ContextClaim: A Context-Driven Paradigm for Verifiable Claim Detection

Anonymous ACL submission

Abstract

Automated fact-checking (AFC) systems typically follow a sequential pipeline comprising four primary stages: (1) claim detection, (2) matching against previously fact-checked claims, (3) evidence retrieval, and (4) claim verification. While research has progressed significantly in the latter stages of the pipeline, claim detection remains a key bottleneck, often relying on subjective heuristics and lacking integration with broader contextual understanding. We introduce Context-Driven Claim Detection (ContextClaim), a novel paradigm that enhances verifiable claim detection by incorporating context retrieval at the initial stage of the AFC pipeline. ContextClaim leverages a knowledge base, such as Wikipedia, to retrieve, aggregate, and filter information about entity mentions, then, generates supplemental context summaries using GPT-4o and Mistral to enrich the assessment process. Our two variants of ContextClaim improve on the verifiable claim detection task over previous state-of-the-art models as well as over ablated versions of ContextClaim. Furthermore, we investigate the generalizability of the paradigm by applying it across both encoder-only and decoder-only language model architectures and in a cross-domain setting. Experimental results consistently show that ContextClaim enhances claim detection performance under most configurations, suggesting its potential for robust and domain-adaptive deployment in real-world misinformation detection systems.

1 Introduction

Automated Fact-Checking (AFC) systems support human efforts by alleviating the burden of manual verification and enhance scalability (Thorne et al., 2018; Zeng et al., 2021). AFC systems are generally composed of modular components, including (i) claim detection, (ii) claim matching against previously fact-checked content, and (iii) claim verification via evidence retrieval and assessment. The design and implementation of these components vary across different research frameworks, but a typical AFC pipeline is illustrated in Figure 1. The initial stages—claim detection and claim matching—are intended to eliminate unverifiable and already-verified claims, thereby streamlining subsequent verification efforts (Shaar et al., 2020). The verification module then focuses on verifying the factual correctness of the remaining claims. Existing approaches to claim detection often rely on subjective criteria such as perceived significance, public interest (Micallef et al., 2022; Das et al., 2023), potential social harm, or attention-worthiness (Shaar et al., 2021; Nakov et al., 2022). Many systems depend heavily on linguistic heuristics and surface-level textual features (Dhar et al., 2019; Favano et al., 2019; Williams et al., 2020; Wührl et al., 2024). Recent work has explored the use of large language models (LLMs), employing in-context learning and fine-tuning to improve claim detection (Sawiński et al., 2023; Li et al., 2024). However, these approaches operate solely on the claim text, without incorporating any form of external context knowledge, which can often be limiting.

To address this limitation, we propose a novel paradigm—Context-Driven Claim Detection (ContextClaim)—which, to the best of our knowledge, is the first to integrate context retrieval into the initial claim detection stage. Rather than retrieving direct evidence for verification, ContextClaim gathers supplementary context from trusted sources, such as Wikipedia, to support the identification of claims as verifiable or unverifiable. The core assumption underlying this approach is that verifiable claims are more likely to align with accessible contextual information, while unverifiable ones exhibit weaker or no alignment. This strategy not only enhances the performance of claim detection but also streamlines downstream verifica-

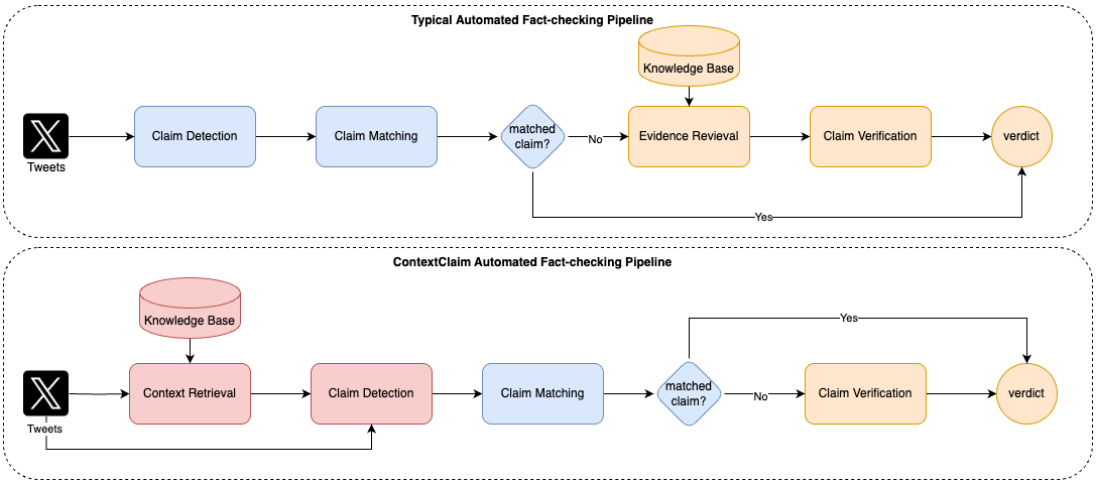

Figure 1: Typical AFC pipeline vs. ContextClaim AFC pipeline.

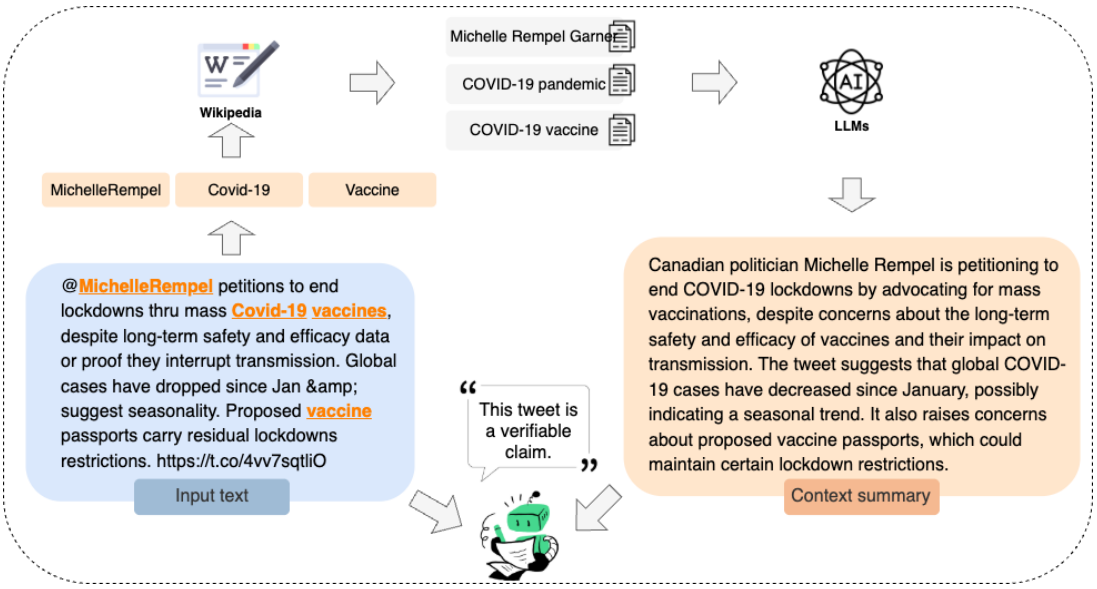

Figure 2: Illustration of the ContextClaim paradigm applied to a specific tweet.

tion by supplying relevant background context in advance.

