# IndicClaimBuster: A Multilingual Claim Verification Dataset

**Pritam Pal, Shyamal Krishna Jana and Dipankar Das**
Jadavpur University, Kolkata, India
{pritampal522, shyamalkrjana516, dipankar.dipnil2005}@gmail.com

## Abstract

The present article introduces **IndicClaimBuster**, a novel multilingual claim verification dataset comprising ≈ 9K claims and their corresponding evidence in English, Hindi, Bengali, and Hindi-English CodeMixed texts. The data set covers three key domains: politics, law and order, and health, to address the challenges of verifiable facts. Each claim was sourced from reputable Indian news portals and is accompanied by three pieces of evidence, two LLM-generated and one manually curated. Additionally, a separate attempt was conducted to generate refuted claims by employing an LLM. We further develop two frameworks: an unsupervised baseline and a two-stage pipeline that comprises evidence retrieval and veracity prediction modules. For retrieval, we fine-tuned SBERT models, with e5-base demonstrating superior average performance across languages, whereas for veracity prediction, multilingual transformers (mBERT, XLM-R, MuRIL, IndicBERTv2) were fine-tuned. Results indicate MuRIL and IndicBERTv2 excel in Indian languages, while XLM-R performs the best for CodeMix. Our work contributes a high-quality multilingual dataset and strong baseline methodologies, offering valuable resources for advancing automated claim verification in linguistically diverse and low-resource settings for Indian languages. The IndicClaimBuster dataset is available at: https://github.com/pritampal98/indic-claim-buster

## 1 Introduction

The proliferation of online misinformation and fake news poses a significant challenge to information integrity. While manual fact-checking remains crucial, its scalability is severely limited by the sheer volume of content, making it an increasingly laborious task to collect evidence and verify claims. The demand for automated fact-checking tools is particularly acute in India, where 33.7% of the population actively uses social media (Kemp, 2025), and a substantial 63.5% of fake news propagates through major social platforms like WhatsApp, Instagram, and Facebook[1].

In recent years, researchers have developed various automated fact-checking tools and datasets (Thorne et al., 2018; Wadden et al., 2020; Saakyan et al., 2021; Aly et al., 2021; Schlichtkrull et al., 2023). Concurrently, several prominent fact-checking websites, such as 'PolitiFact'[2] and 'AltNews'[3], have emerged. However, existing automated fact-checking methodologies and their associated datasets are predominantly focused on the English language. According to our literature, automated claim verification in widely spoken Indian languages, such as Hindi and Bengali, remains largely unexplored. While portals like 'AajTak Bangla' and 'AltNews Hindi' offer fact-checking services for Bengali and Hindi, their processes appear to be primarily manual, as detailed in their methodologies[4][5].

Hindi and Bengali are spoken by millions, with Hindi being the official language of India and Bengali widely used in West Bengal, Tripura, and as the national language of Bangladesh. A unique and significant challenge arises from Code-Mixed texts, which are prevalent on social media platforms (e.g., Twitter, Facebook) and in daily communication. These texts involve a mixture of two or more languages, such as Hindi and English, resulting in complexities including inconsistent grammar, mixed vocabulary, and transliteration issues. Despite the rapid increase in Code-Mixed text usage, automated claim verification in this domain is still largely uncharted territory.

[1] https://bit.ly/fake-news-on-social-media
[2] https://www.politifact.com/
[3] https://www.altnews.in/
[4] https://bangla.aajtak.in/fact-check/methodology
[5] https://www.altnews.in/methodology-for-fact-checking/



*Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pages 2478–2489

Our present research focuses on developing a novel multilingual dataset and a robust framework for claim verification tailored to Indian languages, including English, Hindi, Bengali, and Hindi-English Code-Mixed texts. Leveraging a combination of large language models (LLMs) and meticulous manual human annotation, we have constructed a multilingual dataset containing 9,153 claims, each assigned a veracity label of either SUPPORTS or REFUTES. The key contributions in this paper can be summarized as follows:

- We develop a novel multilingual claim verification dataset, 'IndicClaimBuster', comprising claims in English, Hindi, Bengali, and Code-Mix languages, with evidence sourced from both LLMs and manual curation.
- Further, we propose an automated and cost-effective approach to generate refuting claims from supporting claims, utilizing the GPT-4.1 mini (OpenAI et al., 2024) model.
- Next, we develop a baseline veracity identification framework utilizing the 'Smith-Waterman'(Smith, 1981) algorithm.
- Finally, we establish a two-stage system: an evidence retrieval framework employing a pre-trained Sentence-BERT (SBERT) (Reimers and Gurevych, 2019), followed by a veracity prediction framework utilizing transformer models.

## 2 Related Work

Fact-checking is a time-consuming, yet crucial task that requires a proper understanding of the subject matter. Prior research has addressed various aspects of fact-checking. We categorize existing work into two main areas: high-resource and low-resource settings. In this section, we provide a brief review of the contributions in each category.

Early efforts in high-resource languages, particularly English, include initial benchmark datasets such as those by Popat et al. (2016) and Lim (2018), each comprising approximately 4.5K records. Subsequent datasets, such as the one by Hanselowski et al. (2019) (6.4K records) and the 'WatClaimCheck' dataset by Khan et al. (2022) (33K records), further advanced the field. More recent work has focused on increasing the scale and granularity of the dataset. For instance, Saakyan et al. (2021) developed a dataset of 4K records distinguishing between supporting and refuting claims, while Jiang and Wilson (2021) introduced a 13,696-record dataset aimed at structuring misinformation stories (2,513 information and 11,183 misinformation items). Lee et al. (2023) contributed a dataset of approximately 24,000 records to support data-driven approaches. Larger-scale resources, such as Gangopadhyay et al. (2024)'s dataset of 65K claims, which analyzes claim characteristics and bias, and the comprehensive 'FACTors' dataset by Altuncu et al. (2025), containing 118K claims from 117,993 English reports, exemplify the trend towards extensive high-resource datasets.

