Health Sentinel: An AI Pipeline For Real-time Disease Outbreak Detection

Devesh Pant[1†], Rishi Raj Grandhe[2], Jatin Agrawal[2], Jushaan Singh Kalra[2],
Sudhir Kumar[1], Saransh Khanna[1], Vipin Samaria[1], Mukul Paul[1],
Dr. Satish V Khalikar[1], Vipin Garg[1], Dr. Himanshu Chauhan[3], Dr. Pranay Verma[3],
Akhil VSSG[2], Neha Khandelwal[2], Soma S Dhavala[2], Minesh Mathew[1†]

[1]Wadhwani AI, India, [2]Work done while working at Wadhwani AI
[3]National Centre for Disease Control, Government of India
[†]{devesh, minesh}@wadhwaniai.org

Abstract

Early detection of disease outbreaks is crucial to ensure timely intervention by the health authorities. Due to the challenges associated with traditional indicator-based surveillance, monitoring informal sources such as online media has become increasingly popular. However, owing to the number of online articles getting published everyday, manual screening of the articles is impractical. To address this, we propose Health Sentinel. It is a multi-stage information extraction pipeline that uses a combination of ML and non-ML methods to extract events—structured information concerning disease outbreaks or other unusual health events—from online articles. The extracted events are made available to the Media Scanning and Verification Cell (MSVC) at the National Centre for Disease Control (NCDC), Delhi [1] for analysis, interpretation and further dissemination to local agencies for timely intervention. From April 2022 till date, Health Sentinel has processed over 300 million news articles and identified over 95,000 unique health events across India of which over 3,500 events were shortlisted by the public health experts at NCDC as potential outbreaks.

1 Introduction

Disease surveillance is the continuous collection, analysis and interpretation of health data, particularly data concerning disease outbreaks and other unusual health events. It is essential that disease surveillance collects information in real time for timely interventions. Further, continuous disease surveillance allows us to monitor disease spread patterns and allocate resources more effectively, directing attention and funding to areas where they are most needed.

Traditional surveillance approaches follow a bottom-up approach wherein information is collected from health care workers, public health fa-
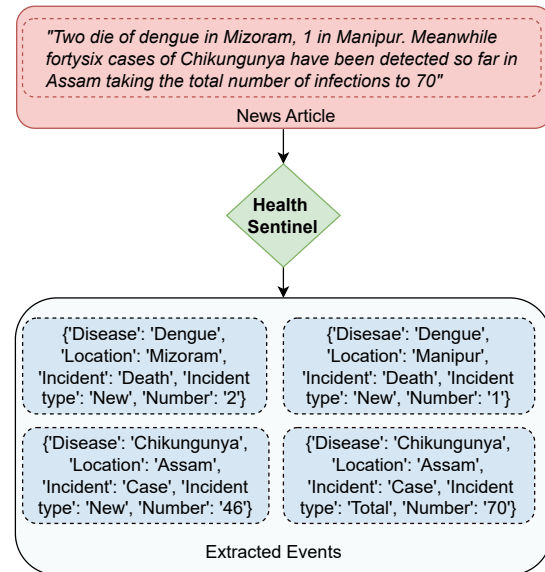

Figure 1: Health Sentinel extracts structured information from online articles reporting unusual health events. The given example shows how our pipeline extracts multiple events from a single news article.

cilities and hospital networks. This approach is commonly referred to as 'Indicator-based surveillance'. It typically involves confirmed case reports, laboratory results, and clinical diagnoses. While indicator-based surveillance ensures that the data collected is mostly reliable, delay in reporting is often a concern. Further, weaker public health systems, and under-reporting particularly in remote and rural areas make indicator-based surveillance challenging (WHO et al., 2008).

In contrast to the indicator-based surveillance, event-based surveillance looks at multiple sources of information either formal or informal such as print media reports, online articles, and social media posts. This approach is designed to detect unusual health events quickly, providing early warnings for potential outbreaks. Owing to the nature of sources, data collected in this approach is likely

[1]Media Scanning and Verification Cell (MSVC)

to be noisy, redundant, and unstructured. Consequently, Information Retrieval (IR) and Natural Language Processing (NLP) techniques are increasingly being used in event-based surveillance systems to filter out irrelevant and redundant data to extract cohesive structured information. (Pilipiec et al., 2023; Valentin, 2020; Abbood et al., 2020; Huff et al., 2016; Valentin et al., 2021)

According to the 2011 census, India has a population exceeding 1.2 billion (of the Registrar general and census commissioner of India). Most of the neglected tropical diseases are prevalent in India. Health threats triggered by climate change (Romanello et al., 2023, 2024) is another significant concern. Health governance in India is decentralized where states hold primary responsibility for healthcare. Traditional, indicator-based disease surveillance in such a setting demands a well coordinated system involving stakeholders belonging to different departments and different state governments.

In this work, we present Health Sentinel, an information extraction pipeline, that feeds structured information concerning public health events to an event-based surveillance system in India. It extracts events related to 122 human and animal diseases which were prioritized based on inputs from public health experts.