The ContextClaim paradigm comprises four components, namely entity extraction, context retrieval, context summarization and verifiable claim detection. The AFC pipeline based on this paradigm is outlined in Figure 1, while Figure 2 provides a detailed example illustrating the paradigm in practice.

To assess the efficacy of ContextClaim, we conduct experiments using the CheckThat! 2022 (Nakov et al., 2022) English dataset (hereafter CT22). We evaluate the paradigm across multiple model configurations, including encoder-only and decoder-only architectures, and compare performance across various language models. Our contributions are summarized as follows:

- We develop ContextClaim, a context-driven paradigm for claim detection in automated fact-checking. The paradigm integrates retrieval and re-ranking mechanisms to collect relevant contextual information for a given claim, followed by the generation of a concise summary to support verifiable claim detection.
- We evaluate the effectiveness of ContextClaim through empirical comparison with baseline models that utilize only the claim text. Our model consistently improves over the baseline models in two evaluation sets (dev_test and $test$), demonstrating the effectiveness of our proposed paradigm.
- We further demonstrate the generalizability of ContextClaim through experiments using both encoder-only and decoder-only language mod-

els, as well as in a cross-domain setting. Results consistently indicate that ContextClaim yields improved performance across all configurations.

2 Related Work

2.1 Claim Detection

Claim detection research has traditionally focused on claim check-worthiness estimation (Kartal and Kutlu, 2023), which classifies claims based on criteria such as public importance or interest (Panchendrarajan and Zubiaga, 2024; Micallef et al., 2022; Das et al., 2023). Over time, new prioritization criteria emerged, including detecting harmful or attention-worthy claims (Shaar et al., 2021; Nakov et al., 2022). However, these approaches often rely on subjective judgments, which vary by domain, audience, and context. As a result, recent research has shifted toward verifiable claim detection, defined as an assertion about the world that is checkable (Konstantinovskiy et al., 2021), thus attempting to minimize bias stemming from subjective judgment. This line of work includes efforts to identify opinionated claims from Reddit (Chakrabarty et al., 2019) and to classify statements as either subjective or objective—where objective statements are better suited for verification tasks (Galassi et al., 2023; Struß et al., 2024).

Early claim detection methods relied on feature engineering and traditional machine learning. Systems like ClaimBuster (Hassan et al., 2017) and ClaimRank (Jaradat et al., 2018) used linguistic and structural features with machine learning and neural networks. The CNC system (Konstantinovskiy et al., 2021) applied sentence embeddings from InferSent (Conneau et al., 2017) along with POS and NER features, feeding them into Logistic Regression or SVM classifiers. With advances in deep learning, pretrained language models became central. ULMFiT, fine-tuned on the IMHO dataset, significantly improved domain adaptation for claim detection (Chakrabarty et al., 2019). Early CheckThat! shared tasks used LSTM-based and feedforward models (Dhar et al., 2019; Hansen et al., 2019; Favano et al., 2019), but transformer-based models have dominated since 2020. For example, a fine-tuned RoBERTa model led to a first-place finish in the English track (Williams et al., 2020). More recently, large language models (LLMs) have advanced the field. In 2023, top-performing systems employed GPT-3 in zero-shot and few-shot settings (Sawiński et al., 2023; Alam et al., 2023). In 2024, the winning system fine-tuned eight open-source LLMs for claim detection (Li et al., 2024), demonstrating the growing effectiveness of LLMs in this domain.

2.2 Evidence Retrieval

Evidence retrieval is typically divided into two steps: document retrieval and rationale or sentence selection. This process, often part of the later stages of fact-checking, identifies supporting evidence to assess a claim's veracity (Guo et al., 2022). The FEVER benchmark (Thorne et al., 2018) was an early effort to incorporate information extraction into claim verification, followed by tasks like the Evidence and Factuality track (Elsayed et al., 2019), which focused on retrieving relevant content for factuality assessment.

Initial studies integrating document retrieval into their models showed performance gains (Soleimani et al., 2020). Later work improved results by combining traditional retrieval methods such as TF-IDF (Ramos et al., 2003) and BM25 (Robertson et al., 2009) with neural architectures (Hanselowski et al., 2018). More recently, generative approaches like GERE (Chen et al., 2022) introduced efficient evidence retrieval to reduce computational cost and select relevant evidence dynamically. RAV (Zheng et al., 2024) proposed a hybrid approach, combining retrieval with joint verification. With the rise of Large Language Models (LLMs), retrieval-augmented generation (RAG) has emerged as a strategy to integrate external knowledge without retraining, enabling models to generate text grounded in retrieved content. RARG (Yue et al., 2024) extends this by assembling scientific evidence and applying reinforcement learning from human feedback (RLHF) for response generation.

Our research draws inspiration from the role of evidence retrieval in supporting claim verification within these established frameworks. We reframe this process as contextual information retrieval (context retrieval) to facilitate the filtering of tweets containing verifiable claims. This approach seeks to enhance the efficacy and efficiency of claim detection while potentially optimizing subsequent claim verification processes by reducing the volume of claims requiring verification.

3 Methodology

We introduce ContextClaim, a context-driven paradigm designed to enhance claim detection by leveraging contextual information from Wikipedia. The paradigm operates through a sequence of components: (1) entity extraction: Given an input tweet x_i, the paradigm first identifies a set of named entities $E_i = \{e_1, e_2, ..., e_m\}$, then (2) context retrieval: For each extracted entity, the system retrieves relevant information from Wikipedia, selecting the most pertinent extracts a_i. These extracts are then aggregated and filtered to construct a comprehensive knowledge base K_i. (3) Context Summarization: The knowledge base K_i is combined with the original tweet to generate a context summary c_i as supplemental information in claim detection. (4) Verifiable claim detection: Finally, both the original tweet x_i and the context summary c_i are input into a fine-tuned language model, which classifies whether the tweet contains a verifiable claim.

3.1 Entity Extraction

Entities in a text often carry the most important information. By extracting these entities, we can convert unstructured input into a more structured form, facilitating the subsequent context retrieval. Instead of relying on general keywords, we specifically use a BERT-based named entity recognition (NER) model fine-tuned to identify entities with four standard types (Devlin et al., 2018): Person (PER), Location (LOC), Organization (ORG), and Miscellaneous (MISC). To address the limitations of standard NER models in recognizing COVID-19-specific entities, we use a word cloud algorithm (Mueller, 2014) to identify frequent and contextually relevant terms in the dataset. These insights allow us to define additional popular topic-related keywords manually, enhancing entity extraction and improving the effectiveness of the retrieval stage.

Formally, let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of input texts. For each text x_i, named entity recognition (NER) identifies a set of entities $E_i = \{e_{i,1}, e_{i,2}, \ldots, e_{i,m_i}\}$, where each entity $e_{i,j}$ is a tuple $(w_{i,j}, t_{i,j})$. Here, $w_{i,j}$ is the entity token and $t_{i,j} \in T$ is its type, with $T = \{\text{PER}, \text{LOC}, \text{ORG}, \text{MISC}, \text{TOPIC}\}$ denoting the set of possible entity types.