Beyond English, research has extended to other languages to address language-specific challenges. For example, Li et al. (2024) focused on fake news detection in Chinese. Political fact-checking bias detection has also attracted attention, with Fernández-Roldán et al. (2023) working on Spanish datasets (313 records) and Lelo (2023) on Portuguese datasets. Moreover, given the global scale of misinformation, multilingual fact-checking approaches have emerged, demonstrated by the work of Gupta and Srikumar (2021) and Quelle et al. (2025).

Despite considerable progress in high-resource languages such as English, Spanish, Chinese, and Portuguese, a notable scarcity of fact-checking datasets remains for Hindi. Similarly, low-resource languages like Bengali lack adequate fact-checking resources. To address this gap, we introduce a well-curated fact-checking dataset, accompanied by thorough analysis, for the Hindi and Bengali languages. Furthermore, recognizing the prevalence of Code-Mixed communication, we also present datasets covering Code-Mixed language, alongside datasets in Hindi, Bengali, and English, thereby contributing a comprehensive resource suite across these languages.

## 3 Development of Dataset

The dataset construction involves four primary stages: 1) Raw data collection, 2) Filtering of Claim Sentences, 3) Evidence gathering, and 4) Data annotation.

### 3.1 Data collection

The English, Bengali, and Hindi Texts were systematically collected from online sources, especially from the headlines of reputed online news portals such as 'Aaj Tak', 'India TV', 'Sangbad Pratidin',

'Live Law', etc., using Python's BeautifulSoup[6] library. The headlines were specifically curated from specific domains of healthcare, politics, and law and order, as the majority of misinformation is spread on the internet from these domains. All the successfully extracted headlines, along with their corresponding language tags, were stored in an Excel spreadsheet for further processing. For consistency, headlines were constrained to a length of at least 10 words.

For Code-Mix data, no news portals were found that have Hindi-English Code-Mix data. Therefore, we utilized a portion of the Code-Mix dataset originally compiled by Nayak and Joshi (2022) for L3Cube. This dataset comprises approximately 40,000 Hindi-English Code-Mixed tweets, all presented in romanized script. However, the scope of this pre-existing dataset did not entirely align with our targeted domains (healthcare, law and order, and politics). To address this, we implemented a filtering methodology using the Llama 3.2 70B (Grattafiori et al., 2024) model. This model was employed to identify and extract tweets from the L3Cube Code-Mix dataset that specifically matched our predefined categories (See Appendix A for exact prompt).

### 3.2 Claim Sentence Filtration

A sentence is considered a claim if it asserts a conclusion or thesis that persuades an audience (Toulmin, 2003). For example, "*Covaxin gets emergency use approval for kids aged 2-18 years*" is a claim as it declares a verifiable event. In contrast, "*How to know if your child is going through some mental health issues?*" is a non-claim as it asks a question rather than asserting a fact.

To identify claims from collected data, eight undergraduate computer science interns were tasked with finding verifiable claims, resulting in 4,757 sentences extracted for evidence collection and verification.

### 3.3 Evidence Gathering

The evidence-gathering phase comprises two distinct methods: evidence collection using LLMs and manual evidence collection.

**Evidence collection using LLM:** Leveraging the state-of-the-art performance of LLMs across various natural language processing tasks, we primarily employed LLMs for automated evidence collection. Two distinct LLMs were selected for this purpose: GPT-4o mini[7] and DeepSeek-V3 (DeepSeek-AI et al., 2025). Each model was accessed via its corresponding API, and the following prompt was utilized for evidence collection:

```
You are an efficient language model
that generates precise evidence for a
given news headline: "{claim}". Provide
a single, well-structured paragraph in
{lang} without repetition. Ensure the
response is within 120 words and does
not contain extra newlines or unnecessary
formatting.

Now, generate the evidence in {lang}
with these constraints.
```

This approach enabled the efficient and scalable retrieval of preliminary evidence, which was subsequently extended to manual evidence collection and the selection of the best evidence to create a robust dataset.

**Manual Evidence Collection:** While LLMs offer an efficient preliminary approach to evidence retrieval, our automated strategy faces inherent limitations, particularly concerning the nuance, contextual understanding, and the critical issue of hallucination. LLMs may generate inaccurate or misleading evidence, a challenge that is particularly pronounced for resource-constrained languages like Bengali and Hindi, as well as for Code-Mixed texts.

To address these limitations and enhance the reliability of our dataset, we employed a team of ten postgraduate human annotators from the Department of Linguistics. These annotators possess advanced proficiency in reading, writing, and comprehending English, Hindi, Bengali, and Code-Mixed texts, enabling them to provide a robust, human-validated layer of evidence collection.

Annotators were instructed to collect concise and informative evidence for each claim, drawing from reputable sources such as government websites, academic journals, and established news portals. The initial guideline stipulated an evidence length between 120 and 150 words to ensure conciseness while capturing essential information.

