Automatic Fact-checking in English and Telugu

Ravi Kiran Chikkala[1,2] **Tatiana Anikina**[3] **Natalia Skachkova**[3]
Ivan Vykopal[4,5] **Rodrigo Agerri**[2] **Josef van Genabith**[3]
[1] Saarland University
[2] University of the Basque Country
[3] German Research Center for Artificial Intelligence, Saarland Informatics Campus
[4] Faculty of Information Technology, Brno University of Technology, Brno, Czech Republic
[5] Kempelen Institute of Intelligent Technologies, Bratislava, Slovakia
rach00004@teams.uni-saarland.de
{tatiana.anikina,natalia.skachkova,josef.van_genabith}@dfki.de
ivan.vykopal@kinit.sk
rodrigo.agerri@ehu.eus

Abstract

False information poses a significant global challenge, and manually verifying claims is a time-consuming and resource-intensive process. In this research paper, we experiment with different approaches to investigate the effectiveness of large language models (LLMs) in classifying factual claims by their veracity and generating justifications in English and Telugu. The key contributions of this work include the creation of a bilingual English-Telugu dataset and the benchmarking of different veracity classification approaches based on LLMs.

1 Introduction

In today's technological world, claim verification plays an important role (Zhang and Gao, 2023), which aims to assess the veracity of claims as "true" or "false" by validating them against trustworthy sources (Panchendrarajan and Zubiaga, 2024). This is necessary to combat false information, especially in multilingual countries such as India, where false information can be propagated in multiple languages via translation technology (Quelle et al., 2025). According to Pradeep et al. (2021), claim verification involves three key steps: (1) retrieval of documents, (2) rationale selection, and (3) label prediction. Currently, multilingual LLMs significantly improve the claim verification process (Schlichtkrull et al., 2023) compared to traditional approaches such as manual fact-checking and simple machine learning classifiers. These language models not only evaluate claims, but also provide justifications, thereby offering a level of explanation that traditional natural language processing (NLP) approaches often lack (Dmonte et al., 2024). To date, most of the work on claim verification in fact-checking has been performed in English. In this work, we address this shortcoming by creating a new fact-checking dataset in Telugu, allowing for large-scale experimentation in Telugu, a language spoken by over 200 million people in the world (Mallareddy, 2012). We achieve this by translating our manually created English dataset into Telugu, resulting in a bilingual English–Telugu dataset that supports multilingual claim verification. Furthermore, LLMs pose several limitations, such as tendencies to hallucinate (Li et al., 2024), they exhibit biases (Lin et al., 2025), smaller models may operate within limited context windows (Ratner et al., 2023), and models may rely on knowledge that may be outdated due to cutoff dates (Cheng et al., 2024). In order to address these challenges, we use Retrieval-Augmented Generation (RAG) (Lewis et al., 2020) with different components, such as prompt compression (Li et al., 2025), document re-ranking (Hui et al., 2022) and query rewriting (Ma et al., 2023).

We explore two research questions.

> ***RQ1:*** How well do LLMs classify domain-specific claims in English versus Telugu?
>
> ***RQ2:*** How do different models and approaches impact the quality of justifications provided by LLMs in a English-Telugu multilingual setting?

To address these research questions, we introduce a new dataset named **Preethi**[1] that covers both English and Telugu. Our experiments demonstrate that RAG-based approach, achieves the highest claim verification scores in both English and Telugu. For justification generation, RAG-based approach obtains the best average score for English, while Simple Prompting achieves the highest average score for Telugu.

[1]We make our complete dataset available at https://huggingface.co/datasets/Blue7Bird/Preethi_dataset

Proceedings of the 1st Workshop on Advancing NLP for Low-Resource Languages associated with RANLP 2025, pages 140–151, Varna, Bulgaria, Sep 13, 2025.
https://doi.org/10.26615/978-954-452-100-4-015

2 Related Work

2.1 Datasets Related to Indian Languages

Several datasets have been proposed for detecting false information in the Indian context. Sharma and Garg (2021b) introduce the Indian Fake News Dataset (IFND), a monolingual English dataset comprising 56,714 claims across various categories relevant to the Indian context. Each claim in IFND is labeled as "true" or "fake". Similarly, Gupta and Srikumar (2021) develop the X-Fact dataset, which includes 31,189 claims and supports multiple Indian languages-though not Telugu. X-Fact has five labels "true", "mostly-true", "partly-true", "mostly-false", and "false". Singhal et al. (2022) annotate the Fact Drill dataset, which comprises 22,435 false claims in 13 Indian regional languages, including fewer than 2,000 samples in Telugu. However, the dataset is not publicly available. Mittal et al. (2023) present the X-CLAIM dataset, which focuses on the identification of claims in multilingual social media posts. X-CLAIM contains 7,000 real-world claims across five Indian regional languages and English, but only 107 Telugu samples in its test set. Schlichtkrull et al. (2023) develop the AVeriTeC dataset, comprising 4,568 real-world claims in English. Each claim in AVeriTeC is classified into one of the four labels: "supported", "refuted", "not enough evidence" and "conflicting evidence/cherry-picking". Raja et al. (2023) create the Dravidian Fake News Dataset (DFND), which consists of 26,000 news articles in Telugu, Tamil, Kannada, and Malayalam, annotated with binary labels: "true" or "fake". However, the DFND is not open source, which poses challenges for reproducibility and further research.

Although some of these datasets support claim verification to varying degrees in Indian languages, none, except AVeriTeC, include human-annotated justifications and Question Answer (QA) pairs. Yet AVeriTeC is not designed for the Indian context. This highlights a research gap: the absence of open source, human-annotated QA pairs, and justification-rich resources for misinformation detection in low-resource Indian languages such as Telugu for the Indian context.