3.2 Context Retrieval

For each extracted entity $e_{i,j}$, we use the MediaWiki Action API[1] to retrieve the top five relevant article extracts:

$$A_{i,j} = \{a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,5}\}$$

To rank these extracts by usefulness, we compute a relevance score that combines two factors: (1) how closely the extract matches the original input text x_i, and (2) how well the Wikipedia article title aligns with the entity $w_{i,j}$. Both are measured using cosine similarity between sentence embeddings produced by a sentence transformer (Wang et al., 2020). The final score is a weighted sum:

$$\text{score}(a_{i,j,k}, x_i) = \alpha \cdot f(a_{i,j,k}, x_i) + \beta \cdot f(\text{title}_k, w_{i,j})$$

where $\alpha = 0.8$ and $\beta = 0.2$ are weights tuned on the dev_test set, and f denotes cosine similarity.

We select the extract with the highest score as the most relevant context for the entity:

$$a^*_{i,j} = \arg \max_{a_{i,j,k} \in A_{i,j}} \text{score}(a_{i,j,k}, x_i)$$

Repeating this for all entities in x_i, we obtain a set of top-ranked extracts:

$$A^*_i = \{a^*_{i,1}, a^*_{i,2}, \ldots, a^*_{i,m_i}\}$$

We then apply a filtering step to remove low-quality entity-extract pairs. Specifically, we retain entities classified under $T_{\text{valid}} = \{\text{PER}, \text{LOC}, \text{ORG}\}$, which consistently yield high-quality extracts, while discarding low-relevance extracts associated with entities from the broader TOPIC category. The remaining extracts define the contextual knowledge base K_i for the input:

$$\hat{A}_i = \{a^*_{i,j} \in A^*_i \mid \text{score}(a^*_{i,j}, x_i) \geq \theta \wedge t_{i,j} \in T_{\text{valid}}\}$$

$$K_i = \bigcup_{a^*_{i,j} \in \hat{A}_i} a^*_{i,j}$$

This filtered set K_i provides the contextual knowledge base used in the next processing stage.

3.3 Context Summarization

Using the contextual knowledge base K_i from the previous step, we generate a context summary c_i for each input tweet x_i via a generation function g:

$$c_i = g(K_i)$$

[1] https://www.mediawiki.org/wiki/API:Main_page

Summarization Prompt
You are a helpful assistant. Provide a factual summarization under 150 words. Tweet: "{clean_tweet}" Relevant Context: {all_extracts} Generate a concise, objective summary to the provided tweet based ONLY on the provided context.

Table 1: Prompt for context summarization.

We adopt a prompt-based summarization approach, with predefined instructions (shown in Table 1) guiding the models to generate contextually relevant summaries. To compare model performance, we evaluate two language models: GPT-4o (Achiam et al., 2023), a state-of-the-art instruction-following model from OpenAI, and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), a lightweight, open-source alternative. Both models are prompted to generate factual summaries under 150 words, using only the content in K_i, to ensure faithfulness and avoid hallucinations. We refer to their outputs as ContextClaim-G4o and ContextClaim-M, abbreviated as CC-G4o and CC-M, respectively, throughout the remainder of this paper.

3.4 Verifiable Claim Detection

In the final step of the ContextClaim paradigm, we assess the verifiability of each claim using the generated context. Given the original input x_i and its corresponding context summary c_i, we feed the pair into a language model that predicts whether the claim can be verified:

$$v_i = h(x_i, c_i)$$

Here, $v_i \in \{0, 1\}$ is a binary label, where $v_i = 1$ indicates that the claim is verifiable based on the context, and $v_i = 0$ indicates that it is unverifiable due to insufficient or ambiguous contextual information.

The classifier can be implemented using various architectures (see Section 4.2). This step completes the ContextClaim workflow, linking entity extraction, evidence retrieval, and summarization to a final, context-based verifiability decision.

4 Experiment

4.1 Dataset

Our experiments utilize the CT22 dataset, which contains 4793 English-language COVID-19 tweets annotated as either verifiable (1) or unverifiable (0) claims. The dataset is divided into four subsets: $train$, dev, dev_test, and $test$. For all experiments, we employ the $train$ and dev sets for training and validation, while utilizing the dev_test and $test$ sets as independent evaluation datasets. After preprocessing—which includes removing URLs, user mention and hashtag symbols, converting emojis, and removing stopwords, along with lemmatization using NLTK—these tweets average around 20 words in length, with tweet lengths ranging from 0 to 73 words.

4.2 Models

To assess the effectiveness of the ContextClaim paradigm, we evaluate both encoder-only and decoder-only models. For encoder-only models, we use BERT-base (Devlin et al., 2018) and RoBERTa-large (Liu et al., 2019). For decoder-only models, we test two recent open-source LLMs: Llama-3-8B-Instruct (AI@Meta, 2024) and Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), chosen for their strong performance across language tasks. This setup enables a direct comparison between encoder-only and decoder-only architectures for the verifiable claim detection task.

4.2.1 Baseline Models

Models that take only the tweet text as input serve as our baselines: BL_{BERT}, $\text{BL}_{\text{RoBERTa}}$, $\text{BL}_{\text{Llama3}}$, and $\text{BL}_{\text{Mistral}}$. For BERT and RoBERTa baselines, we use standard tokenization with '[CLS]' and '[SEP]' tokens, and the '[CLS]' representation is used for classification. Inputs are padded or truncated to 128 tokens. We add a learnable attention layer on top of the final hidden states to help the model focus on key parts of the tweet. These models are fine-tuned for verifiability detection. For Llama3 and Mistral, we use the default tokenization from their Hugging Face implementations. Inputs are formatted as baseline prompts (see Table 2), and no special tokens are manually inserted, as the models handle formatting internally.

Baseline Prompt
Instruction: Determine if this tweet contains verifiable claims. If it contains claims that can be verified, respond "Yes". Otherwise, respond "No". Note: When in doubt, choose "Yes". In the end, respond only with 'Yes' for verifiable claims or 'No' for unverifiable claims. ### Input tweet: {tweet_text} ### Response: {Yes/No}

Table 2: Baseline prompt for verifiable claim detection.

4.2.2 ContextClaim Models

Models that utilize both the tweet and its contextual information implement the full ContextClaim, denoted as CC_{BERT}, $CC_{RoBERTa}$, CC_{Llama3}, and $CC_{Mistral}$. Depending on the context generator, we label these as CC-G4o (using GPT-4o) or CC-M (using Mistral). For CC_{BERT} and $CC_{RoBERTa}$, we use the same tokenization as in the baselines. The tweet and context are first encoded separately, then integrated using a multi-head cross-attention mechanism—where the tweet acts as the query and the context as the key and value. This allows the model to focus on context elements most relevant to the claim. Outputs are then fused for final classification. For CC_{Llama3} and $CC_{Mistral}$, we extend the baseline prompt format to include the tweet and its context, forming a dual-prompt input (see Table 3).