However, during the annotation phase, it became evident that strictly adhering to this word limit was not always feasible without compromising the comprehensiveness of the evidence. In certain instances, to adequately encompass all critical information necessary to support or refute a claim, the

[6] https://pypi.org/project/beautifulsoup4/

[7] https://platform.openai.com/docs/models/gpt-4o-mini

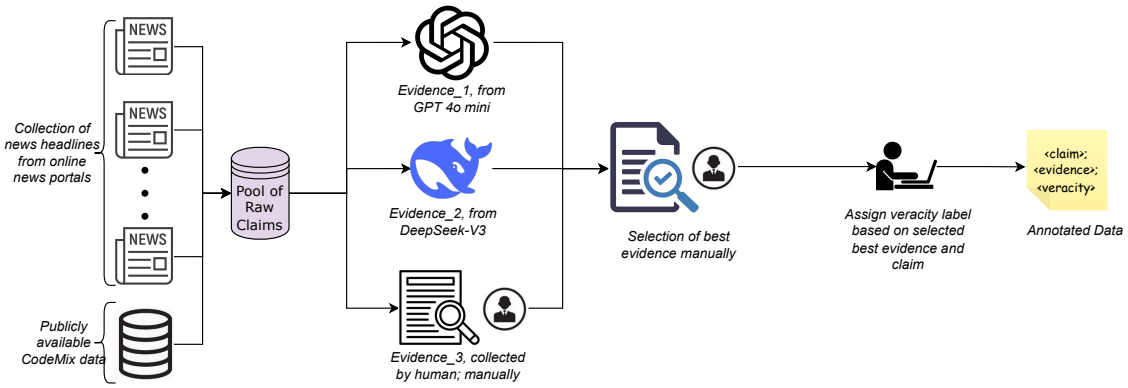


Figure 1: Overview of the data annotation pipeline: from data collection, through LLM-based evidence collection, to manual evidence collection, and finally annotating the best evidence with a veracity label.

word count for a single piece of evidence occasionally exceeded 200 words. This flexibility was permitted to ensure the factual completeness and contextual accuracy of the collected evidence, prioritizing informational integrity over rigid length constraints in such specific cases.

### 3.4 Best Evidence Selection and Veracity Label Annotation

Following the collection of LLM-generated automated pieces of evidence and manual evidence, a separate group of ten annotators from the linguistics department was recruited, selected based on their proficiency in reading and writing of our domain languages, along with demonstrated critical thinking and analytical abilities. The ten annotators were further divided into two groups, each containing five annotators, to identify the best piece of evidence for each claim from three pieces of evidence (two collected by LLMs, and one manually curated), and assign a veracity label as `SUPPORTS` or `REFUTES` to the claim based on the information in the selected evidence. To guide the selection of the best evidence, annotators were instructed to prioritize candidates that met most or all of the following criteria:

- *Context Awareness:* The evidence should be contextually relevant to the claim. Verify that the response accurately represents the context and does not oversimplify complex topics.
- *Detailing and Depth:* The evidence should be detailed, explainable, and contain accurate information. Brief evidence should be rejected. Look for well-structured arguments rather than generic responses.
- *Source of Evidence:* Pick the evidence where proper sources or references are mentioned. Verify whether the sources are genuine.
- *Objectivity and Bias:* The evidence should be neutral and should not contain any political or other type of bias.
- *Logical Consistency:* Select the evidence that logically supports/ refutes the arguments. Avoid pieces of evidence that jump between unrelated ideas and lack structured reasoning.
- *Verification of Named Entities, Dates, and Numbers:* An evidence contains factual information, such as named entities, dates, or numbers (if any), then the evidence can be considered reliable. However, the annotators cross-check named entities, dates, and numbers in the evidence, if available.

During the annotation of best evidence, annotators were asked to rate the best evidence on a scale of 10 to 50, where 10 indicates that the selected best evidence doesn't match the majority of best evidence selection criteria, and 50 indicates that the selected best evidence matches all the best evidence selection criteria.

After identifying the best evidence, annotators proceeded to assign a veracity label based on the selected evidence. Final decisions on both best evidence and veracity labels were made only when there was consensus between the two annotators in the group. Inter-annotator agreement was measured using Fleiss' Kappa (Fleiss, 1971) and Gwet's AC1 (Gwet, 2006). For the best evidence annotation, Fleiss' Kappa was 0.69, and Gwet's AC1 was 0.83, indicating substantial agreement. For the veracity labeling task, Fleiss' Kappa was 0.53 and Gwet's AC1 was 0.90, suggesting strong reliability. Cases with annotation conflicts were subsequently reviewed and resolved by the authors. The overall annotation process is depicted in Figure 1.

### 3.5 Generation of Refute Claims

With the manual annotation process, 4,396 and 361 claim-evidence pairs were annotated with SUPPORTS and REFUTES labels. This is quite justifiable as all the English, Hindi, and Bengali data were collected from reputed news articles; therefore, the number of refuted claim-evidence pairs is too few. Only the majority of refuted claim-evidence pairs were observed in Code-Mix texts (300 claim-evidence pairs). The reason behind this is that the Code-Mix data were originally curated from Twitter; therefore, the number of refuted claim evidence pairs is comparatively higher.

Developing a framework with highly imbalanced data will result in a biased model, specifically for English, Hindi, and Bengali, where a very small number of claim-evidence pairs are refuted.

Therefore, we used a comparatively economic and faster way to generate refute claims from the claim-evidence pairs with SUPPORTS label by altering the critical keywords in the original claim. We utilized the GPT 4.1 mini[8] to generate the refuted claim from the original supported claim. The GPT 4.1 mini model was first instructed to identify important keywords from the claim, such as names, locations, events, quantities, and dates. Next, it was asked to replace these keywords with realistic alternatives and then rewrite the claim in a way that makes it believable and human-like. Along with the prompt, we provided one example for each of the English, Hindi, Bengali, and Code-Mix languages (See Appendix B).

After the process of generating refuted claims, a total of 2,959 claims in English were created. This included 1,479 supportive claims and 1,480 refuted claims. For the other languages, the numbers were as follows: Hindi had 1,393 claims (689 supportive and 704 refuted), Bengali had 2,853 claims (1,404 supportive and 1,449 refuted), and Code-Mix had 1,948 claims (824 supportive and 1,124 refuted). Figure 2 illustrates some examples of original supportive claims alongside their corresponding refuted claims.