2.2 RAG and Other Approaches with LLMs

Recent advances in claim verification have used LLMs and RAG frameworks for claim verification processes (Dmonte et al., 2024). Singal et al. (2024) develop a RAG pipeline that extracts relevant evidence sentences from a knowledge base, which are then passed into an LLM for classification. Yue et al. (2024) introduce a Retrieval-Augmented Fact Verification framework through the synthesis of contrasting arguments (RAFTS) to determine the veracity of the claim. Katranidis and Barany (2024) propose Facts as a Function approach (FaaF), which is based on RAG, to evaluate the factual accuracy of the text generated by LLMs. Vykopal et al. (2024) present a comprehensive review of claim verification frameworks that use LLMs, focusing on methods such as RAG and fine-tuning. Our work is different from previous research, as we implement a RAG pipeline that enhances LLMs' fact-checking capability, using Automatic Scraping, integrating both foundational and Advanced RAG components. We use Really Simple Syndication (RSS) (Wikipedia contributors, 2024) feeds from reputable Indian news sources, chosen for their longstanding credibility and wide readership, to access up-to-date information to assess new claims, as LLMs' have knowledge cutoff dates and may contain outdated information.

3 Preethi Dataset

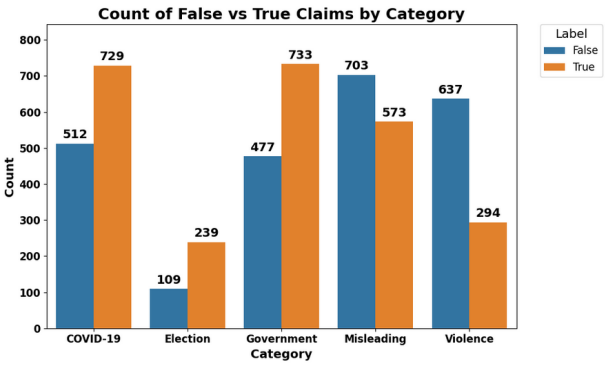

Figure 1: Statistical information of the Preethi dataset about true and false claims across five categories

In this research work, we have created the Preethi dataset, which is based on the publicly available IFND (Sharma and Garg, 2021a). A claim, as defined by Panchendrarajan and Zubiaga (2024), is a statement that can be verified against evidence. The IFND has several inconsistencies such as incomplete claims, non-claims, questions, and entries with multiple claims; these inconsistencies compromise its overall quality as these claims cannot be verified against evidence, see Table 1 for examples of inconsistent claims. We chose IFND because it is publicly available and its inconsistencies

highlight the need for a refined and higher-quality resource, an opportunity we address through the creation of Preethi dataset. We have manually annotated a dataset of 5,006 claims in English with five topics from IFND, namely *Covid-19*, *Election*, *Government*, *Misleading*, and *Violence*. Statistical details of the Preethi dataset are presented in Figure 1. The Preethi dataset is not a strict subset of IFND. Of the 2,568 true claims, 2,500 are sourced from IFND. Among the 2,438 false claims, 2,435 are collected from fact-checking websites. Following, Egelhofer and Lecheler (2019) we treat partially true claims from fact-checking sources as false, given their potential to spread misinformation similar to fully false claims. To reconstruct complete claims from inconsistent IFND entries, we use their original sources, identified via Google Web Search (Google, 2024a) and, when necessary, Microsoft Copilot (Microsoft, 2024).

Claim	Inconsistency
This Video Is Not Of UP Police Chasing A.....	Incomplete
Did Israel bomb Iranian nuclear facilities?	Question
Drop, Don't Extend It	Non-claim
India's Ministry of Culture has NOT announced a relief....Fact Check: Chill. Iceland hasn't declared religions as weapons of mass destruction	Multiple claims

Table 1: Inconsistent claims in IFND

Inspired by the AVeriTeC, we provide additional metadata for each claim, including supporting documents from the Web, the date of the claim, gold justifications, and gold QA pairs. Gold justifications and gold QA pairs are created manually based on the information in the supporting documents. To maintain the quality of the dataset, we have involved three annotators who were trained via detailed guidelines. We achieve a Cohen's Kappa agreement score of 80% for claim veracity labels and 75% for boolean QA pairs, indicating substantial inter-annotator agreement. In addition, all abstractive and extractive QA pairs are manually checked by annotators for correctness and relevance by verifying them against supporting documents. To make our dataset available in Telugu, we translate the English dataset using the Google Translate API (Google, 2024b). To assess the quality of the translated data, we perform a back-translation from Telugu to English and compare it with the original English version. This results in a BLEU (Papineni et al., 2002) score of 0.255 and a METEOR (Banerjee and Lavie, 2005) score of 0.659, indicating moderate consistency between the original and back-translated texts. However, the raw machine translations are not directly used in our experiments. Instead, three native Telugu speakers have manually post-edited the machine translated output and removed the syntactic and semantic errors. The final Telugu dataset is used for experiments, ensuring high-quality translations and minimizing the potential bias introduced by machine translation errors. We calculate post-edits by comparing the initial machine-translated Telugu dataset with the final manually annotated Telugu dataset using Pyter (pyter developers, 2024) to measure the translation error rate (TER) (Snover et al., 2006). A total of 31,465 post-edits are made. Table 2 compares Preethi dataset to the existing benchmark datasets.

Dataset	Justifications	Supports Telugu	QA Pairs
X-CLAIM	✗	✓	✗
AVeriTeC	✓	✗	✓
DFND	✗	✓	✗
IFND	✗	✗	✗
X-Fact	✗	✗	✗
Fact Drill	✗	✓	✗
Preethi (ours)	✓	✓	✓

Table 2: Comparison of Preethi dataset with benchmark datasets.