ContextClaim Prompt

```
### Instruction:
Determine if this tweet contains verifiable claims.
Primary analysis:
- Analyze the tweet text first. If it clearly contains
verifiable factual claims, respond "Yes".
- If it clearly contains only opinions or
unverifiable statements, respond "No".
Secondary analysis (only if primary analysis is unclear):
- Reference the additional information to help clarify
the nature of the claims in the tweet.
Note: When in doubt, choose "Yes". In the end,
respond only with 'Yes' for verifiable claims
or 'No' for unverifiable claims.
### Input tweet: {tweet_text}
### Additional information: {contextual information}
### Response: {Yes/No}
```

Table 3: ContextClaim prompt for verifiable claim detection.

4.3 Experimental Settings

All experiments are conducted on an NVIDIA A100 80GB PCIe GPU, using 12 CPU cores with 7.5 GB memory each. The software environment includes CUDA 11.8, PyTorch 2.6.0, and Hugging Face Transformers 4.49.0. To ensure stability and reproducibility, we initialize random states using multiple seeds $[42, 123, 456, 789, 1024]$ for Python, NumPy, PyTorch, and CUDA. A consistent preprocessing pipeline is applied to all tweets, including the removal of special characters (e.g., URLs), normalization of Twitter-specific symbols (like '@' and '#'), whitespace standardization, and emoji-to-text conversion.

Due to the differing nature of encoder-only and decoder-only architectures, we adopt tailored fine-tuning strategies. Encoder-only models use a custom training loop with gradient accumulation for better training control. Decoder-only models (LLMs) are fine-tuned using HuggingFace's 'SFTTrainer', with 4-bit quantization via 'BitsAndBytesConfig' for memory efficiency, and LoRA-based parameter-efficient fine-tuning (PEFT) to reduce training overhead while maintaining performance. Hyperparameters for both model types are tuned separately (see Appendix B).

To evaluate performance, we use F1-score as the primary metric, supported by accuracy, precision, and recall. All metrics are reported on both the 'dev_test' and '$test$' sets to evaluate in-distribution performance and generalization.

5 Results and Discussion

Our experiments evaluate the performance of baseline models using only tweets as input against our proposed ContextClaim paradigm, which incorporates context summaries generated through two different approaches: CC-G4o and CC-M as mentioned in the Section 3.3. We maintain a clear distinction between the original claim and its contextual information. Table 4 presents comprehensive results across models on both dev_test and $test$ sets. Results show that incorporating contextual information through our ContextClaim paradigm generally improves performance over baseline models across most language model configurations, though the degree of improvement varies by model and contextual information source.

5.1 Model Performance Across Architectures and Evaluation Sets

Our analysis reveals consistent trends in model performance across both evaluation sets and architectural types. In general, models exhibit a performance drop when moving from the dev_test set to the more challenging $test$ set, with F1 scores typically declining by 2–8%. This distribution shift suggests that the $test$ set contains more complex or diverse claims. Among the models, CC_{Llama3}-G4o is the most robust, with only a 2.5% drop, whereas CC_{BERT}-M sees a larger decrease of 8.6%. Notably, recall remains more stable across datasets than precision, indicating models are generally more reliable in detecting verifiable claims than in classifying them precisely.

Performance also varies by model architecture. Encoder-only models like BERT and RoBERTa

Model	*dev_test*				*test*			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
BL_{BERT}	0.7908	0.7970	0.8969	0.8438	**0.6964**	0.6960	**0.8658**	**0.7706**
CC_{BERT}-G4o	0.7996	0.8048	**0.9003**	0.8499	0.6948	0.7025	0.8443	0.7667
CC_{BERT}-M	**0.8031**	**0.8117**	0.8951	**0.8514**	0.6956	**0.7064**	0.8362	0.7655
$BL_{RoBERTa}$	0.8083	0.8039	**0.9225**	0.8586	0.6964	0.6891	0.8953	0.7774
$CC_{RoBERTa}$-G4o	0.8114	0.8084	0.9202	0.8602	**0.7243**	0.7108	**0.9034**	**0.7955**
$CC_{RoBERTa}$-M	**0.8182**	**0.8187**	0.9143	**0.8637**	0.7163	**0.7117**	0.8792	0.7864
BL_{Llama3}	0.5529	**0.8460**	0.3558	0.4997	0.5323	**0.7896**	0.2819	0.4122
CC_{Llama3}-G4o	0.6484	0.6485	**0.9652**	0.7757	0.6255	0.6190	**0.9597**	0.7526
CC_{Llama3}-M	**0.6773**	0.6747	0.9419	**0.7862**	**0.6627**	0.6564	0.9060	**0.7613**
$BL_{Mistral}$	**0.7964**	0.8031	**0.8973**	**0.8475**	0.6891	0.6861	**0.8725**	0.7678
$CC_{Mistral}$-G4o	0.7900	**0.8118**	0.8682	0.8389	**0.7490**	**0.7876**	0.7919	**0.7893**
$CC_{Mistral}$-M	0.7746	0.7874	0.8804	0.8310	0.7264	0.7490	0.8121	0.7783

Table 4: Performance comparison of verifiable claim detection models. CC = ContextClaim.

tend to outperform decoder-only models such as Llama3 and Mistral, likely due to their bidirectional attention mechanisms. However, decoder models, especially Mistral when paired with CC-G4o, show competitive results on the *test* set. Llama3, in particular, demonstrates strong improvements from contextual input: recall increases by approximately 60% and F1 scores by 30%, highlighting its ability to utilize additional contextual information. In contrast, models with stronger baselines (e.g., RoBERTa, Mistral) exhibit more modest gains, suggesting that the benefit of added context diminishes as base performance improves.

5.2 The Impact of Context Quality

Our experiments demonstrate that enriching tweets with contextual information significantly improves performance in verifiable claim detection. Further, an investigation into ablated versions of ContextClaim which do not use context or use context only without the claim, as shown in Appendix C.1, demonstrates the overall better performance of the full paradigm. By incorporating context summaries generated by large language models (LLMs) based on Wikipedia content, the task shifts from relying solely on the tweet's linguistic features to leveraging additional context that supports or challenges the verifiability of a claim. To quantify the informational value of these summaries, we employ a Natural Language Inference (NLI) model [2] (Williams et al., 2017) to compute entailment, neutral, and contradiction scores between each tweet and its associated context, using CC-G4o and CC-M, respectively.

[2] https://huggingface.co/FacebookAI/roberta-large-mnli

The entailment score measures how well the context aligns with the original tweet, the neutral score reflects additional information introduced, and the contradiction score indicates semantic conflict. As shown in Figure 3, CC-G4o produces a higher average entailment score (0.53) than CC-M (0.36), indicating that it more faithfully preserves the tweet's content. CC-G4o also displays a bimodal distribution in entailment, suggesting that its contexts are either highly aligned or largely unrelated, while CC-M concentrates around lower scores, implying more frequent addition of loosely related information. In contrast, CC-M shows higher neutral scores, pointing to broader contextual enrichment. Both context types maintain low contradiction scores, demonstrating strong factual consistency. These characteristics reveal a trade-off between precision and coverage: CC-G4o offers more focused, fact-dense context that enhances precision, whereas CC-M provides a wider range of information, which can improve recall. This trade-off is reflected in model performance. On the *dev_test* set, encoder-based models (e.g., BERT and RoBERTa) perform slightly better with CC-M due to its broader coverage. However, on the *test* set, CC-G4o consistently enables better generalization. For example, $CC_{Mistral}$-G4o achieves an F1-score improvement of approximately 1.1% over $CC_{Mistral}$-M. GPT-4o-generated contexts also lead to notable precision gains, particularly for decoder-based models; $CC_{Mistral}$-G4o shows a 10% increase in precision compared to its baseline. Meanwhile, both context types significantly boost recall. Notably, CC_{Llama3}-G4o and CC_{Llama3}-M improve recall from 0.2819 to 0.9597 and 0.9060, respectively, and $CC_{RoBERTa}$-G4o achieves the highest