As shown in Figure 2, Health Sentinel follows a multi-stage process. It begins with data ingestion where articles are periodically crawled from the web. Followed by this step, a binary text classifier filters out irrelevant articles. Next, all articles are translated to English. Once an article is identified to carry information on one or more unusual health events, we extract structured information from it. This is referred to as Event Extraction (EE) (Xiang and Wang, 2019) in Information Retrieval (IR) and Natural Language processing (NLP). In our case, an "event" comprises, i) Disease- the specific disease or "others" if the disease is not among a predefined list of 122 diseases our health experts have curated, ii) Location - the geographical area where the disease occurrence is reported, iii) Incident - the nature of the event, such as case or death , iv) Incident type - whether the incident is New or Total (cumulative), and v) Number - the numerical value associated with the incident and its type (number of cases or number of fatalities). An example where 4 distinct events are extracted from a single article is shown in Figure 1. Followed by event extraction, similar events are clustered together to isolate

unique event occurrences. The unique events are finally passed on to an expert for further review.

The highlights of Health Sentinel are listed as follows:

1. Health Sentinel, unlike most existing systems, scans the entire internet scouting for unusual health events.

2. To the best of our knowledge, Health Sentinel is the first system that supports media scanning in multiple Indian languages. It supports 13 languages: English, Hindi, Telugu, Kannada, Gujarati, Tamil, Punjabi, Bengali, Marathi, Malayalam, Oriya, Assamese, and Urdu.

3. We demonstrate that LLMs, including the recent open-source models perform better for event extraction compared to the previously popular approaches like Named Entity Recognition (NER) and Question Answering (QA).

4. We propose a clustering logic that uses language model embeddings for text similarity and DFS search on a graph built based on pairwise similarities and curated rules.

5. Since its inception in April 2022, Health Sentinel has identified over 95,000 unique health events of which over 3,500 events were shortlisted by public health experts at NCDC.

6. The datasets used in this work will be made publicly available at https://github.com/WadhwaniAI/Health-Sentinel.

2   Related Work

Most ML-based disease surveillance approaches employ rule-based techniques along with classical ML models (Cabatuan and Manguerra, 2020; Zeng et al., 2021) . MediSys (J et al., 2010) and ProMED (Yu and Madoff, 2004) are the two most popular disease surveillance systems. MediSys uses pattern-matching techniques to extract events from articles which leads to many false positives. Since it is rule-based, extending it to other languages is non-trivial. On the other hand, in ProMED, filtering relevant information and further analysis is mostly performed by involving humans. In recent times, new tools like GRITS (Huff et al., 2016) , EventEpi (Abbood et al., 2020) and Padi 3.0 (Valentin et al., 2021) have been developed for disease surveillance. Although these systems use ML for tasks like classification and clustering, none of them exploit recent advances in event extraction using deep learning

techniques and LLMs.

EE, or extracting structured information from unstructured text, is a well studied problem in IR and NLP. Deep learning-based NER and QA models have extensively been used for EE tasks (Xiang and Wang, 2019). However, LLMs are increasingly being used for EE tasks in zero-shot, few-shot and finetuned settings (Simon et al., 2024). LLMs have demonstrated impressive results in information extraction tasks using few-shot approach without requiring task-specific fine-tuning (Kojima et al., 2023). These results have further been validated by studies exploring use of OpenAI models like GPT-3.5 and GPT-4[2] for EE in various scenarios (Wei et al., 2024; Polak and Morgan, 2024; Gao et al., 2023). Similar to these works, we use LLMs for few-shot EE. Dagdelen et al. demonstrate that GPT-3 and Llama-2 (Touvron et al., 2023) can be finetuned for extracting structured information from scientific text. We have not explored LLM finetuning for EE for disease surveillance owing to the lack of training data. Harrod et al. (Harrod et al., 2024) use LLMs for extraction of structured epidemiological data from documents and geotagging each record. Their work is similar to ours as they extract structured disease related information and use LLMs for the same. However, their work focuses on extracting information concerning Rift Valley Fever (RVF) alone. Secondly, the objective of their work is not disease surveillance but creation of a structured epidemiological dataset for RVF from past documents—PDFs of research articles and other documents concerning RVF. In contrast, our work uses LLMs for event extraction from web articles for real time surveillance of 100+ diseases.

3 Method

In the following sections, we present details of each stage in the Health Sentinel system. The overall flow can be referred to in Figure 2.

3.1 Data Ingestion

For a real-time system, it is essential to continuously monitor the web for newly added articles. Health Sentinel achieves this using three services: Common Crawl[3], Google Alerts[4] and custom crawlers. We use Common Crawl's news

database to fetch the latest published articles every few hours. We configured Google Alerts using keywords in 12 Indic languages and English. The keywords for Google Alerts were selected by public health experts based on the 122 disease/health events that we are interested in monitoring. We have additionally designed custom crawlers for a few news websites that are not covered by the former two services.