3.1 QA Pairs

Each claim in our dataset has three manually created QA pair types; see Table 3 for examples.

Boolean: Our dataset contains 4,010 indirect and 996 direct boolean QA pairs. Direct QA pairs rephrase the claim itself as a yes/no question, while indirect QA pairs pose a related yes/no question that helps verify the validity of the claim.

Abstractive: QA pairs are created by summarizing the relevant information about the claim.

Extractive: QA pairs, in which the answer is a direct snippet or a phrase taken word-for-word.

Claim	*The Eiffel Tower is in London*
QA Type	**Question(Q) & Answer(A)**
Direct Boolean	Q: Is the Eiffel Tower in London? A: No
Indirect Boolean	Q: Is the Eiffel Tower in France? A: Yes
Abstractive	Q: What is the Eiffel Tower? A: a well known monument....
Extractive	Q: Where is the Eiffel Tower? A: Paris, France.

Table 3: Boolean, Abstractive and Extractive QA pairs

4 Methodology and Experiments

This section discusses different approaches that are used in our experiments: *1) Simple Prompting* and *RAG* approaches that include *2) Naive RAG*; *3) Advanced RAG* and *4) Automatic Scraping*. In our

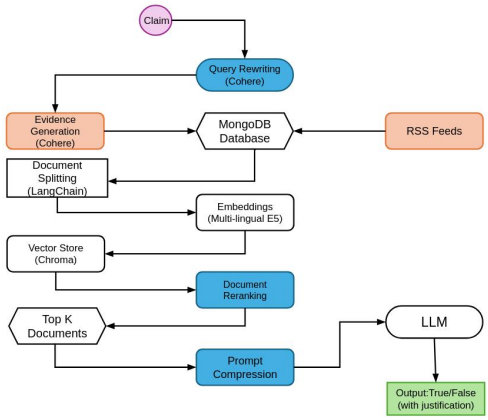

Figure 2: RAG Approaches

experiments, we use gold justifications to evaluate the justifications generated by LLMs and gold QA pairs to assess the quality of QA pairs generated by LLMs. For claim veracity evaluation, we use F1 score. In order to evaluate the justifications generated by LLMs, we use METEOR, ROUGE-L (R-L) (Lin, 2004), ChrF (Popović, 2015), BERTScore (Zhang* et al., 2020) and BLEURT (Sellam et al., 2020). We make our complete code [2] and additional details public.

4.1 Simple Prompting

In Simple Prompting, we use a zero-shot approach (Wei et al., 2022), where the LLM relies solely on its pre-trained knowledge and general language understanding to classify only claims, operating without any additional supporting documents. In this approach the LLM is given a claim as input, and it is tasked to classify a claim as "false" or "true" and provide reasoning or justification for its decision. Without such explanations, the classification may appear arbitrary or unsupported. For the experiments, we consider the Simple Prompting approach as a baseline.

4.2 RAG Approaches

Since LLMs are not updated regularly and have a fixed knowledge cut-off date, they may hallucinate. To address this, we use RAG. In order to find supporting documents for claims, we use the Cohere c4ai-command-r7b-12-2024 (Cohere For AI, 2024) model for English and Telugu. To handle the large number of new claims that appear every day, we use RSS feeds. These feeds are updated regularly by different on-line news sources, providing up-to-date information. To manage this data, we choose the MongoDB (MongoDB Inc., 2024) database for our experiments. It is a NoSQL database suitable for unstructured data, making it ideal for storing RSS feeds and supporting documents retrieved by Cohere. We collect RSS feeds from reliable Indian news sources such as NDTV (NDTV, 2024) for English and Eenadu (Eenadu, 2024) for Telugu. Chroma (Chroma, 2024), a vector database, is used to store documents using vector representations.

4.2.1 Naive RAG

In the Naive RAG approach, as shown in Figure 2 (excluding the steps highlighted in blue), the process unfolds as follows:

Step 1: Cohere c4ai-command-r7b-12-2024 model is prompted to provide supporting documents for a given claim. These documents are then stored in MongoDB.

Step 2: The MongoDB retriever, which uses string matching, identifies, and retrieves documents relevant to the claim. The retrieved documents are processed through the LangChain text splitter (LangChain, Inc., 2025), which divides documents into smaller segments.

Step 3: These segments are converted to vector embeddings using multilingual E5 Text embeddings (Wang et al., 2024). These embeddings are then stored in Chroma.

Step 5: Cosine similarity is used to compare the embeddings of the claim with the documents stored in Chroma. The top three documents with the highest cosine similarity scores are retrieved from Chroma and used as evidence for the LLM.

Step 6: Finally, the LLM analyzes the evidence in the context of the claim and classifies the claim as true or false, along with justifications for its decision.

4.2.2 Advanced RAG

Advanced RAG is similar to Naive RAG but with additional components such as query re-writing, document re-ranking, and prompt compression. In Figure 2 the additional components of Advanced RAG are highlighted in blue.

Query Re-writing: For query re-writing, we use the Cohere c4ai-command-r7b-12-2024 model, which modifies the original claim to improve its quality for better retrieval of documents. This includes correcting spelling errors, rephrasing, or adding additional context to a claim for better understanding. See Table 4 for examples. According

[2] https://github.com/formallinguist/Automatic-Fact-Checking

to Skitalinskaya and Wachsmuth (2023), the criteria for re-writing a claim include maintaining syntactic and semantic coherence, being grammatically correct, and removing ambiguity. A good claim is precise, includes relevant context, and is not ambiguous. In our experiments, we observe that 50 % - 60 % of claims undergo this process. We calculate this using string matching. We have manually verified 50 claims in English and Telugu to check the quality of the re-written claims. We observe that re-written claims in English are syntactically and semantically coherent, while Telugu re-written claims have grammatical errors.