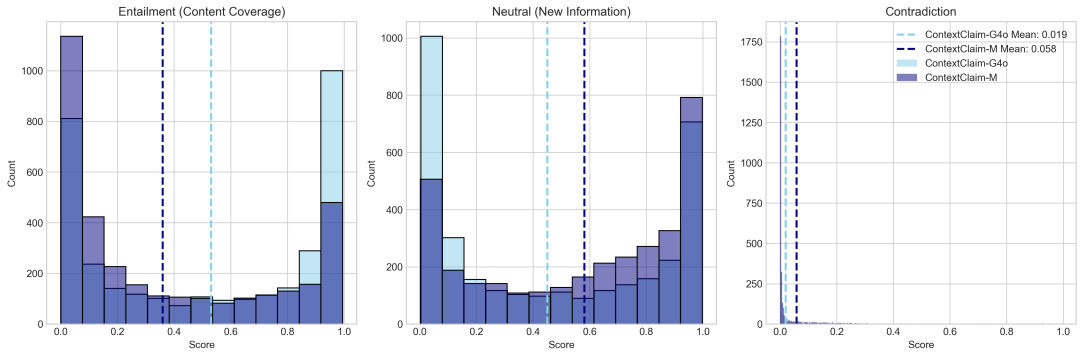

Figure 3: Information gain distribution in CC-G4o and CC-M.

recall score (0.9034) across its variants.

Overall, these results confirm that contextual summaries—especially those with high factual precision from GPT-4o—improve model performance both in terms of precision and recall depending on the context's characteristics.

6 Error Analysis

To better understand the limitations of our ContextClaim paradigm, we conducted a detailed error analysis of the $\text{CC}_{\text{RoBERTa}}$-G4o model using five different random initialization seeds. This multi-seed approach helps us distinguish between consistent model weaknesses and performance variations due to randomness in initialization.

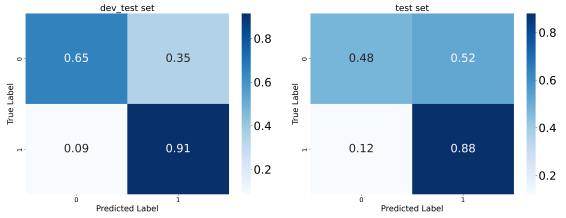

Figure 4: Confusion matrix of $\text{CC}_{\text{RoBERTa}}$-G4o

Our analysis highlights two major issues: (1) a persistent struggle with identifying unverifiable claims, and (2) failures in how the model incorporates contextual information for certain examples. As shown in Figure 4, the model consistently struggles more with unverifiable claims (label 0) compared to verifiable ones. On the *dev_test* set, 35% of unverifiable claims are misclassified, 52% on the *test* set. In contrast, verifiable claims are correctly classified 91% of the time. This imbalance suggests the model has trouble learning what makes a claim unverifiable. Interestingly, the low false negative rate (9%) indicates that when the model does label a claim as unverifiable, it's usually right—suggesting it has selected on some reliable patterns, but not all. Additionally, the same 89 examples in the *dev_test* set and 42 in the *test* set were misclassified across all five seeds, pointing to specific cases that consistently challenge the model, rather than errors caused by random variation. We summarize representative failure cases and the corresponding reasoning behind the model's misclassifications in Appendix C.2. The examples highlight difficulties such as confusion between factual and opinion-based claims, misinterpretation of rhetorical language, and poor handling of references to inaccessible or private information.

7 Conclusion

We present Context-Driven Claim Detection (ContextClaim), a novel paradigm for identifying verifiable claims. To the best of our knowledge, ContextClaim is the first method to incorporate contextual information retrieval from trusted sources to construct a dynamic knowledge base. This knowledge base is subsequently distilled into a concise contextual summary to support the detection of verifiable claims. For context summarization, we employ two large language models—GPT-4o and Mistral—resulting in two variants: CC-G4o and CC-M, respectively. CC-G4o generally demonstrates superior factual precision and denser summarization, attributed to its improved preservation of the original content, such as tweet-specific semantics. Experimental results show that integrating ContextClaim with existing claim detection models leads to substantial performance improvements. Additionally, both encoder-only and decoder-only language models, when augmented with ContextClaim, consistently outperform baseline models that utilize only the raw claim text as input.

Limitations

While ContextClaim shows promise for verifiable claim detection, several limitations remain. First, we do not examine its effectiveness in out-of-domain settings, particularly when test domains differ substantially from COVID-19-related content, which may restrict the method's applicability in more diverse real-world scenarios. Second, the paradigm assumes access to trustworthy knowledge sources; however, in cases where source reliability is uncertain (e.g., when retrieving content via general search engines like Google), the accuracy and consistency of the contextual summaries may be compromised. Lastly, due to practical constraints, including resource limitations and the scope of this study, we have not conducted human evaluations on quality and utility of the generated context. These limitations motivate us to plan to explore a more generalized and domain-adaptive solution supported by a more comprehensive evaluation framework for context-driven claim detection.

Acknowledgments

Omitted for blind review.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

AI@Meta. 2024. Llama 3 model card.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Firoj Alam, Alberto Barrón-Cedeño, Gullal S Cheema, Sherzod Hakimov, Maram Hasanain, Chengkai Li, Rubén Míguez, Hamdy Mubarak, Gautam Kishore Shahi, Wajdi Zaghouani, and 1 others. 2023. Overview of the clef-2023 checkthat! lab task 1 on check-worthiness in multimodal and multigenre content. *Working Notes of CLEF*.

Tuhin Chakrabarty, Christopher Hidey, and Kathleen McKeown. 2019. Imho fine-tuning improves claim detection. *arXiv preprint arXiv:1905.07000*.

Jiangui Chen, Ruqing Zhang, Jiafeng Guo, Yixing Fan, and Xueqi Cheng. 2022. Gere: Generative evidence retrieval for fact verification. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2184–2189.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Anubrata Das, Houjiang Liu, Venelin Kovatchev, and Matthew Lease. 2023. The state of human-centered nlp technology for fact-checking. *Information processing & management*, 60(2):103219.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Rudra Dhar, Subhabrata Dutta, and Dipankar Das. 2019. A hybrid model to rank sentences for check-worthiness. In *CLEF (Working Notes)*.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019. CheckThat! at CLEF 2019: Automatic Identification and Verification of Claims. In Leif Azzopardi, Benno Stein, Norbert Fuhr, Philipp Mayr, Claudia Hauff, and Djoerd Hiemstra, editors, *Advances in Information Retrieval*, volume 11438, pages 309–315. Cham.

L Favano, M Carman, and P Lanzi. 2019. Theearthisflat's submission to clef'19 checkthat. *Challenge. In: Cappellato et al.[8]*.