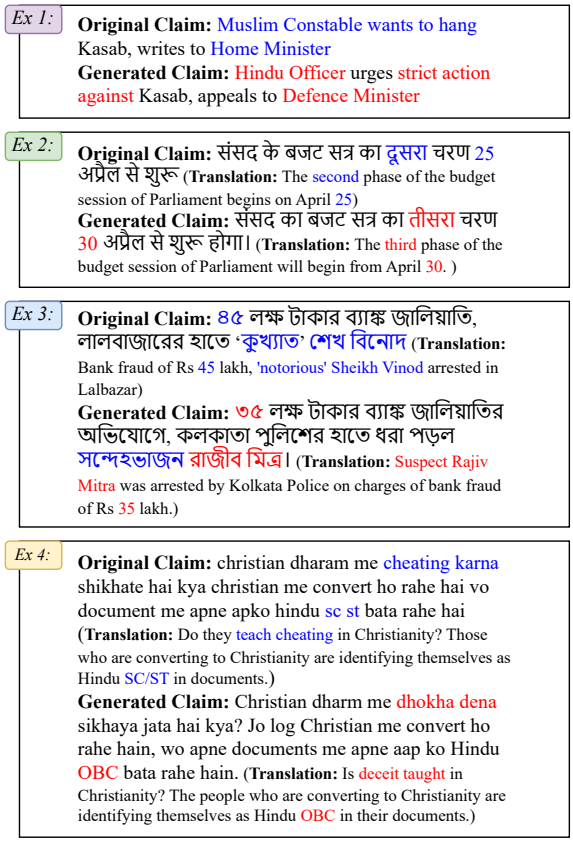


Figure 2: Original claim and its corresponding refuted claim generated by GPT 4.1 mini

## 4 Methodology

We developed two approaches for claim verification using our dataset: an algorithm-based method utilizing the Smith-Waterman algorithm and a two-stage framework that includes evidence retrieval and verification with pre-trained transformer models. Before detailing our methodology, we split the dataset into train and test sets, as shown in Table 1.

| Split | Language | #SUPPORTS | #REFUTES |
|---|---|---|---|
| Train | All | 3502 | 3819 |
| Test | English | 297 | 295 |
| | Hindi | 141 | 138 |
| | Bengali | 280 | 291 |
| | Code-Mix | 176 | 214 |

Table 1: Distribution of test and all training instances.

**Task Formulation:** Given a claim $c_i$ and a comprehensive pool of evidence $E$, which includes all Evidence_1, Evidence_2, and Evidence_3 items from the whole dataset. Each claim is associated with a best evidence $e_{best,i}(e_{best,i} \in E)$ and a corresponding veracity label $v(c_i, e_{best,i}) \in \{\text{SUPPORTS}, \text{REFUTES}\}$. The objective is to:

1. For a given claim $c_i$, retrieve a set of up to five pieces of evidence, denoted as $\{\hat{e}_{1,i}, \ldots, \hat{e}_{k,i}\}; k \in \{1, 3, 5\}$, from the evidence pool $E$.

[8] https://platform.openai.com/docs/models/gpt-4.1-mini

2. Predict the veracity label of the claim $c_i$ utilizing the retrieved evidence $\{\hat{e}_{1,i}, \ldots, \hat{e}_{k,1}\}$ as SUPPORTS or REFUTES. This can be mathematically formulated as $\hat{v}(c_i, \{\hat{e}_{1,i}, \ldots, \hat{e}_{k,i}\} \in \{\text{SUPPORTS}, \text{REFUTES}\}$.

### 4.1 Baseline System

This section presents an algorithm-based baseline system for veracity prediction using claims and annotated best evidence pairs from the dataset. The core of the algorithm operates on pairwise sequence alignment of claim and evidence pairs. To determine the pairwise sequence alignment score, we employ the Smith-Waterman sequence alignment algorithm 1. The evaluation of the evidence follows a series of steps, which are outlined in algorithm 2.

**Algorithm 1** SmithWaterman$_{\text{Modified}}$

**Require:** $seq_1$, $seq_2$
$m \leftarrow size(seq_1)$, $n \leftarrow size(seq_2)$
$scoring_{mat}[m+1][n+1]$ *of floating value*
$match \leftarrow lagn_{\text{match}}$, $gap \leftarrow -1$, $mismatch \leftarrow -1$, $score \leftarrow 0$, $maxscore \leftarrow 0$
**for** *i from 0 to (m+1)* **do**
  $scoring_{mat}[i][0] \leftarrow 0$
**end for**
**for** *j from 0 to (n+1)* **do**
  $scoring_{mat}[0][j] \leftarrow 0$
**end for**
**for** *i from 0 to (m+1)* **do**
  **for** *j from 0 to (n+1)* **do**
    **if** $seq_1[i-1] = seq_2[j-1]$ **then**
      $score \leftarrow match$
    **else**
      $score \leftarrow mismatch$
    **end if**
    $scoring_{mat}[i][j] \leftarrow max(0, scoring_{mat}[i][j-1] + gap, scoring_{mat}[i-1][j] + gap, scoring_{mat}[i-1][j-1] + score)$
    **if** $maxscore \leq scoring_{mat}[i-1][j-1]$ **then**
      $maxscore \leftarrow scoring_{mat}[i][j]$
    **end if**
  **end for**
**end for**
**return** *maxscore*

The Smith-Waterman algorithm has three primary parameters: match, mismatch, and gap penalty. When a match is found, it generates a score, with match-score values being language-specific. For English, Hindi, Bengali, and Code-Mix, the match-score value was fixed to specific values of 2.5, 3, 3, and 8, respectively. These match scores were determined through trial and error using the training data.

This score is represented by the variable $lagn_{match}$. In cases of mismatches or gaps between pairs, the algorithm deducts a small fixed amount from the score; both gap and mismatch penalties are set to -1 for all language types. The final score is stored in a designated variable.