Document Re-ranking: For document re-ranking, we use bert-multilingual-passage-reranking-msmarco (ambeRoad, 2022). It calculates the relevance of each document with respect to the claim and then sorts the documents by the scores to determine the best matches. This ensures the documents that are most relevant for the claim are ranked higher for further processing. Unlike cosine similarity in the Naive RAG approach, which only compares vector proximity, here, the CrossEncoder evaluates the relationship between the claim and the document in context. The top three re-ranked documents are considered for further processing.

Prompt Compression: For prompt compression, we use the Cohere c4ai-command-r7b-12-2024 model. This involves reducing the length of a prompt while retaining its most important information. This helps in scenarios where there is a limited context window for an LLM.

4.2.3 Automatic Scraping

In Automatic Scraping, we extract content from URL in the supporting documents of the Preethi dataset using BeautifulSoup (BS4) (Richardson). To overcome the limitation of the context window of the LLMs, we use a sentence-transformers/paraphrase-multilingual-mpnet-base-v2 (Reimers and Gurevych, 2019). This model identifies the most relevant sentences from the supporting documents by comparing their semantic similarity to the given claim. We retrieve up to 3,000 characters of content that are most relevant to the claim. This selected content is then used as the context for the LLM and is referred to as the *refined context*. This approach is repeatable with new data if the claim and its URL are available.

4.3 Evaluation of QA pairs

In claim verification, asking good questions is crucial (Schlichtkrull et al., 2023). To assess the quality of QA pairs generated by LLMs, we calculate their similarity to gold-standard QA pairs. We use an in-context learning approach (Dong et al., 2024), where the gold QA pair serve as reference to guide the LLM in generating boolean, abstractive, and extractive QA pairs in the desired format for a claim. For evaluation, we follow the approach of Schlichtkrull et al. (2023), we first compute METEOR scores and then apply the Hungarian algorithm (Kuhn, 1955) to identify the optimal one-to-one matching between LLM-generated and gold-standard QA pairs by maximizing METEOR scores. Table 7 provides English and Telugu scores.

4.4 Experiments

We use the models listed in Table 5 for experiments.

Versions of models	Parameters
Gemma-2 (Team, 2024)	9B
Llama-3 (AI@Meta, 2024)	70B
Llama-3.3 (Meta, 2024)	70B
Llama-3 (Meta, 2024)	8B
Mixtral (Jiang et al., 2024)	8x7B

Table 5: Models for experiments

We select models that are trained on publicly available online data. All LLMs are instructed in English, and we experiment with three different prompt templates, selecting the best-performing one for our experiments. To ensure consistency of the results, each experiment is conducted three times, with same temperature. We calculate variance across the three runs for both English and Telugu using the F1 scores of the best-performing models. For English, the Naive RAG exhibits the highest variance, while the Advanced RAG shows the lowest. For Telugu, Automatic Scraping results in the highest variance, whereas the Naive RAG has the lowest. Table 6 shows average scores of model performance across English and Telugu datasets. We use multiple evaluation metrics in our experiments to gain a comprehensive understanding of the models' performance, as no single metric can fully capture the quality of a model's output for justification generation.

No.	Original Claim	Re-written Claim
1	Jharkhand new hotspot of illicit opium cultivation: NCB	The NCB reports significant opium cultivation in Jharkhand, identifying it as a potential hotspot.
2	Govt confident of privatising Air India, BPCL by first half of 2021-22 divestment secretary	The Indian Government's Divestment Strategy: Privatization of Air India and BPCL by 2022 and the Secretary's Statement on Future Plans.