From the URLs of the news articles collected through these sources, 'title' and 'description' tags are extracted. These tags provide concise information about the webpage content, such as headline of the news and a summary of the article body. Given the extremely wide scope of HTML-based webpages, it is infeasible to effectively extract the relevant content of the webpage from its body while isolating noisy information like advertisements. Therefore, for further processing, the text used from an article is the concatenation of the 'title' and 'description' fields.

The articles undergo a rule-based filtering based on three criteria: a domain blocklist, recency (only the most recently published articles are retained), and language. The block-list contains domain names of non-Indian news websites, allowing us to filter out more than 90% of irrelevant articles that cover news outside India. The source language of an article is identified using langid (Lui and Baldwin, 2012), and only articles in the 13 supported languages are retained.

3.2 Article Classification

A substantial portion (nearly 87%) of the articles at this stage are irrelevant to Health Sentinel as they do not carry any health events-related information. A keyword-spotting mechanism fails to filter out irrelevant ones as it cannot take the article's context into account. For instance, "What is Dengue? 10 ways to stay safe this monsoon" is an article related to human health but doesn't contain any actionable event information. Therefore, we train a binary classifier to discard irrelevant articles and effectively reduce the throughput for stages downstream, particularly translation and event extraction. To develop this classifier, we finetuned multiple Transformer-based, encoder-only (BERT-like) models that had been pretrained for language modeling tasks. For English, we experimented with six different models. For the Indic languages, we tried out four models selected based on their general performance on these languages. List of all the
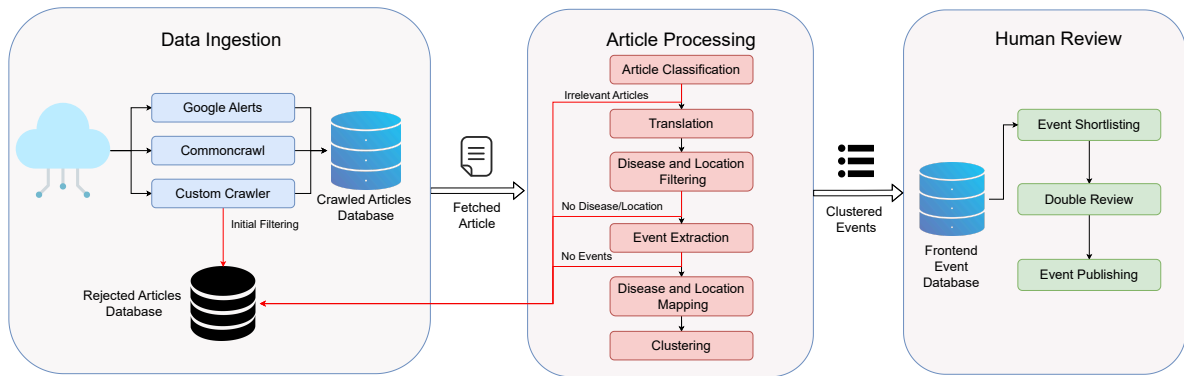

Figure 2: System Overview of Health Sentinel. Health Sentinel combines rule based and ML techniques alongside a human-in-the-loop system to ensure a high level of reliance and efficiency. Its data ingestion pipeline continuously collects news articles from the web and stores them in a database. The article processing pipeline retrieves these articles, filters out irrelevant data, and extracts health events. The extracted events are then sent for expert review before publication for ground-level action.

models we tried out is given in Section A.1 in the Appendix. The best-performing model for each of the 13 supported languages was selected based on the models' validation set performance.

3.3 Translation

Once relevant articles are identified, we translate them into English. This is necessary because most of the ML models including LLMs used in the subsequent stages of the pipeline perform better in English compared to low-resource Indian languages (Li et al., 2024). While paid APIs such as Google Translate[5] and Microsoft Azure Translate[6] have long been preferred for low-resource languages, recent open-source models including IndicTrans (Ramesh et al., 2022) and IndicTrans2 (Gala et al., 2023) perform on par or even better than these APIs for many Indic languages (Gala et al., 2023). Due to the superior performance on most of the Indic languages translation benchmarks, we use IndicTrans2 in our pipeline for translating articles into English.

3.4 Disease and Location based Filtering

Though our article classifier significantly reduces the number of irrelevant articles, some still pass through. Moreover, despite domain-block-listing, a large share of articles collected from the web discuss health events outside India. To address this, we implement additional filters to ensure that each article mentions both a disease related to humans, animals, or plants and an Indian location.

To identify diseases, we use an ensemble approach combining keyword-spotting with a disease NER model. The keywords include scientific names and common synonyms used in the media for diseases relevant to our system, curated by experts. For the disease NER, we use the open-source BioBERT[7]. For location identification, we construct an exhaustive list of Indian locations, including names of states, districts, sub-districts, and their synonyms used in the media. This list is provided to an NER model[8], which identifies the locations mentioned in the article. Any article that does not mention both a relevant disease and an Indian location is discarded.

3.5 Event Extraction

While developing Health Sentinel, we explored two approaches for event extraction. The first approach uses a combination of QA and NLI, while the second one uses LLMs.