**Algorithm 2** Claim Verification with Evidence scoring

**Require:** dataset
**Require:** $score_e[size(dataset)]$ *of floating values*
$org \leftarrow dataset[label]$
**for** *i from* 0 *to* $size(dataset)$ **do**
  $c_t, e_t \leftarrow Tokenize(dataset[claim][i], dataset[evidence][i])$ ▷ [word index tokenization]
  $score \leftarrow SmithWaterman_{Modified}(c_t, e_t)$
  $score_e[i] \leftarrow \frac{score}{size(c_t)}$
  $i \leftarrow i + 1$
**end for**
**for** *i from* 0 *to* $size(score_e)$ **do**
  **if** $\tau \leq score_e[i]$ **then**
    $pred \leftarrow 1$
  **else**
    $pred \leftarrow 0$
  **end if**
**end for**
$prec \leftarrow Precision\_Score(pred, org)$,
$rec \leftarrow Recall\_Score(pred, org)$
$f1 \leftarrow F1\_Score(pred, org)$

From Algorithm 1, after getting the $maxscore$ from the Smith-Waterman algorithm, it was evaluated in Algorithm 2 and applied a threshold value ($\tau = 0.5$) to classify the predicted $score_e$ and get the predicted label of the claim. Finally, the performance was evaluated by comparing the predicted labels with the original labels.

### 4.2 Two Stage Framework

This section provides a brief overview of the two-stage framework, comprising evidence retrieval and veracity prediction.

#### 4.2.1 Evidence Retrieval

The evidence retrieval framework utilizes pre-trained multilingual SBERT models, specifically 'LaBASE' (Feng et al., 2022), 'multilingual-e5-base' (Wang et al., 2024), 'multilingual-mpnet-base' [9], and 'xlm-r' [10]. These models were fine-tuned on the training dataset to improve their ability to understand the semantic similarity between claims and evidence pairs. The goal of this fine-tuning process was to learn an embedding function, represented as $SBERT(\cdot)$, that maps semantically similar texts to close points in a high-dimensional vector space.

For fine-tuning, the training samples were constructed to allow the models to distinguish between relevant and irrelevant pieces of evidence specifically. Given a claim $c_i$ from the training set, which

[9] https://bit.ly/paraphrase-multilingual-mpnet-base-v2
[10] https://bit.ly/paraphrase-xlm-r-multilingual-v1

was associated with its best evidence $e_{best,i}$, a positive pair $(c_i, e_{best,i})$ was assigned a label of $y = 1$, indicating high semantic similarity. To facilitate the model's discriminative learning, a negative pair was created by randomly sampling an evidence piece $e_{rand,i}(e_{rand,i} \in E)$ from the evidence pool $E$ where $e_{rand,i} \neq e_{best,i}$. This negative pair $(c_i, e_{rand,i})$ was assigned a label of $y = 0$, with a significantly low semantic similarity.

The `CosineSimilarityLoss` was utilized to maximize the cosine similarity between the embeddings of positive pairs (where $y = 1$) and minimize the cosine similarity between the embeddings of negative pairs (where $y = 0$). The optimization was performed using the `AdamW` (Loshchilov and Hutter, 2019) optimizer, configured with a learning rate of 2e-5, a weight decay of 0.01, and an epsilon value of 1e-8.

Following the fine-tune process, the entire evidence pool $E$ was transformed to a scalable embedding space. This was achieved by encoding the evidence piece $e_j \in E$ into a dense vector embedding $V_{e_j} = SBERT(e_j)$ using the finetuned SBERT model. For a given query claim $c_q$, the evidence retrieval framework operates as follows: 1) The query claim $c_q$ is first encoded into its vector embedding $V_{c_q} = SBERT(c_q)$. 2) The cosine similarity is then computed between the claim's embedding $V_{c_q}$ and the embedding of every evidence piece $V_{e_j}$ in the entire evidence pool $E$. This yields a set of similarity scores $score_j = cos\_sim(V_{c_q}, V_{e_j} | e_j \in E)$. 3) Finally, the framework returns top $k$ evidence pieces from the evidence pool $E$ that exhibit the highest cosine similarity scores with the claim $c_q$.

#### 4.2.2 Veracity Prediction

The veracity prediction framework employs several pre-trained multilingual transformer models, specifically `multilingual BERT` (mBERT) (Devlin et al., 2018), `XLM-RoBERTa` (XLM-R) (Conneau et al., 2019), `MuRIL` (Khanuja et al., 2021), and `IndicBERTv2` (Doddapaneni et al., 2023). While mBERT and XLM-R were pre-trained on a broad spectrum of languages (104 and 100 languages, respectively, including Indian languages such as Hindi and Bengali), MuRIL and IndicBERTv2 were specifically designed and trained with Indian linguistic contexts in mind. This specialized training enables MuRIL and IndicBERTv2 to process Indian languages more effectively than mBERT and XLM-R.

**Framework Description:** The input to the framework consists of a claim $c_i$ concatenated with a set of retrieved evidence pieces $\{\hat{e}_{1,i}, \ldots, \hat{e}_{k,i}\}$. To distinctly separate the claim and evidence components for the model, a specific formatting scheme was applied. The claim portion was prefixed with '`[CLAIM]:`' followed by two newline characters (\n\n), which then precede '`[EVIDENCE]:`' and the evidence text.

The concatenated input was subsequently tokenized into a fixed sequence length of 512 tokens. Tokenization was performed using the tokenizer corresponding to the specific transformer model. These tokenizers generate sequences of input IDs and attention masks, which were then fed as input to their respective transformer encoders.

Following the transformer encoder, the pooled output was derived from the last hidden state representation of the special starting token (e.g., `[CLS]` for mBERT, MuRIL, and IndicBERTv2, and `<s>` for XLM-R model)[11]. This pooled output was obtained by applying a learned linear transformation followed by a $tanh$ activation function. The resulting pooled representation then undergoes a dropout layer with a rate of 0.2.