Table 4: Comparison of original and re-written claims

Model	Approach	F1 (Claim Verif.)		English Justification Generation Scores						Telugu Justification Generation Scores					
		En	Te	METEOR	R-L	ChrF	BLEURT	BERTScore	Avg-En	METEOR	R-L	ChrF	BLEURT	BERTScore	Avg-Te
Llama-3-70B	SP	80.16	42.95	0.288	0.283	39.92	0.48	0.87	0.464	0.126	0.165	25.34	0.45	0.72	0.343
	N-RAG	58.16	40.77	0.267	0.275	38.51	0.47	0.86	0.451	0.140	0.163	23.94	0.44	0.71	0.339
	A-RAG	61.21	44.31	0.256	0.259	**41.11**	**0.56**	0.88	0.473	0.134	0.174	26.08	0.48	0.72	0.354
	AS	**86.14**	80.45	0.281	0.289	37.72	0.47	0.89	0.461	0.123	0.162	24.70	0.45	0.72	0.340
Llama-3.3-70B	SP	75.07	70.68	0.275	0.275	38.55	0.47	0.89	0.459	**0.172**	**0.229**	**32.79**	**0.51**	0.71	**0.390**
	N-RAG	57.44	38.86	0.286	0.282	35.41	0.43	0.87	0.444	0.106	0.123	27.04	0.40	0.72	0.324
	A-RAG	59.38	41.76	0.259	0.250	37.81	0.42	0.88	0.437	0.135	0.174	28.29	0.43	0.72	0.348
	AS	77.22	**80.58**	0.308	0.318	39.81	0.49	**0.90**	0.482	0.163	0.196	31.84	0.50	**0.73**	0.381
Llama-3-8B	SP	56.21	48.45	0.294	0.279	39.85	0.50	0.89	0.472	0.138	0.194	29.64	0.42	**0.73**	0.356
	N-RAG	52.41	47.29	0.266	0.280	37.59	0.45	0.86	0.446	0.139	0.184	28.21	0.41	0.72	0.347
	A-RAG	60.11	49.75	0.254	0.304	38.41	0.49	0.89	0.464	0.133	0.203	29.61	0.44	0.72	0.358
	AS	70.83	50.77	0.288	0.291	38.64	0.47	0.87	0.461	0.124	0.164	25.96	0.42	0.72	0.338
Mixtral-8x7B	SP	56.95	49.22	0.285	0.273	38.94	0.49	0.88	0.463	0.110	0.129	27.59	0.29	0.72	0.305
	N-RAG	57.19	51.24	0.293	0.303	37.51	0.47	0.86	0.460	0.153	0.172	28.39	0.41	**0.73**	0.350
	A-RAG	59.26	55.49	0.280	0.293	38.66	0.51	0.89	0.472	0.146	0.213	28.66	0.43	0.72	0.359
	AS	84.08	73.86	**0.316**	**0.340**	41.03	0.48	0.87	**0.483**	0.087	0.114	23.27	0.28	0.70	0.283
Gemma-2-9B	SP	64.72	57.41	0.208	0.283	31.89	0.46	0.87	0.428	0.125	0.183	26.66	0.43	**0.73**	0.347
	N-RAG	62.21	52.39	0.197	0.264	30.51	0.45	0.87	0.417	0.103	0.173	28.41	0.43	0.72	0.342
	A-RAG	63.81	50.77	0.180	0.283	34.74	0.49	**0.90**	0.440	0.094	0.213	30.49	0.46	0.72	0.358
	AS	83.23	78.05	0.217	0.277	36.77	0.48	0.87	0.442	0.114	0.152	24.68	0.42	0.72	0.331

Table 6: Scores across different metrics for English (En) and Telugu (Te). Approaches include Simple Prompting (SP), Naive RAG (N-RAG), Advanced RAG (A-RAG), and Automatic Scraping (AS). The best results for each metric and language are highlighted in **bold**, while the best scores per metric and language for each model are underlined. ChrF scores are normalized (divided by 100) when computing average scores for English and Telugu.

Model	En	Te
Llama-3-70B	0.101	0.072
Llama-3.3-70B	0.140	0.090
Llama-3-8B	0.178	**0.124**
Mixtral-8x7B	**0.208**	0.089
Gemma-2-9B	0.126	0.079

Table 7: QA pairs Hungarian METEOR scores for English (En) and Telugu (Te). Best scores are highlighted in **bold**

5 Results and Discussion

We analyze claim verification and justification generation results for English and Telugu to answer RQ1 and RQ2, and also analyze QA pair results.

5.1 Claim Verification

In order to answer RQ1, we examine the claim verification results presented in Table 6.

5.1.1 English

Simple Prompting: Within the Simple Prompting approach across models, Llama-3-70B achieves the highest F1 score, likely due to its large size and English-focused training, enabling strong reasoning without external supporting documents. In contrast, Llama-3-8B performs the worst, likely due to its smaller size. Interestingly, Llama-3-70B outperforms both Naive and Advanced RAG under Simple Prompting, showing the largest performance gap of 23.95 points of F1 score between the best and worst performing models.

Automatic Scraping: We observe that all models obtain their highest F1 scores with this approach. With Automatic Scraping Llama-3-70B has the highest F1 score and Llama-3-8B has the lowest F1 score. This suggests that Automatic Scraping provides high-quality, relevant context that helps LLMs verifying and classifying the claims. All models perform better with Automatic Scraping compared to Simple Prompting.

Naive RAG: Gemma-2-9B achieves the highest F1 score and Llama-3-8B has the lowest F1 score. We observe that the Naive RAG approach does not improve the models' performance with respect to Simple Prompting except for Mixtral-8x7B. One possible reason for the relatively low F1 scores

across models is that the Cohere model may not retrieve suitable supporting documents, particularly for claims related to the Indian context. This limitation at the evidence retrieval stage can significantly impact the quality of context available to the LLM, thus reducing overall performance.

Advanced RAG: Gemma-2-9B achieves the highest F1 score and Mixtral-8x7B shows the lowest F1 score. We observe that models consistently perform slightly better with Advanced RAG compared to Naive RAG. This improvement may be attributed to the additional components in Advanced RAG that enhance the models' overall performance. However, results with the Simple Prompting approach remain superior except for Llama-3-8B and Mixtral-8x7B. Notably, this approach results in the smallest performance gap of 4.55 points in average F1 score between the best and worst performing models.

5.1.2 Telugu

Simple Prompting: Within the Simple Prompting approach, Llama-3.3-70B obtains the highest F1 score, likely due to some knowledge of Telugu in its pre-training data, as it was trained on open-source web documents. In contrast, all other models have low F1 scores. This could be due to the limited presence of Telugu in their pre-training corpora.

Naive RAG: Gemma-2-9B has the highest F1 score and Llama-3.3-70B has the lowest F1 score. The relatively low scores across models may be attributed to the Cohere model's limited ability to retrieve relevant supporting documents for claims in Telugu. Since Telugu is a low-resource language, the amount and quality of content available in it would be significantly lower compared to English. In this approach, only Mixtral-8x7B performs better than the models with Simple Prompting. This approach has the lowest performance gap of 14.03 F1 points between the best and the worst performing models.