3.5.1 Event Extraction using QA and NLI

In this approach, we use previously extracted location and disease data (see Section 3.4) for retrieving remaining entities—Incident, Incident type and Number.

Numbered Events Extraction: News articles reporting health events often include numerical information about cases or deaths (see the example in Figure 1). We use a QA model to extract such numbers by asking structured questions such as: *"How many new cases of [Disease] are there in*

---

[5]https://cloud.google.com/translate
[6]azure translator

[7]biobert disease ner
[8]chatbot ner

[Location]?" or *"How many total deaths due to [Disease] were reported in [Location]?".* If the model provides an answer, the entities used in the question, along with the extracted numerical value form an event. To ensure comprehensive extraction, we have carefully designed a diverse set of questions to account for variations in how the information may be presented in articles (see Table 5 in the Appendix). These templates cover different combinations of Incident (cases vs. deaths) and Incident type (new vs. total), while disease and location are dynamically inserted. For this task, we use `deepset-roberta-large-squad2`[9], an off-the-shelf extractive QA model.

Numberless Events Extraction: Some articles discuss important health information without providing numerical data. For example, a statement such as "Dengue is on the rise in Karnataka" highlights a significant health concern but lacks explicit numbers, while "Monkeypox: No need to be afraid, says Kerala Health Minister" contains disease information but no actionable event. The absence of numerical information makes it challenging to differentiate between actionable health events and general health information. To handle such cases, we use NLI. Hypotheses such as *"Cases of [Disease] have risen in [Location]"* or *"People are dying of [Disease] in [Location]"* are generated, and the article text is provided as the premise to the NLI model. If the model determines that the premise entails the hypothesis, the corresponding combination of disease, location, and incident is considered as an event. For this task, we use off-the-shelf NLI model `microsoft-deberta-large-nli`[10].

3.5.2 Event Extraction Using LLMs

To implement this approach, we designed a system prompt P that assigns the task of event extraction to the LLM. The prompt includes descriptions of each entity that constitutes an event and guides the model's response generation through few-shot examples $\{E\}$.

Formally, the LLM takes an article A as input and generates a structured JSON response:

$$\mathcal{E} = \text{LLM}(A, P, \{E\}) = \{e_1, e_2, \ldots, e_n\}$$

where each extracted event e_i is a dictionary containing the set entities—Disease, Location, Incident, Incident type and Number—that forms the

event.

$$e_i = \{(k_1, v_1), (k_2, v_2), \ldots, (k_m, v_m)\}$$

where k_j represents an entity and v_j is its corresponding value extracted from A. We also leverage LLMs' capability to filter out irrelevant content that may have bypassed earlier filtering at the article classification stage (see Section 3.2). The prompt explicitly distinguishes between general health information and actionable health events, instructing the model to focus solely on the latter. An example prompt is shown in Table 7 in the Appendix. Articles with no events extracted by the LLM are re-processed using another prompt, serving as a double-check for the LLM's extraction. This prompt focuses on identifying events without numerical information, similar to the NLI approach described in the previous method.

We experimented with various prompt designs and selected the most effective one based on both quantitative and qualitative evaluations. We have experimented with both proprietary LLMs and open-source ones. Table 2 can be referred to for the list of LLMs we have tried out for the event extraction.

3.6 Mapping of Disease and Location

This stage ensures that the extracted disease and location names align with standardized disease and location names used by the Media Scanning and Verification Cell. For disease mapping, we first use a curated dictionary that maps common synonyms and media terms to standardized disease names. If an extracted disease does not get mapped this way, we use an LLM to map it to the nearest standard name. For location mapping, we employ a hierarchical dictionary to assign extracted locations to administrative levels such as states, districts, and sub-districts. For any extracted location that fail to get mapped using the above approach, we prompt an LLM to map the location to an Indian state (see SectionA.3 for more details).

3.7 Clustering

A health event is often reported by multiple media outlets and other online sources. Since the previous stages in our pipeline do not check if an extracted event is a duplicate of another, we use a clustering mechanism at the end to find clusters of unique events. This stage uses a combination of ML techniques and rules to isolate unique health events. Articles are only clustered at a day-level to main-

---

[9]roberta-large-squad2
[10]deberta-large-mnli

tain consistency and ease of use. The clustering involves the following steps.

1. A pretrained sentence transformer, `paraphrase-distilroberta-base-v2`[11]., is used to create an embedding of the article associated with an event.

2. Cosine similarity is calculated for every pair of article embeddings to generate a 2D similarity matrix for each pair of events.

3. A rule set is used to analyze the extracted event information for every pair of events to determine the threshold to apply on the similarity score. Using the threshold, each similarity score is set 0 or 1. This creates a 2D match matrix with 1's and 0's.

4. A Depth First Search (DFS) is performed on the match matrix to get all the disjoint graphs. Each disjoint graph is treated as a cluster.

5. We run a conflict check on each cluster and further break it down if it has any events with conflicting information.

Further details on the clustering are given in Section A.4 in the Appendix.