**Classification:** For classification, the output of the dropout layer was passed through an output layer of 2 hidden units, corresponding to the number of veracity classes (SUPPORTS/REFUTES). This output layer utilizes the softmax activation function to produce class probabilities.

**Training:** For training, the training data was split into 90% for training and 10% for validation. The `SparseCategoricalCrossEntropy` loss function was used with a learning rate of 2e-5, and optimization was done with the `AdamW` optimizer, set to a weight decay of 0.01 and epsilon of 1e-8. The frameworks were trained for up to 10 epochs with a mini-batch size of 8.

## 5 Experiment and Results

This section summarizes the experimental setting and outcomes for the evidence retrieval and veracity prediction task on the test data.

### 5.1 Experimental Setup

All models for evidence retrieval and veracity prediction were trained using PyTorch and TensorFlow, utilizing an NVIDIA RTX-5000 GPU. Pretrained transformer models and tokenizers were

[11] `[CLS]` and `<s>` are special tokens that typically represent the aggregate sequence information.

obtained from the HuggingFace library. The fine-tuned evidence retrieval models were evaluated with Success@k for $k \in \{1, 3, 5\}$, while the veracity prediction models' performance was measured using the macro-averaged F1-score.

$$\text{Success@k} = \begin{cases} 1 & \text{if } E(c) \in \{\hat{e}_1, \hat{e}_2, \ldots, \hat{e}_k\} \\ 0 & \text{otherwise} \end{cases}$$

During training, only gold evidence (the best evidence as determined by annotators) was used for veracity label prediction. Evaluation utilized three sources: gold evidence, top-1 retrieved evidence, and top-3 retrieved evidence from the best evidence retrieval model. The algorithm-based approach was assessed only with gold evidence.

### 5.2 Results

**Evidence Retrieval:** The results of the Evidence Retrieval task, shown in Table 2, indicate that the `e5-base` model performs best overall for English and Bengali. For English, it achieved Success@5, Success@3, and Success@1 scores of 70.10%, 58.61%, and 25.52%, while for Bengali, the scores were 59.02%, 51.14%, and 29.77%.

| Model | Language | S@5 | S@3 | S@1 |
|---|---|---|---|---|
| LaBASE | English | 62.16 | 51.18 | 25.68 |
| | Hindi | **82.44** | **69.18** | 24.73 |
| | Bengali | 54.14 | 47.99 | 26.21 |
| | Code-Mix | 23.59 | 18.72 | **11.03** |
| e5-base | English | **70.10** | **58.61** | **26.52** |
| | Hindi | 78.14 | 68.46 | **45.52** |
| | Bengali | **59.02** | **51.14** | **29.77** |
| | Code-Mix | **24.10** | **18.72** | 10.51 |
| MPNet | English | 59.80 | 49.16 | 21.28 |
| | Hindi | 69.89 | 51.97 | 19.71 |
| | Bengali | 34.85 | 27.85 | 15.41 |
| | Code-Mix | 23.85 | 18.46 | 8.72 |
| XLM-R | English | 61.66 | 51.01 | 24.16 |
| | Hindi | 67.03 | 54.48 | 20.43 |
| | Bengali | 33.10 | 27.50 | 15.94 |
| | Code-Mix | 17.18 | 14.36 | 7.18 |

Table 2: Result for evidence retrieval (S@5, S@3 and S@1 represents the Success@5, Success@3 and Success@1 scores respectively.)

In Hindi, the `LaBASE` model excelled, achieving Success@5 and Success@3 scores of 82.44% and 69.18%, respectively. In contrast, the `e5-base` model had the highest Success@1 score at 45.52%. For Code-Mix, `e5-base` achieved the best Success@5 and Success@3 scores of 24.10% and 18.72%, but `LaBASE` outperformed it at Success@1 with 11.03%. Overall, evidence retrieval in Code-Mix was notably poor due to its complex combination of Hindi and English, which complicated the retrieval process.

Aggregating performance across all languages, the `e5-base` model yielded the best average scores with Success@5, Success@3, and Success@1 at 57.84%, 49.23%, and 28.08%, respectively. In comparison, the `LaBASE` model achieved average Success@5, Success@3, and Success@1 scores of 55.58%, 46.77%, and 21.91%, respectively. Given its superior average performance across diverse linguistic contexts, the `e5-base` model is designated as the most effective evidence retrieval model. Consequently, the subsequent assessment of veracity label prediction performance was conducted using evidence retrieved by the `e5-base` model.

| Model | Language | F1-Score | | |
|---|---|---|---|---|
| | | Gold | Top-3 | Top-1 |
| Baseline system (algorithm base) | English | 68.41 | - | - |
| | Hindi | 69.38 | - | - |
| | Bengali | 70.55 | - | - |
| | Code-Mix | 56.27 | - | - |
| mBERT | English | 92.21 | 89.18 | 89.16 |
| | Hindi | 90.67 | 89.96 | 87.06 |
| | Bengali | 89.13 | 84.90 | 82.82 |
| | Code-Mix | 78.56 | 69.42 | **70.15** |
| XLM-R | English | 93.58 | 90.02 | 89.86 |
| | Hindi | 94.26 | 91.76 | 89.94 |
| | Bengali | 89.48 | 86.32 | 82.24 |
| | Code-Mix | **84.55** | **71.03** | 69.58 |
| MuRIL | English | **93.92** | **92.23** | **91.89** |
| | Hindi | **95.70** | **92.11** | **91.38** |
| | Bengali | **91.94** | **87.03** | **83.31** |
| | Code-Mix | 84.23 | 63.57 | 65.57 |
| IndicBERTv2 | English | 93.91 | 91.05 | 90.37 |
| | Hindi | 94.96 | 90.32 | 89.60 |
| | Bengali | 91.94 | 86.86 | 83.06 |
| | Code-Mix | 80.10 | 62.21 | 68.81 |

Table 3: Result for veracity prediction

**Veracity Prediction:** The veracity prediction results shown in Table 3 indicate that the best overall performance is achieved when using gold evidence annotated by human annotators.