Advanced RAG: Mixtral-8x7B has the highest F1 score and Llama-3.3-70B has the lowest F1 score. The F1 scores across models suggest that the Advanced RAG generally performs slightly better than the Naive RAG for Telugu, with the exception of Gemma-2-9B. This exception may be due to Gemma-2-9B not having received suitable documents as context. The modest improvements seen with Advanced RAG can likely be attributed to its additional components. However, F1 scores for Telugu remain relatively low compared to those for English. Among the evaluated models, Llama-3-70B, Llama-3-8B, and Mixtral-8x7B outperform Simple Prompting.

Automatic Scraping: Under this method, in which the context is in English, Llama-3.3-70B achieves the highest F1 score, demonstrating its ability to transfer knowledge from English to Telugu. In comparison, the smaller Llama-3-8B has the lowest F1 score. These results highlight that LLMs perform significantly better in Telugu when provided with suitable supporting documents. Here, all models perform better than Simple Prompting. The performance gap between the best and the lowest performing model is 29.81 average F1 score, which is highest using this technique.

Automatic scraping has the highest scores for claim verification as it uses reliable supporting documents as context. To answer RQ1, our experiments show that LLMs perform better at claim verification in English compared to Telugu.

5.2 Justification Generation

As shown in Table 6, we compare the results of justification generation score (JGS) for Telugu and English to answer RQ2. JGS is an average of METEOR, R-L, ChrF, BLUERT and BERTScore. We observe that for English and Telugu different models and approaches have high scores across different metrics. However, for English, Mixtral-8x7B with Automatic Scraping has the highest overall average JGS. The best overall JGS in Telugu is attained by Llama-3.3-70B using Simple Prompting. Manual review of 100 justifications from various methods reveals no clear link between claim verification and JGS.

5.3 QA pairs

As shown in Table 7, Mixtral-8x7B achieves the highest METEOR score for English, likely because it is trained on predominantly English data. In contrast, Llama-3-8B, despite being a small model, achieves the best METEOR score for Telugu. This performance may result from its closer adherence to the reference QA pairs, whereas larger models tend to "hallucinate" or be creative (Lin et al., 2022), which negatively affects similarity scores .

6 Error Analysis

In this section, we present the qualitative and quantitative error analysis for English and Telugu.

6.1 Qualitative Error Analysis

We manually analyze 100 samples from the best performing models for each task: Llama-3-70B (English) and Llama-3.3-70B (Telugu) for claim verification; Mixtral-8x7B (English) and Llama-3.3-70B (Telugu) for justification generation.

6.1.1 Claim Verification

Example 1 Bias Error
Model output : The indian air force conducted an air strike on a jaish-e-mohammed training camp in balakot, pakistan, on february 26, 2019, *reportedly killing several terrorists.*

Example 2 Hallucination Error
Model output : covishield, will be priced at *around ₹200-₹300 per dose, not ₹1,000.*

Example 3 Retrieval Error
Claim: Congress mla calls kumaraswamy's absence at tipu jayanti celebrations an insult to muslims.
Model output: the provided context discusses ramdas athawale's criticism of raj thackeray, not yogesh sagar's protest against road closures for friday prayers.

Example 4 Translation error
Claim in Telugu: రక్షణ మంత్రి రాజ్‌నాథ్ సింగ్ *BRO* నిర్మించిన 44 వ్యూహాత్మక వంతెనలను ప్రారంభించారు, అందులో 7 లడఖ్‌లో ఉన్నాయి.
Claim in English: Defense Minister Rajnath Singh inaugurated 44 strategic bridges built by *BRO*, out of which 7 are in Ladakh.
Model output Translation from Telugu: It is *not Rajnath singh's brother,* it is defense minister Rajnath singh who inaugurated these 44 strategic bridges.

Figure 3: Different types of Errors

We have focused on identification of biases (Dev et al., 2022), hallucinations (Li et al., 2024), retrieval, and translation errors. **Biases** are unfair patterns in responses that occur when the model favors certain views, stereotypes, or groups over others. As shown in Example one in Figure 3, there is a potential bias toward labeling individuals as terrorists. **Hallucinations** occur when LLMs generate information that is factually incorrect. In Example two in Figure 3 the language model hallucinates about the pricing of the Covishield vaccine. **Retrieval errors** in RAG approaches refer to the failing of the model to obtain relevant or sufficient contextual knowledge to support accurate reasoning, leading to incorrect or unsupported output. Example three in Figure 3 shows that the retrieved documents are not related to the claim about *kumaraswamy* and *tipu jayanti*. Finally, **translation errors** are uniquely observed when there is a language mismatch in the claim or between claim and context - for example, when there are acronyms in English and the claim is in Telugu. In such scenarios, the models attempt to translate the English acronyms to Telugu as in Example four in Figure 3 where it can be observed that BRO acronym which is in English is translated to "brother" in Telugu.

Approach	B	H	R	O
SP	13.14%	4.81%	–	1.92%
AS	4.91%	1.02%	3.85%	4.08%
N-RAG	12.12%	5.39%	13.54%	10.76%
A-RAG	11.98%	1.22%	16.75%	9.27%

Table 8: English errors with percentage (relative to 5006 claims). B: Biases, H: Hallucinations, R: Retrieval, O: Other.

Approach	B	H	R	T	O
SP	5.17%	10.71%	–	–	13.16%
AS	1.00%	2.46%	–	0.26%	12.86%
N-RAG	8.79%	9.35%	1.62%	8.63%	20.59%
A-RAG	0.50%	8.25%	10.53%	–	13.18%

Table 9: Telugu errors. B: Biases, H: Hallucinations, R: Retrieval, T: Translation, O: Other.

6.2 Justification Generation

We manually evaluate generated 100 justifications against the gold-standard justifications from different approaches. We observe that Automatic Scraping enables LLMs to generate good-quality justifications in English and Telugu. Manual inspection further reveals that the quality of text generation is generally good for English across different models and approaches. However, outputs in Telugu often exhibit syntactic and semantic errors, along with instances of Tenglish (a mix of Telugu and English) script.