3.8 Human-In-the-Loop

Before any action is taken on the extracted health events, public health experts at NCDC review them using on-ground epidemiological indicators.

4 Experiments and Results

4.1 Datasets

4.1.1 Article Classifier Dataset

We collected $34,527$ English articles sourced from the internet and manually labeled them as relevant or irrelevant. This dataset consists of $7,374$ articles in the positive class and $27,153$ articles in the negative class. This dataset was further split into training, validation, and test sets while ensuring that the test set contains a representative range of diseases to validate the classifier's performance across different scenarios. In order to train the classifier for other languages, the English dataset is translated into other 12 Indic languages using the `IndicTrans2` model.

4.1.2 End-to-End Evaluation Dataset

Articles in this dataset were sourced from news articles captured by a human-based media disease surveillance system for a period before Health Sentinel's deployment. Out of 1005 articles in the

dataset, 610 contain events (relevant articles), and 395 have no events (irrelevant articles). The dataset contains 71 unique diseases across more than 250 unique locations in India.

4.1.3 Clustering Evaluation Dataset

The dataset has 869 events spread across 7 different dates with 503 clusters that are clustered on a per-day basis as shown in Table 4. The ideal cluster compositions were annotated by health experts.

4.2 Results

4.2.1 Article Classifier

After experimenting with multiple BERT-like models, we selected the best model for each language based on recall. The results are presented in Table 1. For English, the `roberta-base` (Liu et al., 2019a) model performed the best. For other languages, `google/muril-base-cased` and `xlm-roberta-base` yielded the best results. We observe that these classifiers isolate non-health related articles as irrelevant with near perfect accuracy. However, they tend to struggle with health-related articles that do not contain any health events. For instance, "Exclusive: Monsoon can host a buffet of illnesses. Doctor reveals secrets to guarding against seasonal infections, allergies" was considered relevant by the article classifier. Overall, all selected models achieve a recall and F1-score of approximately 96%, making them highly effective as an initial filter for irrelevant articles.

4.2.2 Event Extraction

Results of event extraction are shown in Table 2. We report event-level precision, recall, and F1-score to evaluate overall performance. We also evaluate location and disease extraction separately to highlight entity-specific performance. Additionally, exact match accuracy measures how closely the extracted events resemble the ground truth, while detection rate reflects the model's ability to extract at least one event in relevant articles.

As shown in Table 2, LLMs surpass traditional NER methods in extracting disease and location information. They effectively filter out irrelevant articles, such as those related to injuries, accidents, and general health information. The qualitative results are shown in Table 3. Among the tested LLMs, proprietary models outperform open-source ones. The GPT-4o-Mini model achieves the best overall results. However, Llama3.1-8b and Gemma2-9b

---

[11]paraphrase-distilroberta-base-v2

Language	Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
English	roberta-base	0.99	0.99	0.96	0.97	0.98
Hindi	google/muril-base-cased	0.98	0.96	0.97	0.96	0.98
Telugu	xlm-roberta-base	0.98	0.97	0.95	0.96	0.97
Kannada	google/muril-base-cased	0.98	0.98	0.96	0.97	0.98
Gujarati	google/muril-base-cased	0.98	0.96	0.96	0.96	0.98
Tamil	google/muril-base-cased	0.99	0.97	0.96	0.97	0.98
Punjabi	xlm-roberta-base	0.98	0.96	0.95	0.96	0.97
Bengali	xlm-roberta-base	0.98	0.97	0.95	0.96	0.97
Marathi	xlm-roberta-base	0.98	0.97	0.96	0.96	0.97
Malayalam	google/muril-base-cased	0.98	0.95	0.95	0.95	0.97
Oriya	xlm-roberta-base	0.98	0.96	0.94	0.95	0.97
Assamese	google/muril-base-cased	0.98	0.95	0.95	0.95	0.97
Urdu	xlm-roberta-base	0.98	0.96	0.96	0.96	0.97

Table 1: Performance of the best classification models for each language, evaluated on the respective test sets. All models are downloaded from `https://huggingface.co/models` and finetuned on the respective training data.

Model	Event Extraction			Exact Match	Detection	Disease Extraction			Location Extraction		
	P	R	F1	Accuracy	Rate	P	R	F1	P	R	F1
QA+NLI based Pipeline	0.41	0.40	0.40	0.37	0.70	0.55	0.52	0.54	0.52	0.49	0.50
Llama3.1-8b[12]	0.50	0.50	0.50	0.43	0.95	0.77	0.79	0.78	0.68	0.70	0.69
Gemma2-9b[13]	0.54	0.50	0.52	0.45	**0.96**	**0.84**	0.78	0.81	0.77	0.71	0.74
GPT3.5-Turbo	0.62	0.61	0.61	0.54	0.95	0.81	0.79	0.80	0.78	0.76	0.77
GPT-4o-Mini	**0.70**	**0.67**	**0.68**	**0.61**	0.92	0.83	**0.80**	**0.81**	**0.81**	**0.77**	**0.79**

Table 2: Performance comparison of end-to-end event extraction using different models, showing results across multiple metrics— precision (P), recall (R), and F1-score at the event-level, and for individual entities— disease and location. LLM-based pipelines achieve significantly better results compared to the QA and NLI-based methods, with GPT-4o-Mini performing best overall.