For the English, Hindi, and Bengali languages, the `MuRIL`-based framework consistently outperforms other models across all evaluation settings. This is closely followed by `IndicBERTv2`, `XLM-R`, and `mBERT`. This suggests that models pre-trained on Indian linguistic contexts are more effective than general multilingual models.

In contrast, for Code-Mix, the `XLM-R`-based

framework achieves the highest performance, with F1-scores of 84.55% for gold evidence and 71.03% for the top-3 retrieved evidence. Meanwhile, the highest score for top-1 retrieved evidence is obtained by mBERT at 70.15%.

The performance of the algorithmic approach is relatively good. With a simple algorithm, the baseline scores for English, Hindi, and Bengali are 68.41%, 69.38%, and 70.55%, respectively. However, for Code-Mix texts, the F1-score decreases slightly to 56.27%.

## 6 Conclusion

This paper introduces 'IndicClaimBuster', a novel multilingual claim verification dataset comprising 9,153 claims in English, Hindi, Bengali, and Code-Mixed languages. The dataset addresses three crucial fact-checking domains: Politics, Law and Order, and Health. It includes real-world claims that have been carefully sourced from reputable Indian news portals, along with a counterclaim generated by GPT-4.1 mini for each supporting claim.

Our experimental results indicate that the e5-base and LaBASE models achieve the highest performance in the evidence retrieval task. For veracity prediction, the MuRIL and IndicBERTv2 models perform exceptionally well across Indian languages (English, Hindi, and Bengali), while the XLM-R model excels specifically in the Code-Mix language. Additionally, the algorithm-based approach provides an acceptable baseline performance across all languages.

## Limitations

The proposed work has several limitations. First, while social media platforms like X (formerly Twitter) are primary sources of misinformation, their paid APIs prevent direct raw data collection from such platforms. Consequently, we opted for a cost-effective alternative by collecting data from news headlines via web scraping. Although Reddit was considered a potential alternative, it currently lacks sufficient data relevant to our predefined domains in Bengali, Hindi, and CodeMix. Future efforts will involve a more rigorous exploration of Reddit to identify suitable posts in these languages and domains.

Second, for economic reasons, we utilized 'mini' versions of GPT models, specifically GPT-4o mini and GPT-4.1 mini, for evidence collection and refuted claim generation, respectively. While these mini versions yielded promising results, their use inherently limits the full capabilities of the more powerful GPT models, which could potentially provide superior output quality.

Third, we only retrieve evidence from a specific pool, which consists of Evidence 1, Evidence 2, and Evidence 3 for all claims in the dataset. For a completely new claim, it is almost impossible to find suitable evidence from the existing pool, as its size is fixed. However, we plan to implement a Retrieval-Augmented Generation (RAG) system or a Google Search-based retrieval system to handle new claims in our future work.

Fourth, despite India's rich linguistic diversity, encompassing 22 scheduled and 38 non-scheduled languages[12], our dataset currently incorporates only Hindi and Bengali due to existing linguistic barriers and resource constraints. Our future research aims to expand this linguistic coverage to include more Indian languages, such as Assamese and Odia.

Fifth, we introduce algorithm-based baseline systems that utilize only the gold-standard annotated evidence. In future work, we plan to extend this to Top-3 and Top-1 evidence retrieval systems.

Finally, due to resource limitations, all evidence retrieval and veracity prediction frameworks were fine-tuned with a mini-batch size of 8. This constraint prevented us from exploring the potential benefits of fine-tuning models with larger batch sizes (e.g., 16, 32, or 64), which could potentially enhance model performance.

## Acknowledgment

This work was supported by the Defence Research and Development Organisation (DRDO), New Delhi, under the project "Claim Detection and Verification using Deep NLP: an Indian perspective".

## References

Enes Altuncu, Can Başkent, Sanjay Bhattacherjee, Shujun Li, and Dwaipayan Roy. 2025. Factors: A new dataset for studying the fact-checking ecosystem. *arXiv preprint arXiv:2505.09414*.

Rami Aly, Zhijiang Guo, Michael Sejr Schlichtkrull, James Thorne, Andreas Vlachos, Christos Christodoulopoulos, Oana Cocarascu, and Arpit Mittal. 2021. The fact extraction and VERification

[12] https://www.mha.gov.in/sites/default/files/EighthSchedule_19052017.pdf

over unstructured and structured information (FEVEROUS) shared task. In *Proceedings of the Fourth Workshop on Fact Extraction and VERification (FEVER)*, pages 1–13, Dominican Republic. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

DeepSeek-AI, Aixin Liu, Bei Feng, and et al. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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.

Sumanth Doddapaneni, Rahul Aralikatte, Gowtham Ramesh, Shreya Goyal, Mitesh M. Khapra, Anoop Kunchukuttan, and Pratyush Kumar. 2023. Towards leaving no indic language behind: Building monolingual corpora, benchmark and models for indic languages. *Preprint*, arXiv:2212.05409.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.

Alejandro Fernández-Roldán, Carlos Elías, Carlos Santiago-Caballero, and David Teira. 2023. Can we detect bias in political fact-checking? evidence from a spanish case study. *Journalism practice*, pages 1–19.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Susmita Gangopadhyay, Sebastian Schellhammer, Salim Hafid, Danilo Dessi, Christian Koß, Konstantin Todorov, Stefan Dietze, and Hajira Jabeen. 2024. Investigating characteristics, biases and evolution of fact-checked claims on the web. In *Proceedings of the 35th ACM Conference on Hypertext and Social Media*, pages 246–258.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, and et al. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Ashim Gupta and Vivek Srikumar. 2021. X-fact: A new benchmark dataset for multilingual fact checking. *arXiv preprint arXiv:2106.09248*.