6.3 Quantitative Error Analysis

We use the mistral-saba-24B LLM (AI, 2025) as a judge, following an in-context learning approach.We manually select one misclassified claim with its justification from each error type as a demonstration for the judge. Misclassified claims and their justifications are filtered and they are then classified by the LLM into the predefined error categories, with uncategorized errors labeled as "Other." Tables 8 and 9 report category-wise error percentages. Manual verification of 50 errors per language confirms accurate quantification.

7 Conclusion

In this project, we introduced a new English-Telugu claim verification dataset with manually annotated QA pairs and justifications. We used it to benchmark Simple Prompting and RAG approaches with LLMs. Our results show that the models perform better in English than in Telugu, highlighting challenges in claim verification and justification generation in Telugu.

Limitations

The results of our experiments are based on a dataset of 5,006 claims with only two labels from five topics. Performance may vary with larger and more diverse datasets. In India, claims occur in multiple languages, but for this study, we work in one language at a time. We need to explore different prompt templates for Telugu and English, as some templates perform better than others. Our dataset consists only of textual claims, excluding images and videos, which are also commonly associated with the spread of false claims. Although we have relied on lexical and semantic similarity metrics, we have not incorporated additional text generation metrics to detect hallucinations. Our evaluation relies exclusively on automatic metrics such as R-L, METEOR, and BERTScore. While these provide surface-level and semantic overlap, they may not adequately capture the true quality of either QA pairs or justifications. In particular, justifications can often be expressed in many valid ways that differ substantially from the reference, leading to artificially low metric scores, while conversely, outputs that are lexically or semantically similar to the reference may still be incorrect. The limited variance in our reported BERTScore values (0.70–0.73) for Telugu further suggests that these metrics may not be sensitive enough to meaningful differences in justification quality. A more robust assessment would require human evaluation, which could better judge correctness, faithfulness, and usefulness of both the questions/answers and the justifications. Future work should therefore complement automatic metrics with systematic human evaluation. Naive RAG and Advanced RAG approaches that we use for experiments often require significant processing time, particularly for languages like Telugu. This is due to the complexity of tokenization, retrieval, and generation stages, which may not be as optimized for low-resource languages as they are for English. We have used RSS feeds from only a small number of sources and we have not performed ablation studies on the individual components of Advanced RAG. Since our dataset is derived through translation from English, it may not fully represent native Telugu. Translations tend to exhibit different levels of formality, topic distribution, and cultural biases compared to texts in Telugu produced by native speakers. Therefore, while our dataset serves as a useful resource, we acknowledge that future work should prioritize collecting and incorporating more native-authored Telugu data.

Acknowledgements

We thank Begari Kaveri, Sujatha Theetla and Ravi Teja Chikkala for reviewing and editing the Telugu translations and inter-annotation agreement of the Preethi dataset.

This project was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) as part of the project TRAILS (01IW24005), by *DisAI - Improving scientific excellence and creativity in combating disinformation with artificial intelligence and language technologies*, a project funded by Horizon Europe under GA No.101079164, by the *European Union NextGenerationEU* through the Recovery and Resilience Plan for Slovakia under the project No. 09I01-03-V04-00007, and by Saarland University and University of Basque country in collaboration with Erasmus Mundus Language and Communication Technologies (EMLCT).

References

Mistral AI. 2025. Mistral small 3: A 24 billion parameter language model.

AI@Meta. 2024. Llama 3 model card.

ambeRoad. 2022. bert-multilingual-passage-reranking-msmarco. https://huggingface.co/amberoad/bert-multilingual-passage-reranking-msmarco. Accessed: 2025-01-12.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Jeffrey Cheng, Marc Marone, Orion Weller, Dawn Lawrie, Daniel Khashabi, and Benjamin Van Durme. 2024. Dated data: Tracing knowledge cutoffs in large language models. In *Proceedings of the Conference on Language Modeling (COLM)*.

Chroma. 2024. ChromaDB – The AI-native open-source vector database. https://www.trychroma.com/. Accessed: 2025-01-12.

Cohere For AI. 2024. Cohere c4ai-command-r7b-12-2024. https://huggingface.co/CohereForAI/c4ai-command-r7b-12-2024. Accessed: 2025-01-12.

Sunipa Dev, Emily Sheng, Jieyu Zhao, Aubrie Amstutz, Jiao Sun, Yu Hou, Mattie Sanseverino, Jiin Kim, Akihiro Nishi, Nanyun Peng, and Kai-Wei Chang. 2022. On measures of biases and harms in NLP. In *Findings of the Association for Computational Linguistics: AACL-IJCNLP 2022*, pages 246–267, Online only. Association for Computational Linguistics.

Alphaeus Eric Dmonte, Roland Oruche, Marcos Zampieri, Prasad Calyam, and Isabelle Augenstein. 2024. Claim verification in the age of large language models: A survey. *ArXiv*, abs/2408.14317.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1107–1128, Miami, Florida, USA. Association for Computational Linguistics.

Eenadu. 2024. Eenadu – Telugu News Portal. https://www.eenadu.net/. Accessed: 2025-01-12.

Jana Laura Egelhofer and Sophie Lecheler. 2019. Fake news as a two-dimensional phenomenon: A framework and research agenda. *Annals of the International Communication Association*, 43(2):97–116.

Google. 2024a. Google Search. https://www.google.com/. Accessed: 2024-11-12.

