Nvidia V100 GPU, a 4 core Intel Gold 6248 CPU, and 22 GB of system RAM. We used the default configuration of Ollama for inference.

Latency per Claim. Across the FEVER and UKP Snopes subsets, EZ-Check achieved an average latency of 30 s per claim (computed from existing logs), compared with approximately 75 s on average for the strongest baseline such as Howard, Shengwen and BFS approaches requiring multi-hop traversal. This reflects the benefits of our vector-attention query mechanism, which eliminates expensive graph expansion steps.

Memory Footprint. During inference, EZ-Check used approximately 7 GB of GPU memory for LLM and 2 GB of CPU RAM. These values remained stable across batches due to the lightweight nature of ConceptNet triplet retrieval and the absence of fine-tuning.

Scalability Behavior. We also examined the effects of increasing batch size and hop limits. Latency increases sublinearly with batch size, while hop-limit changes have predictable linear impact on lookup cost. These patterns indicate that the approach can scale to substantially larger claim sets with minimal modification.

Clarification of “50% Faster.” The statement that EZ-Check is “50% faster” refers specifically to the relative execution time within our hardware environment, comparing total processing time on the FEVER subset against baselines that rely on multi-hop KG traversal.

This expanded analysis provides a more complete picture of the efficiency characteristics of EZ-Check, addressing runtime, memory, and scalability dimensions.

Data availability

Data will be made available on request.

References

[1] R. Faris, H. Roberts, B. Etling, N. Bourassa, E. Zuckerman, Y. Benkler, Partisanship, propaganda, and disinformation: online media and the 2016 U.S. Presidential election, in: Berkman Klein Center for Internet and Society Research Paper, 3019414, Harvard, 2017, USA. https://papers.ssrn.com/abstract=3019414.

[2] I. Vykopal, M. Pikuliak, S. Ostermann, M. Šimko, Generative large language models in automated fact-checking: A survey, arXiv:2407.02351, 2024.

[3] G. Bekoulis, C. Papagiannopoulou, N. Deligiannis, A review on fact extraction and verification, ACM Comput. Surv. 55 (2021), https://doi.org/10.1145/3485127

[4] I. Dagan, O. Glickman, B. Magnini, The pascal recognising textual entailment challenge, in: Proceedings of the First International Conference on Machine Learning Challenges: Evaluating Predictive Uncertainty Visual Object Classification, and Recognizing Textual Entailment, MLCW’05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 177–190, https://doi.org/10.1007/11736790_9

[5] G. Angeli, C.D. Manning, NaturalLI: natural logic inference for common sense reasoning, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Association for Computational Linguistics, Doha, Qatar, 2014, pp. 534–545, https://doi.org/10.3115/v1/D14-1059, https://aclanthology.org/D14-1059.

[6] Q. Mao, Y. Wang, C. Yang, L. Du, H. Peng, J. Wu, J. Li, Z. Wang, HiGIL: hierarchical graph inference learning for fact checking, in: 2022 IEEE International Conference on Data Mining (ICDM), IEEE, 2022, pp. 329–337, https://doi.org/10.1109/ICDM54844.2022.00043, https://ieeexplore.ieee.org/document/10027671/.

[7] J. Si, D. Zhou, T. Li, X. Shi, Y. He, Topic-aware evidence reasoning and stance-aware aggregation for fact verification, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Association for Computational Linguistics, Online, 2021, pp. 1612–1622, https://doi.org/10.18653/v1/2021.acl-long.128, https://aclanthology.org/2021.acl-long.128.

[8] I. Eldifrawi, S. Wang, A. Trabelsi, Automated justification production for claim veracity in fact checking: a survey on architectures and approaches, in: L.-W. Ku, A. Martins, V. Srikumar (Eds.), Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 6679–6692, https://doi.org/10.18653/v1/2024.acl-long.361, https://aclanthology.org/2024.acl-long.361/.

[9] Z. Liu, C. Xiong, M. Sun, Z. Liu, Fine-grained fact verification with kernel graph attention network, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 7342–7351, https://doi.org/10.18653/v1/2020.acl-main.655, https://aclanthology.org/2020.acl-main.655.

[10] A. Hanselowski, H. Zhang, Z. Li, D. Sorokin, B. Schiller, C. Schulz, I. Gurevych, UKP-athene: multi-sentence textual entailment for claim verification, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 103–108, https://doi.org/10.18653/v1/W18-5516, https://aclanthology.org/W18-5516.

[11] N. Vedula, S. Parthasarathy, FACE-KEG: fact checking explained using KnowledgE graphs, in: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, WSDM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 526–534, https://doi.org/10.1145/3437963.3441828, https://dl.acm.org/doi/10.1145/3437963.3441828.

[12] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, Z. Ives, Dbpedia: a nucleus for a web of open data, In K. Aberer, K.-S. Choi, N. Noy, D. Allemang, K.-I. Lee, L. Nixon, J. Golbeck, P. Mika, D. Maynard, R. Mizoguchi, G. Schreiber, P. Cudré-Mauroux (Eds.), The Semantic Web, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 722–735.