Kilem Li Gwet. 2006. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48.

Andreas Hanselowski, Christian Stab, Claudia Schulz, Zile Li, and Iryna Gurevych. 2019. A richly annotated corpus for different tasks in automated fact-checking. *arXiv preprint arXiv:1911.01214*.

Shan Jiang and Christo Wilson. 2021. Structurizing misinformation stories via rationalizing fact-checks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 617–631.

Simon Kemp. 2025. Digital 2025: India — DataReportal – Global Digital Insights.

Kashif Khan, Ruizhe Wang, and Pascal Poupart. 2022. Watclaimcheck: A new dataset for claim entailment and inference. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1293–1304.

Simran Khanuja, Diksha Bansal, Sarvesh Mehtani, Savya Khosla, Atreyee Dey, Balaji Gopalan, Dilip Kumar Margam, Pooja Aggarwal, Rajiv Teja Nagipogu, Shachi Dave, Shruti Gupta, Subhash Chandra Bose Gali, Vish Subramanian, and Partha Talukdar. 2021. Muril: Multilingual representations for indian languages. *Preprint*, arXiv:2103.10730.

Sian Lee, Aiping Xiong, Haeseung Seo, and Dongwon Lee. 2023. "fact-checking" fact checkers: A data-driven approach. *Harvard Kennedy School Misinformation Review*.

Thales Lelo. 2023. Assessing the consistency of fact-checking in political debates. *Journal of Communication*, 73(6):587–600.

Yupeng Li, Haorui He, Jin Bai, and Dacheng Wen. 2024. Mcfend: A multi-source benchmark dataset for chinese fake news detection. In *Proceedings of the ACM Web Conference 2024*, pages 4018–4027.

Chloe Lim. 2018. Checking how fact-checkers check. *Research & politics*, 5(3):2053168018786848.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. *Preprint*, arXiv:1711.05101.

Ravindra Nayak and Raviraj Joshi. 2022. L3Cube-HingCorpus and HingBERT: A code mixed Hindi-English dataset and BERT language models. In *Proceedings of the WILDRE-6 Workshop within the 13th Language Resources and Evaluation Conference*, pages 7–12, Marseille, France. European Language Resources Association.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, and et al. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Kashyap Popat, Subhabrata Mukherjee, Jannik Strötgen, and Gerhard Weikum. 2016. Credibility assessment of textual claims on the web. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, pages 2173–2178.

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.

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*. Association for Computational Linguistics.

Arkadiy Saakyan, Tuhin Chakrabarty, and Smaranda Muresan. 2021. COVID-fact: Fact extraction and verification of real-world claims on COVID-19 pandemic. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2116–2129, Online. Association for Computational Linguistics.

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. Averitec: A dataset for real-world claim verification with evidence from the web. *Preprint*, arXiv:2305.13117.

T Smith. 1981. Smith-waterman algorithm. *Advances in Applied Mathematics*, 2:482–489.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. FEVER: a large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana. Association for Computational Linguistics.

Stephen E. Toulmin. 2003. *The uses of argument*.

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online. Association for Computational Linguistics.

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

# A Appendix A: Prompt for topic identification using Llama 3.2 70B

```
You are an efficient
language model that can
precisely find topic in an
hinglish codemix text
(mixing of Hindi and
English in romanized
script).

Now, provide the topic of
'{text}' in English
language

The topics should be in
between {{'politics',
'indian election',
'national', 'hindu-muslim',
'sports', 'entertainment',
'unknown'}}

Remember only generate the
exact output. No exatra
explanation is needed
```

Prompt for topic identification utilizing the Llama 3.2 70B model. The selected topics, derived from primary dataset analysis, exclude the healthcare topic due to the absence of relevant data. The 'indian election' and 'hindu-muslim' topics were considered under the law and order domain.

## B Appendix B: Prompt for generating refute claim using GPT 4.1 mini

```
You are given a news text **claim**.

Your task:
1. Identify important keywords and
numeric values (e.g., names, locations,
events, quantities, dates).
2. Replace them with realistic
alternatives (plausible names, numbers,
places, events).
3. Do **not entirely change** the meaning
— just reword the original into a
**realistic counter claim**.
4. The rewritten claim should be
believable and human-like.
5. Output **only the regenerated counter
claim** — no explanations, no keyword
lists.

---

### Examples:

**English**
Input: "The health minister inaugurated a
300-bed hospital in Mumbai on Sunday."
Output: "The education minister
inaugurated a 250-seat research institute
in Pune on Monday."

**Hindi**
Input: "राजस्थान सरकार ने 1000 करोड़ का बजट
सिर्फ महिला सुरक्षा के लिए रखा।"
Output: "मध्य प्रदेश सरकार ने 800 करोड़ का बजट
किसानों के लिए निर्धारित किया।"

**Bengali**
Input: "সরকার আগামী বছরে ৫০ হাজার শিক্ষক
নিয়োগের কথা ঘোষণা করেছে।"
Output: "সরকার আগামী বছরে ৩০ হাজার
স্বাস্থ্যকর্মী নিয়োগের পরিকল্পনা করছে।"

**Hindi-English CodeMix (Roman script)**
Input: "Delhi me 200 logon ne protest
march organize kiya Parliament ke
saamne."
Output: "Lucknow me 150 students ne
silent rally kiya CM Office ke saamne."

---

Now re-write the following claim using
the above instructions.

**Claim**: {claim}
"""
```