Google. 2024b. Google Translate. https://translate.google.com/?sl=ta&tl=en&op=translate. Accessed: 2024-11-12.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 675–682, Online. Association for Computational Linguistics.

Kai Hui, Honglei Zhuang, Tao Chen, Zhen Qin, Jing Lu, Dara Bahri, Ji Ma, Jai Gupta, Cicero Nogueira dos Santos, Yi Tay, and Donald Metzler. 2022. ED2LM: Encoder-decoder to language model for faster document re-ranking inference. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3747–3758, Dublin, Ireland. Association for Computational Linguistics.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L'elio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts. *ArXiv*, abs/2401.04088.

Vasileios Katranidis and Gabor Barany. 2024. Faaf: Facts as a function for the evaluation of generated text.

Harold W. Kuhn. 1955. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97.

LangChain, Inc. 2025. langchain_text_splitters.base.TextSplitter (API Reference). https://python.langchain.com/api_reference/text_splitters/base/langchain_text_splitters.base.TextSplitter.html. Accessed: 2025-01-12.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Junyi Li, Jie Chen, Ruiyang Ren, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2024. The dawn after the dark: An empirical study on factuality hallucination in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10879–10899, Bangkok, Thailand. Association for Computational Linguistics.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. 2025. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7182–7195, Albuquerque, New Mexico. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Luyang Lin, Lingzhi Wang, Jinsong Guo, and Kam-Fai Wong. 2025. Investigating bias in LLM-based bias detection: Disparities between LLMs and human perception. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 10634–10649, Abu Dhabi, UAE. Association for Computational Linguistics.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland. Association for Computational Linguistics.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of*

the 2023 Conference on Empirical Methods in Natural Language Processing, pages 5303–5315, Singapore. Association for Computational Linguistics.

K. Mallareddy. 2012. Evolution of telugu language teaching and challenges to present curricular trends. *IOSR Journal of Humanities and Social Science*, 5:33–36.

Meta. 2024. Llama-3.3-70b-instruct.

Microsoft. 2024. Microsoft Copilot. https://copilot.microsoft.com/. Accessed: 2024-11-12.

Shubham Mittal, Megha Sundriyal, and Preslav Nakov. 2023. Lost in translation, found in spans: Identifying claims in multilingual social media. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3887–3902, Singapore. Association for Computational Linguistics.

MongoDB Inc. 2024. MongoDB. https://www.mongodb.com. Accessed: 2025-1-12.

NDTV. 2024. NDTV - Latest News and Updates. https://www.ndtv.com/. Accessed: 2025-01-12.

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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Ronak Pradeep, Xueguang Ma, Rodrigo Nogueira, and Jimmy Lin. 2021. Scientific claim verification with VerT5erini. In *Proceedings of the 12th International Workshop on Health Text Mining and Information Analysis*, pages 94–103, online. Association for Computational Linguistics.

pyter developers. 2024. pyter: Python Text Segmentation. https://pypi.org/project/pyter/. Accessed: 2025-06-12.

Dorian Quelle, Calvin Yixiang Cheng, Alexandre Bovet, and Scott A. Hale. 2025. Lost in translation: using global fact-checks to measure multilingual misinformation prevalence, spread, and evolution. *EPJ Data Science*, 14(1):22.

Eduri Raja, Badal Soni, and Samir Kumar Borgohain. 2023. Fake news detection in dravidian languages using transfer learning with adaptive finetuning. *Eng. Appl. Artif. Intell.*, 126:106877.

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402, Toronto, Canada. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Leonard Richardson. Beautiful soup 4 documentation. https://beautiful-soup-4.readthedocs.io/en/latest/. Accessed: January 15, 2025.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: a dataset for real-world claim verification with evidence from the web. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Sharma and Garg. 2021a. Ifnd dataset. Accessed: 2024-11-07.

Dilip Sharma and Sonal Garg. 2021b. Ifnd: a benchmark dataset for fake news detection. *Complex Intelligent Systems*, 9.

Ronit Singal, Pransh Patwa, Parth Patwa, Aman Chadha, and Amitava Das. 2024. Evidence-backed fact checking using RAG and few-shot in-context learning with LLMs. In *Proceedings of the Seventh Fact Extraction and VERification Workshop (FEVER)*, pages 91–98, Miami, Florida, USA. Association for Computational Linguistics.

Shivangi Singhal, Rajiv Ratn Shah, and Ponnurangam Kumaraguru. 2022. Factdrill: A data repository of fact-checked social media content to study fake news incidents in india. *Proceedings of the International AAAI Conference on Web and Social Media*, 16(1):1322–1331.

Gabriella Skitalinskaya and Henning Wachsmuth. 2023. To revise or not to revise: Learning to detect improvable claims for argumentative writing support. In

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 15799–15816, Toronto, Canada. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Gemma Team. 2024. Gemma.

Ivan Vykopal, Matú Pikuliak, Simon Ostermann, and Marián Simko. 2024. Generative large language models in automated fact-checking: A survey. *ArXiv*, abs/2407.02351.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024. Multilingual e5 text embeddings: A technical report. *ArXiv*, abs/2402.05672.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Wikipedia contributors. 2024. Rss – wikipedia, the free encyclopedia. Accessed: 2025-03-07.

Zhenrui Yue, Huimin Zeng, Lanyu Shang, Yifan Liu, Yang Zhang, and Dong Wang. 2024. Retrieval augmented fact verification by synthesizing contrastive arguments. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10331–10343, Bangkok, Thailand. Association for Computational Linguistics.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xuan Zhang and Wei Gao. 2023. Towards LLM-based fact verification on news claims with a hierarchical step-by-step prompting method. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 996–1011, Nusa Dua, Bali. Association for Computational Linguistics.