#	Article	QA+NLI Pipeline	GPT-4o-mini Pipeline
1	Mysterious Disease In AP's Eluru Claims 1 Life, 347 Falls Ill, Samples Sent To Delhi.	{'Disease': 'Falls', 'Location': 'Eluru', 'Incident': 'death', 'Incident type': 'total', 'Number': '347'}	[{'Disease': 'Mysterious Disease', 'Location': 'Eluru', 'Incident': 'case', 'Incident type': 'new', 'Number': '347'}, {'Disease': 'Mysterious Disease', 'Location': 'Eluru', 'Incident': 'death', 'Incident type': 'new', 'Number': '1'}]
2	Corona turmoil in North Korea.. 21 people died of fever. North Korea	North Korea (North Korea) is trembling with fever.	{'Disease': 'Corona', 'Location': 'Korea', 'Incident': 'death', 'Incident type': 'new', 'Number': '21'}	[]
3	In Himachal, 535 people admitted to hospital after drinking contaminated water.	[]	{'Disease': 'Food poisoning infection', 'Location': 'Himachal', 'Incident': 'case', 'Incident type': 'new', 'Number': '535'}
4	Mancherial brothers' death: Two brothers passed away within hours.. knowing that the younger brother had died of a heart attack.. the elder brother went there and got a heart attack.	{'Disease': 'Cardiac arrest', 'Location': 'Mancherial', 'Incident': 'death', 'Incident type': '_', 'Number': '_'}	[]

Table 3: Qualitative comparison of event extraction by GPT-4o-Mini and QA+NLI pipelines. In example 1, the LLM-based pipeline identifies a disease missed by QA+NLI. In Example 2, it filters out an irrelevant international event mistakenly extracted by QA+NLI. Example 3 shows the LLM capturing an illness caused by contaminated water, which QA+NLI misses. In the final example, it excludes the article lacking an infectious disease component, unlike QA+NLI. Overall, LLM's inherent knowledge enables more accurate event extraction and contextual filtering of articles.

show competitive performance. Exact match accuracy is around 60% even for the best-performing LLM. We observe that the models struggle to extract all events when multiple events are present in an article. Additionally, LLMs sometimes misinterpret 'new cases' as 'total cases,' resulting in errors. Nevertheless, the system maintains high detection rate, ensuring that most relevant articles

Date	Data Points	Clusters	ARI	NMI	V-Measure
05/24/24	91	72	0.94	0.99	0.99
05/25/24	63	55	0.89	0.99	0.99
05/26/24	64	46	0.99	1.00	1.00
06/09/24	107	85	0.84	0.99	0.99
06/10/24	81	65	0.91	0.99	0.99
06/11/24	103	79	0.79	0.99	0.99
06/21/24	360	101	0.84	0.94	0.94
Avg.	–	–	**0.89**	**0.98**	**0.98**

Table 4: Per-day clustering dataset statistics and performance. Clusters are evaluated using Adjusted Rand Index (ARI), Normalized Mutual Information (NMI), and V-Measure.

are captured for the human review stage.

4.2.3 Clustering

To quantitatively evaluate the quality of the clusters formed, we employ three key metrics, Adjusted Rand Index(ARI) (Hubert and Arabie, 1985), Normalized Mutual Information(NMI) (Strehl and Ghosh, 2002), and V-Measure (Rosenberg and Hirschberg, 2007) (explained in Section A.4.2 in the Appendix). Clustering performance is reported in Table 4. For the evaluation metrics, higher values indicate strong agreement between the generated clusters and the ground truth.

5 Deployment and Impact

Health Sentinel was launched in April 2022 with support for English and Hindi. Over the two years, it has expanded the support to 11 additional languages. The system has undergone multiple upgrades, particularly in the event extraction module, to integrate the latest deep learning models and LLMs. To date, it has processed over 300 million articles from over 500,000 unique domains and has identified over 95,000 unusual unique health events. On a daily basis, Health Sentinel processes around 375,000 news articles and identify around 150 unusual unique health events. Since deployment, over 3,500 events have been shortlisted by the health experts at NCDC. Notably, only a small percentage of detected events were shortlisted by the human experts. This can be attributed to the following: i) duplicates that are not clustered correctly, ii) common outbreaks that occur during expected seasons are often not shortlisted by the health authority, and iii) if the disease is endemic in the location, such events are not shortlisted.

To better understand the impact of Health Sentinel,

we compare it with the human-based surveillance. We observed the following, i) number of published events saw a 150% increase compared to the previous years where only human-based surveillance existed ii) in 2024, 96% of the health events published by the surveillance system were extracted by the Health Sentinel (only 4% were found by manual scanning of the media), and iii) the number of media sources covered has grown exponentially because of automated media scanning and multilingual support.