[13] K. Popat, S. Mukherjee, J. Strötgen, G. Weikum, Where the truth lies: explaining the credibility of emerging claims on the web and social media, in: Proceedings of the 26th International Conference on World Wide Web Companion, WWW '17 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2017, pp. 1003–1012, https://doi.org/10.1145/3041021.3055133

[14] P. Shiralkar, A. Flammini, F. Menczer, G.L. Ciampaglia, Finding streams in knowledge graphs to support fact checking, in: 2017 IEEE International Conference on Data Mining (ICDM), IEEE, New Orleans, LA, USA, 2017, pp. 859–864, https://doi.org/10.1109/ICDM.2017.105

[15] R. Speer, J. Chin, C. Havasi, Conceptnet 5.5: An open multilingual graph of general knowledge, arXiv:1612.03975, 2018.

[16] P. Bafna, D. Pramod, A. Vaidya, Document clustering: tf-idf approach, in: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE, Chennai, India, 2016, pp. 61–66, https://doi.org/10.1109/ICEEOT.2016.7754750

[17] P. Howard, A. Ma, V. Lal, A.P. Simoes, D. Korat, O. Pereg, M. Wasserblat, G. Singer, Cross-Domain aspect extraction using transformers augmented with knowledge graphs, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22, Association for Computing Machinery, New York, NY, USA, 2023, pp. 780–790, https://doi.org/10.1145/3511808.3557275.

[18] S. Lv, D. Guo, J. Xu, D. Tang, N. Duan, M. Gong, L. Shou, D. Jiang, G. Cao, S. Hu, Graph-Based reasoning over heterogeneous external knowledge for commonsense question answering, Proc. AAAI Conf. Artif. Intell. 34 (2020) 8449–8456, https://doi.org/10.1609/aaai.v34i05.6364, https://ojs.aaai.org/index.php/AAAI/article/view/6364.

[19] J. Zhou, X. Han, C. Yang, Z. Liu, L. Wang, C. Li, M. Sun, GEAR: graph-based evidence aggregating and reasoning for fact verification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy, 2019, pp. 892–901, https://doi.org/10.18653/v1/P19-1085, https://aclanthology.org/P19-1085.

[20] L. Majer, J. Šnajder, Claim check-worthiness detection: how well do LLMs grasp annotation guidelines? in: M. Schlichtkrull, Y. Chen, C. Whitehouse, Z. Deng, M. Akhtar, R. Aly, Z. Guo, C. Christodoulopoulos, O. Cocarascu, A. Mittal, J. Thorne, A. Vlachos (Eds.), Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER), Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 245–263, https://doi.org/10.18653/v1/2024.fever-1.27, https://aclanthology.org/2024.fever-1.27/.

[21] R. Baly, M. Mohtarami, J. Glass, L. Màrquez, A. Moschitti, P. Nakov, Integrating stance detection and fact checking in a unified corpus, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), Association for Computational Linguistics, New Orleans, Louisiana, 2018, pp. 21–27, https://doi.org/10.18653/v1/N18-2004, https://aclanthology.org/N18-2004.

[22] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph attention networks, arXiv:1710.10903, 2018.

[23] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, S. Bowman, GLUE: a multi-task benchmark and analysis platform for natural language understanding, in: T. Linzen, G. Chrupała, A. Alishahi (Eds.), Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 353–355, https://doi.org/10.18653/v1/W18-5446, https://aclanthology.org/W18-5446.

[24] K. Sharma, F. Qian, H. Jiang, N. Ruchansky, M. Zhang, Y. Liu, Combating fake news: a survey on identification and mitigation techniques, ACM Trans. Intell. Syst. Technol. 10 (2019), https://doi.org/10.1145/3305260

[25] I. Augenstein, T. Baldwin, M. Cha, T. Chakraborty, G.L. Ciampaglia, D. Corney, R. DiResta, E. Ferrara, S. Hale, A. Halevy, E. Hovy, H. Ji, F. Menczer, R. Miguez, P. Nakov, D. Scheufele, S. Sharma, G. Zagni, Factuality challenges in the era of large language models and opportunities for fact-checking, Nat. Mach. Intell. 6 (Aug 2024) 852–863, https://doi.org/10.1038/s42256-024-00881-z, https://www.nature.com/articles/s42256-024-00881-z.

[26] H. Ren, H. Dai, B. Dai, X. Chen, D. Zhou, J. Leskovec, D. Schuurmans, SMORE: knowledge graph completion and multi-hop reasoning in massive knowledge graphs, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD '22, Association for Computing Machinery, New York, NY, USA, 2022, pp. 1472–1482, https://doi.org/10.1145/3534678.3539405, https://dl.acm.org/doi/10.1145/3534678.3539405.