Andrea Galassi, Federico Ruggeri, Alberto Barrón-Cedeño, Firoj Alam, Tommaso Caselli, Mucahid Kutlu, Julia Maria Struß, Francesco Antici, Maram Hasanain, Juliane Köhler, and 1 others. 2023. Overview of the clef-2023 checkthat! lab: Task 2 on subjectivity in news articles. In *24th Working Notes of the Conference and Labs of the Evaluation Forum, CLEF-WN 2023*, pages 236–249. CEUR Workshop Proceedings (CEUR-WS. org).

Zhijiang Guo, Michael Schlichtkrull, and Andreas Vlachos. 2022. A Survey on Automated Fact-Checking. *Transactions of the Association for Computational Linguistics*, 10:178–206.

Andreas Hanselowski, Hao Zhang, Zile Li, Daniil Sorokin, Benjamin Schiller, Claudia Schulz, and Iryna Gurevych. 2018. Ukp-athene: Multi-sentence textual entailment for claim verification. *arXiv preprint arXiv:1809.01479*.

Casper Hansen, Christian Hansen, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural weakly supervised fact check-worthiness detection with contrastive sampling-based ranking loss. In *CLEF (Working Notes)*.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, Vikas Sable, Chengkai Li, and Mark Tremayne. 2017. ClaimBuster: The first-ever end-to-end fact-checking system. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Israa Jaradat, Pepa Gencheva, Alberto Barron-Cedeno, Lluis Marquez, and Preslav Nakov. 2018. ClaimRank: Detecting Check-Worthy Claims in Arabic and English. *Preprint*, arxiv:1804.07587.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Yavuz Selim Kartal and Mucahid Kutlu. 2023. ReThink Before You Share: A Comprehensive Study on Prioritizing Check-Worthy Claims. *IEEE Transactions on Computational Social Systems*, 10(1):362–375.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2021. Toward Automated Factchecking: Developing an Annotation Schema and Benchmark for Consistent Automated Claim Detection. *Digital Threats: Research and Practice*, 2(2):1–16.

Yufeng Li, Rrubaa Panchendrarajan, and Arkaitz Zubiaga. 2024. Factfinders at checkthat! 2024: refining check-worthy statement detection with llms through data pruning. *arXiv preprint arXiv:2406.18297*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Nicholas Micallef, Vivienne Armacost, Nasir Memon, and Sameer Patil. 2022. True or false: Studying the work practices of professional fact-checkers. *Proceedings of the ACM on Human-Computer Interaction*, 6(CSCW1):1–44.

Andreas Mueller. 2014. wordcloud: A little word cloud generator in python. `https://github.com/amueller/word_cloud`.

Preslav Nakov, Alberto Barrón-Cedeño, Giovanni Da San Martino, Firoj Alam, Rubén Míguez, Tommaso Caselli, Mucahid Kutlu, Wajdi Zaghouani, Chengkai Li, Shaden Shaar, and 1 others. 2022. Overview of the clef-2022 checkthat! lab task 1 on identifying relevant claims in tweets. In *2022 Conference and Labs of the Evaluation Forum, CLEF 2022*, pages 368–392. CEUR Workshop Proceedings (CEUR-WS. org).

Rrubaa Panchendrarajan and Arkaitz Zubiaga. 2024. Claim detection for automated fact-checking: A survey on monolingual, multilingual and cross-lingual research. *Natural Language Processing Journal*, 7:100066.

Juan Ramos and 1 others. 2003. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 29–48. Citeseer.

Stephen Robertson, Hugo Zaragoza, and 1 others. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Marcin Sawiński, Krzysztof Węcel, Ewelina Paulina Księżniak, Milena Stróżyna, Włodzimierz Lewoniewski, Piotr Stolarski, and Witold Abramowicz. 2023. Openfact at checkthat! 2023: head-to-head gpt vs. bert-a comparative study of transformers language models for the detection of check-worthy claims. *Working Notes of CLEF*.

Shaden Shaar, Firoj Alam, Giovanni Da San Martino, Alex Nikolov, Wajdi Zaghouani, Preslav Nakov, and Anna Feldman. 2021. Findings of the nlp4if-2021 shared tasks on fighting the covid-19 infodemic and censorship detection. *arXiv preprint arXiv:2109.12986*.

Shaden Shaar, Giovanni Da San Martino, Nikolay Babulkov, and Preslav Nakov. 2020. That is a known lie: Detecting previously fact-checked claims. *arXiv preprint arXiv:2005.06058*.

Amir Soleimani, Christof Monz, and Marcel Worring. 2020. Bert for evidence retrieval and claim verification. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42*, pages 359–366. Springer.

Julia Maria Struß, Federico Ruggeri, Alberto Barrón-Cedeño, Firoj Alam, Dimitar Dimitrov, Andrea Galassi, Georgi Pachov, Ivan Koychev, Preslav Nakov, Melanie Siegel, and 1 others. 2024. Overview of the clef-2024 checkthat! lab task 2 on subjectivity in news articles. In *CEUR Workshop Proceedings*, volume 3740, pages 287–298. CEUR-WS.

James Thorne, Andreas Vlachos, Oana Cocarascu, Christos Christodoulopoulos, and Arpit Mittal. 2018. The Fact Extraction and VERification (FEVER) Shared Task. *Preprint*, arxiv:1811.10971.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. 2020. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Evan Williams, Paul Rodrigues, and Valerie Novak. 2020. Accenture at checkthat! 2020: If you say so: Post-hoc fact-checking of claims using transformer-based models. *arXiv preprint arXiv:2009.02431*.

Amelie Wührl, Yarik Menchaca Resendiz, Lara Grimminger, and Roman Klinger. 2024. What makes medical claims (un) verifiable? analyzing entity and relation properties for fact verification. *arXiv preprint arXiv:2402.01360*.

Zhenrui Yue, Huimin Zeng, Yimeng Lu, Lanyu Shang, Yang Zhang, and Dong Wang. 2024. Evidence-driven retrieval augmented response generation for online misinformation. *arXiv preprint arXiv:2403.14952*.

Xia Zeng, Amani S. Abumansour, and Arkaitz Zubiaga. 2021. Automated fact-checking: A survey. *Language and Linguistics Compass*, 15(10):e12438.

Liwen Zheng, Chaozhuo Li, Xi Zhang, Yu-Ming Shang, Feiran Huang, and Haoran Jia. 2024. Evidence retrieval is almost all you need for fact verification. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 9274–9281.

A More Details and Analysis of ContextClaim

A.1 ContextClaim Paradigm

Figure 5 shows the complete workflow of the ContextClaim paradigm discussed in Section 3.

A.2 Statistics of CT22 and Retrieved Contextual Information

Table 5 presents the statistics of CT22 English dataset. To study the impact of contextual information in the claim detection task, we enhance the CT22 dataset by attaching retrieved context to each tweet. This is done by applying the first three components of our paradigm—entity extraction, context retrieval, and generation—to build an extended version of the dataset. This format allows us to explicitly incorporate contextual information alongside the original tweets for use in the claim detection task. Some tweets do not receive contextual information because they lack identifiable entities or have no relevant matches in Wikipedia. Table 6 shows how contextual data is distributed across the four subsets.