6 Limitations and Future Work

In this section, we discuss known limitations of our system. Due to the lack of robust pre-trained models for Indic languages for event extraction and text embeddings, we need to translate non-English articles to English following the article classification stage. We have observed that named entities such as disease names and location names are sometimes mistranslated, leading to a lower performance for articles sourced from these languages. We experimented with fine tuning `IndicTrans2` (Gala et al., 2023) framework with an emphasis on correctly translating or transliterating named entities as appropriate. While this improved translation of the entities, it led to a decline in overall performance of the model.

Another known limitation is the lack of full context for the event extraction step, since we read only title and description of an article. Reading the full body of online articles is practically challenging since the body text is almost always clubbed with advertisements and other unrelated content and the format varies from page to page. Currently we are developing custom HTML source parsers for selected websites so that full body of the articles from these websites can be read.

7 Conclusion

In this paper, we presented Health Sentinel for automating media-based disease surveillance in a multilingual setting. Health Sentinel works by leveraging the capabilities of different sequentially connected Machine Learning models to maintain an optimal level of latency while maximizing the ability to identify unusual health events. Health Sentinel has demonstrated promising results across multiple evaluation metrics and has greatly increased the capability of disease surveillance in India.

8 Ethical and Societal Implications

Relying on online content for disease surveillance presents several challenges that can impact the reliability of the information. News sources may introduce risks of misinformation or sensationalism in reporting. Moreover, the system might inherit biases in regional coverage or language representation, potentially leading to uneven event detection.

To address these challenges, we conducted a detailed assessment of online news sources across different regions and languages in the country and implemented the following measures:

1. Keyword expansion for Indic languages. We curated a list of 7,000 disease-related keywords for Google Alerts by translating and transliterating disease names into Indic languages. This ensures early detection from regional news sources, even before these events appear on national news websites.

2. Local news coverage. We identified regional news websites that are often overlooked by platforms like Common Crawl or Google News. To address this, we developed a custom crawler that manually collects articles from these sources, improving regional representation.

3. Source filtering and periodic reviews. To handle the issue of misinformation, we maintain a list of unreliable news sources. News sources flagged as unreliable are periodically evaluated and blacklisted from entering the system in the future. Additionally, the clustering feature in our pipeline helps group similar news articles about the same event. This allows human reviewers to cross-check information from multiple sources, identify fake news and mitigate the impact of exaggerated or inaccurate information.

Despite these safeguards, some misinformation may still slip through the system. Additionally, using LLMs for event extraction can introduce noise due to hallucinations. Therefore, before the extracted information is published for field use, it must undergo a review by the health experts.

References

Auss Abbood, Alexander Ullrich, Rüdiger Busche, and Stéphane Ghozzi. 2020. Eventepi—a natural language processing framework for event-based surveillance. *PLoS computational biology*, 16.

Melvin Cabatuan and Michael Manguerra. 2020. Machine learning for disease surveillance or outbreak monitoring: A review. In *IEEE HNICEM*.

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.

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.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Jun Gao, Huan Zhao, Changlong Yu, and Ruifeng Xu. 2023. Exploring the feasibility of chatgpt for event extraction. *arXiv preprint arXiv:2303.03836*.

Karlyn Harrod, Prabin Bhandari, and Antonios Anastasopoulos. 2024. From text to maps: Llm-driven extraction and geotagging of epidemiological data. In *Workshop on NLP for Positive Impact*.

L. Hubert and P. Arabie. 1985. Comparing partitions. *Journal of Classification*, 2(1):193–218.

Andrew G Huff, Nathan Breit, Toph Allen, Karissa Whiting, and Christopher Kiley. 2016. Evaluation and verification of the global rapid identification of threats system for infectious diseases in textual data sources. *Interdisciplinary perspectives on infectious diseases*, (1).

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, Gokul N.C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-trained Multilingual Language Models for Indian Languages. In *Findings of EMNLP*.

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.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. ALBERT: A lite BERT for self-supervised learning of language representations. *CoRR*, abs/1909.11942.

Zihao Li, Yucheng Shi, Zirui Liu, Fan Yang, Ali Payani, Ninghao Liu, and Mengnan Du. 2024. Quantifying multilingual performance of large language models across languages. *Preprint*, arXiv:2404.11553.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019a. Roberta: A robustly optimized bert pretraining approach. Cite arxiv:1907.11692.

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

Marco Lui and Timothy Baldwin. 2012. langid.py: an off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, page 25–30, USA. Association for Computational Linguistics.

Office of the Registrar general and census commissioner of India. Census 2011. https://censusindia.gov.in/2011-Common/CensusData2011.Html. Accessed on 26 March 2025.

Patrick Pilipiec, Isak Samsten, and András Bota. 2023. Surveillance of communicable diseases using social media: A systematic review. *PLoS One*, 18(2).

Maciej P Polak and Dane Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications*, 15.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2022. Samanantar: The Largest Publicly Available Parallel Corpora Collection for 11 Indic Languages. *Transactions of the Association for Computational Linguistics*, 10:145–162.

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.