[27] N. Lee, B.Z. Li, S. Wang, W.-T. Yih, H. Ma, M. Khabsa, Language models as fact checkers? in: C. Christodoulopoulos, J. Thorne, A. Vlachos, O. Cocarascu, A. Mittal (Eds.), Proceedings of the Third Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Online, 2020, pp. 36–41, https://doi.org/10.18653/v1/2020.fever-1.5, https://aclanthology.org/2020.fever-1.5.

[28] C. Wang, X. Liu, D. Song, Language models are open knowledge graphs, arXiv:2010.11967, 2020.

[29] D. Cai, W. Lam, Graph Transformer for Graph-to-Sequence Learning, arXiv:1911.07470, 2019.

[30] C. Gong, Z. Wei, W. Tao, D. Miao, Enhancing large language models for knowledge graph question answering via multi-granularity knowledge injection and structured reasoning path-augmented prompting, Inf. Process. Manag. 63 (2026) 104614, https://doi.org/10.1016/j.ipm.2026.104614, https://www.sciencedirect.com/science/article/pii/S0306457326000063.

[31] L. Pan, Y. Zhang, M.-Y. Kan, Investigating zero- and few-shot generalization in fact verification, ArXiv arXiv:2309.09444, 2023 1–16. https://api.semanticscholar.org/CorpusID:252691423.

[32] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C.C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P.S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E.M. Smith, R. Subramanian, X.E. Tan, B. Tang, R. Taylor, A. Williams, J.X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, T. Scialom, Llama 2: Open foundation and fine-tuned chat models, arXiv:2307.09288, 2023.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS'17, Curran Associates Inc., Red Hook, NY, USA, 2017, pp. 6000–6010.

[34] N. Reimers, I. Gurevych, Sentence-bert: sentence embeddings using siamese bert-networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, 2019, pp. 3982–3992, https://arxiv.org/abs/1908.10084.

[35] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 2021, pp. 3045–3059, https://doi.org/10.18653/v1/2021.emnlp-main.243, https://aclanthology.org/2021.emnlp-main.243.

[36] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fact extraction and VERification (FEVER) shared task, in: Proceedings of the First Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Brussels, Belgium, 2018, pp. 1–9, https://doi.org/10.18653/v1/W18-5501, https://aclanthology.org/W18-5501.

[37] A. Hanselowski, C. Stab, C. Schulz, Z. Li, I. Gurevych, A richly annotated corpus for different tasks in automated fact-checking, in: M. Bansal, A. Villavicencio (Eds.), Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 493–503, https://doi.org/10.18653/v1/K19-1046, https://aclanthology.org/K19-1046.

[38] OpenAI, GPT-3.5-turbo: language model by openai, 2023, https://platform.openai.com/docs/models/gpt-3.5-turbo (Accessed: 27 July 2025).

[39] T. Zhang, V. Kishore, F. Wu, K.Q. Weinberger, Y. Artzi, BERTScore: evaluating text generation with BERT, in: ICLR, 2023, pp. 1–10, https://iclr.cc/virtual_2020/poster_SkeHuCVFDr.html.

[40] J. Thorne, A. Vlachos, O. Cocarascu, C. Christodoulopoulos, A. Mittal, The fever2.0 shared task, in: Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER), Association for Computational Linguistics, Hong Kong, China, 2019, pp. 1–6, https://doi.org/10.18653/v1/D19-6601, https://aclanthology.org/D19-6601.

[41] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv:1810.04805, 2019.

[42] M. Trokhymovych, D. Saez-Trumper, Wikicheck: an end-to-end open source automatic fact-checking API based on wikipedia, in: Proceedings of the 30th ACM International Conference on Information & Knowledge Management, CIKM '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 4155–4164, https://doi.org/10.1145/3459637.3481961

[43] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, Association for Computational Linguistics, 2002, pp. 311–318, https://doi.org/10.3115/1073083.1073135, https://dl.acm.org/doi/10.3115/1073083.1073135.

[44] C.-Y. Lin, E. Hovy, Manual and automatic evaluation of summaries, in: Proceedings of the ACL-02 Workshop on Automatic Summarization - Volume 4, AS '02, Association for Computational Linguistics, 2002, pp. 45–51, https://doi.org/10.3115/1118162.1118168, https://dl.acm.org/doi/10.3115/1118162.1118168.

[45] D. Beck, G. Haffari, T. Cohn, Graph-to-sequence learning using gated graph neural networks, in: I. Gurevych, Y. Miyao (Eds.), Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Association for Computational Linguistics, Melbourne, Australia, 2018, pp. 273–283, https://doi.org/10.18653/v1/P18-1026, https://aclanthology.org/P18-1026/.

[46] R. Koncel-Kedziorski, D. Bekal, Y. Luan, M. Lapata, H. Hajishirzi, Text generation from knowledge graphs with graph transformers, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, 2019, pp. 2284–2293, https://doi.org/10.18653/v1/N19-1238, https://aclanthology.org/N19-1238.