Figure 6 shows that the word-length distributions of CT22 across all subsets are generally similar, while Figure 7 highlights that the $test$ and dev sets contain slightly longer tweets and exhibit more variation in length than the $train$ and dev_test sets. Although the dev_test and $test$ sets show similar overall word-length distributions, their maximum tweet lengths differ: 63 tokens for dev_test versus 33 for $test$. These differences are considered when interpreting performance metrics across evaluation sets.

Subset	Verifiable	Unverifiable	Total
train	2,122	1,202	3,324
dev	195	112	307
dev_test	574	337	911
test	149	102	251

Table 5: Statistics of CT22 English dataset.

Subset	Verifiable	Unverifiable	Total
train	2,069	1,125	3,194
dev	191	103	294
dev_test	565	326	891
test	141	90	231

Table 6: Statistics of retrieved contextual information on the CT22 dataset.

B Hyperparameters

B.1 Hyperparameter Optimization

For encoder-only models, we conduct systematic hyperparameter optimization across them to ensure optimal performance and fair comparison between baselines and our proposed approach. For this purpose, we employ the Optuna framework (Akiba et al., 2019), utilizing Bayesian optimization with the Tree-structured Parzen Estimator (TPE) sampler. Each model conducts 20 independent trials with a MedianPruner strategy implemented to terminate underperforming trials early, thus conserving computational resources. Given the imbalanced nature of our dataset and the specific requirements of claim verification systems, we design a multi-objective optimization approach. While maximizing the F1 score on the dev set served as our primary metric due to its balance of precision and recall, we also prioritize individual precision and recall metrics. This approach reflects our goal of filtering out as many unverifiable claims as possible to reduce the workload for subsequent claim verification stage, while still maintaining high recall for verifiable claims. Specifically, we employ a weighted combination of these metrics (0.6 for F1 score, 0.2 for precision, and 0.2 for recall) to select the optimal configuration. The best-performing hyperparameter configuration for each model is determined by the highest combined score across all trials, ensuring that each model was optimized

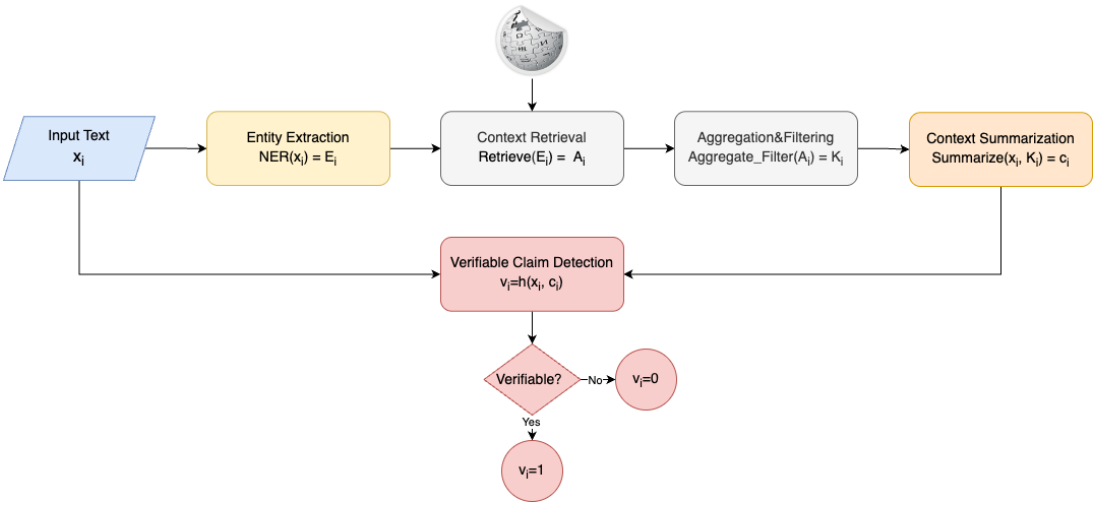

Figure 5: Proposed ContextClaim paradigm.

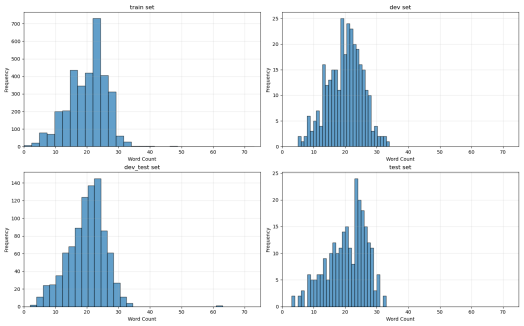

Figure 6: Word length distribution across datasets in CT22.

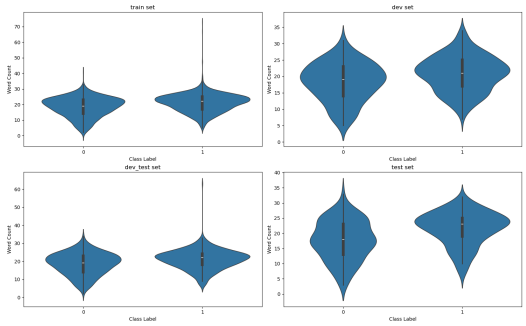

Figure 7: Word length distribution by class in CT22.

to its full potential for fair comparison. The hyperparameter search space and the final selected configurations for BERT and RoBERTa models corresponding to each dataset are detailed in Table 7, showing the optimized parameters used in our experiments.

B.2 Hyperparameter Configurations

Table 8 presents the fine-tuning configurations for decoder-only models, selected through empirical tuning to balance computational efficiency and performance.

C Further Detailed Analysis

C.1 Ablation Studies

As discussed in Section 3, the ContextClaim paradigm was introduced to improve verifiable claim detection by combining the original claim with additional context summaries. We compared this approach to a baseline that uses only the claim (Baseline), and also explored the impact of using two different types of contextual information—CC-G4o and CC-M—within the paradigm. In this section, we conduct an ablation study to further understand the role of each input component. Specifically, we examine model performance when only the generated context is used, and compare it against the Baseline and full ContextClaim setups.

Table 9 shows F1 scores for different models and input settings on both the *dev_test* and *test* sets. We find that using only contextual information (Context-G4o or Context-M) achieves results that are often close to those of the Baseline. For example, on the *dev_test* set, Context-G4o reaches 97.4% of Baseline performance for RoBERTa (0.8314 vs. 0.8586) and 96.6% for BERT (0.8222 vs. 0.8438). This suggests that the generated context alone can provide strong signals for classification—sometimes nearly as informative as the original claim. We also observe a consistent trend: GPT-4o-generated contexts tend to perform better than those from Mistral when used alone, with the exception of Llama3. For this

Hyperparameter	Search Space	BERT Models			RoBERTa Models		
		Baseline	CC-G4o	CC-M	Baseline	CC-G4o	CC-M
Epochs	5 to 15	8	5	8	12	20	20
Batch Size	{8, 12, 16, 20}	8	20	20	15	5	12
Dropout Rate	0.1 to 0.35	-	0.24	0.19	-	0.24	0.25
Warmup Ratio	0.1 to 0.2	0.18	0.1	0.14	0.2	0.1	0.18
Learning Rate	5.00E-06 to 5.00E-05	5.00E-05	4.00E-05	2.50E-05	8.10E-06	2.00E-05	7.10E-06

Table 7: Hyperparameter configurations for encoder-only models.

Parameter	Llama3	Mistral
Epochs	3	3
Batch size	16	8
Warmup ratio	0.1	0.05
Learning rate	1.00E-5	3.00E-4
Optimizer	adamw	paged_adamw[8]
Grad. accum.	2	4
LoRA r/α	64/16	64/16
LoRA dropout	0.1	0.1
Target modules	q,v,o	q,v

[8] 8-bit quantization; q,v,o: q_proj, v_proj, o_proj; Weight decay: 0.001; Max grad. norm: 1.0; Scheduler: cosine w/ restarts

Table 8: Decoder-only model hyperparameters.

model, Context-M outperforms Context-G4o by a large margin, with the F1 score for Context-G4o about 30% lower. In most other cases, Context-G4o has a 2–4% performance edge over Context-M, which supports the idea that GPT-4o's context captures more of the original claim's content. Importantly, when we combine both the claim and the context (ContextClaim), we generally see improvements over using either input alone. For instance, on the $test$ set, ContextClaim-G4o with RoBERTa achieves an F1 score of 0.7955—about a 1.6% increase over the better of the Baseline (0.7774) and Context-G4o (0.7829). While the gain is modest, it is consistent across different models, showing that the combination of both inputs provides complementary information that improves model performance.

Looking across both evaluation sets, we find the improvements from using context remain stable despite changes in data distribution. Context-only models, in particular, show strong generalization for Llama3, suggesting that the generated context may contain more domain-invariant features that help the model remain robust across different sets. In summary, while using context alone already provides strong classification signals, combining it with the original claim in the ContextClaim paradigm leads to the best overall performance by effectively leveraging the strengths of both input types.

Eval.	Model	F1 Score			
		BERT	RoBERTa	Llama3	Mistral
dev_test	Baseline	0.8438	0.8586	0.4997	**0.8475**
	CC-G4o	0.8499	0.8602	0.7757	0.8389
	CC-M	**0.8514**	**0.8637**	**0.7862**	0.8310
	C-G4o	0.8222	0.8314	0.3333	0.8189
	C-M	0.8141	0.8061	0.6172	0.7875
test	Baseline	**0.7706**	0.7774	0.4122	0.7678
	CC-G4o	0.7667	**0.7955**	0.7526	**0.7893**
	CC-M	0.7655	0.7864	**0.7613**	0.7783
	C-G4o	0.7584	0.7829	0.3198	0.7518
	C-M	0.7564	0.7488	0.6292	0.7452

Table 9: F1 scores for different paradigms across base models. CC = ContextClaim; C = Context only.

C.2 Detailed Error Analysis

Table 10 presents six representative error cases. The first three are unverifiable claims wrongly predicted as verifiable; the last three are the opposite. The "number of error" column shows how consistently each was misclassified across the seeds. In Example 1, a tweet speculates on Aaron Rodgers' vaccine motivations. Although the context provides accurate background, it reinforces the error by emphasizing connections to real-world entities without addressing the unverifiability of the stated motivation. This pattern reflects a broader difficulty the model has in separating factual content from unverifiable speculation. Another issue is the confusion between facts and opinions. For example, Example 3 contains a subjective statement about the NFL, which the model incorrectly labels as verifiable, even though the context confirms it's an opinion. On the other hand, Example 5 is a fact-based claim about voting deadlines, yet it's misclassified as unverifiable—possibly due to the presence of both factual and opinion elements, which the model fails to identify. The model also struggles with non-literal language. Examples 2 and 4 show how satire, hyperbole, or rhetorical language can lead to misclassification, as the model

fails to recognize exaggeration or irony as cues for unverifiability or misunderstands rhetorical elements in factual claims. Finally, in Example 6, the claim refers to a confidential Pfizer agreement. Though potentially verifiable, the model misclassifies it, suggesting difficulty in assessing claims that reference private or inaccessible information without directly presenting supporting information.

These examples highlight several areas where the model could be improved: better integration and interpretation of context, improved ability to distinguish between factual and opinion content, stronger handling of rhetorical language, and more nuanced reasoning about claims involving non-public information.

Tweet text	Contextual information	Class label	#Error
Rodgers heard the vaccine could help protect his family and he wanted no part of it	The tweet references Aaron Rodgers' reported stance on vaccines, implying he wanted no involvement with them despite hearing they could protect his family. Rodgers, an accomplished quarterback currently with the New York Jets, has been a public figure in discussions about vaccine hesitancy, a term describing the reluctance or refusal to vaccinate despite available services and strong supporting evidence of vaccine safety and effectiveness.	0	5
My cousin got the vaccine and it turned him into a llama and he fell into a river and had to hitch a ride with a village leader back to his castle, just saying do your own research	Vaccines are biological preparations that provide immunity against diseases. They have been extensively studied and verified for safety and effectiveness, typically containing agents that mimic a disease-causing organism.	0	4
Good thing the NFL is really focused on vaccines for some of the fittest people in the world and not the systemic abuse so prevalent with so many of its active and former players.	The tweet criticizes the NFL for its emphasis on vaccinations among players, who are considered some of the fittest individuals, rather than addressing issues of systemic abuse involving active and former players.	0	3
Guys Joe Rogan is NOT a scientist! Don't take what he says seriously. Coming up next, the Jonas brothers and big bird reminding you to get your vaccine!	Joseph James Rogan is an American podcaster known for hosting "The Joe Rogan Experience," a widely influential podcast where he discusses a variety of topics including science. Despite this influence, Rogan is not a trained scientist. The tweet humorously contrasts Rogan's non-expert status with other celebrities, such as the Jonas Brothers and Big Bird, promoting COVID-19 vaccinations, emphasizing that while entertaining, celebrity opinions on scientific matters should be considered cautiously.	1	5
CALIFORNIANS: My friend needs your support to stop the Republican recall. Vote no and return your ballot by tomorrow, 9/14 at 8PM. Vaccines, climate change, immigrant rights, minimum wage, reproductive rights, gun safety and more are on the ballot. VoteNoOnRecall	The tweet urges Californians to support Governor Gavin Newsom by voting against the Republican-led recall effort. It emphasizes the importance of returning ballots by the deadline to protect policies on issues like vaccines, climate change, immigrant rights, minimum wage, reproductive rights, and gun safety. Newsom, a Democrat, has been California's governor since 2019.	1	4
PFIZERLEAK: EXPOSING THE PFIZER MANUFACTURING AND SUPPLY AGREEMENT. (thread) Background: Pfizer has been extremely aggressive in trying to protect the details of their international COVID19 vaccine agreements. Luckily, I've managed to get one. PfizerLeak Pfizer	A tweet claims to have exposed a manufacturing and supply agreement related to Pfizer's COVID-19 vaccine. The tweet suggests that Pfizer has been actively trying to keep the details of its international vaccine agreements confidential. The individual behind the tweet, using the hashtag #PfizerLeak, asserts they have obtained one of these agreements. Pfizer, a well-established American pharmaceutical company founded in 1849, has been a key player in developing COVID-19 vaccines during the pandemic.	1	3

Table 10: Error examples of false positive and false negative on the $test$ set.