Marina Romanello, Claudia Di Napoli, Carole Green, Harry Kennard, Pete Lampard, Daniel Scamman, Maria Walawender, Zakari Ali, Nadia Ameli, Sonja Ayeb-Karlsson, and 1 others. 2023. The 2023 report of the lancet countdown on health and climate change: the imperative for a health-centred response in a world facing irreversible harms. *The Lancet*, 402.

Marina Romanello, Maria Walawender, Shih-Che Hsu, Annalyse Moskeland, Yasna Palmeiro-Silva, Daniel Scamman, Zakari Ali, Nadia Ameli, Denitsa Angelova, Sonja Ayeb-Karlsson, and 1 others. 2024. The 2024 report of the lancet countdown on health and climate change: facing record-breaking threats from delayed action. *The Lancet*, 404.

Andrew Rosenberg and Julia Hirschberg. 2007. V-measure: A conditional entropy-based external cluster evaluation measure. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 410–420.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Étienne Simon, Helene Olsen, Huiling You, Samia Touileb, Lilja Øvrelid, and Erik Velldal. 2024. Generative approaches to event extraction: Survey and outlook. In *Workshop on the Future of Event Detection (FuturED)*.

Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles – a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3:583–617.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Sarah Valentin. 2020. *Extraction and combination of epidemiological information from informal sources for animal infectious diseases surveillance*. Ph.D. thesis, Université Montpellier.

Sarah Valentin, Elena Arsevska, Julien Rabatel, Sylvain Falala, Alizé Mercier, Renaud Lancelot, and Mathieu Roche. 2021. Padi-web 3.0: A new framework for extracting and disseminating fine-grained information from the news for animal disease surveillance. *One Health*, 13.

Xiang Wei, Xingyu Cui, Ning Cheng, Xiaobin Wang, Xin Zhang, Shen Huang, Pengjun Xie, Jinan Xu, Yufeng Chen, Meishan Zhang, Yong Jiang, and Wenjuan Han. 2024. Chatie: Zero-shot information extraction via chatting with chatgpt. *Preprint*, arXiv:2302.10205.

WHO and 1 others. 2008. A guide to establishing event-based surveillance. https://iris.who.int/bitstream/handle/10665/207737/9789290613213_eng.pdf. Accessed on 15 August 2024.

32

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.

Daniel Zeng, Zhidong Cao, and Daniel B Neill. 2021. Artificial intelligence–enabled public health surveillance—from local detection to global epidemic monitoring and control. In *Artificial intelligence in medicine*, pages 437–453. Elsevier.

A Technical Appendix

A.1 Building the Article Classifier

A substantial portion of the articles (around 87%) that pass the domain and the language filter are irrelevant to Health Sentinel. These articles can be from different genres of news articles including entertainment, crime, accidents, politics, finance—none of which are related to health and must be discarded. Additionally, health news articles can be further divided into two types, Health Event articles and Health Information articles.

- **Health Event:** These articles contain news regarding disease outbreaks, spread, and updates within specific geographical area. They are considered relevant to Health Sentinel as they contain actionable health events. Example: "Two die of dengue in Mizoram, 1 in Manipur. Meanwhile forty-six cases of Chikungunya have been detected so far in Assam taking the total number of infections to 70"

- **Health Information:** These articles contain general information regarding different diseases, prevention methods and treatment options. They are considered irrelevant to Health Sentinel as they do not contain any actionable events. Example: "What is Dengue? How does dengue spread and 10 ways to stay safe this monsoon"

Given that this stage of the pipeline encounters a massive amount of articles that belong to 13 different languages(English + 12 Indic languages), we developed separate binary classifiers for each supported language. We intentionally avoided translating all articles to English before classification, as this would be is a time-consuming process that compromises our ability to maintain low latency. Considering that over 87% of articles are anyway irrelevant to our system at this stage, this step did not have any payoff.

To train these classifiers, we collected 34,527 English articles from different genres to ensure broad representation. The positive class comprised of 7,374 articles covering all diseases that were considered as important and additional relevant diseases. Special care was taken to include health information articles about the same diseases in the negative class, which contained 27,153 articles. Additionally, we collected a wide range of non-health related news genres as part of the negative class

to ensure classification robustness. These articles were all then translated to the 12 different supported Indic languages using `IndicTrans2` model to create 13 separate datasets for fine tuning each model.

For selecting the best model, we selected 6 different pretrained models for each of the 13 supported languages as follows:

- **English:**
  `roberta-base` (Liu et al., 2019b), `distilbert-base-cased` (Sanh et al., 2019), `albert-base-v2` (Lan et al., 2019), `xlnet-base-cased` (Yang et al., 2019), `bert-base-cased` (Devlin et al., 2018), `bert-base-uncased` (Devlin et al., 2018).

- **12 Indic Languages:**
  `xlm-roberta-base` (Conneau et al., 2019), `google/muril-base-cased` (Khanuja et al., 2021), `ai4bharat/indic-bert` (Kakwani et al., 2020), `bert-base-multilingual-cased` (Devlin et al., 2018).

All models were finetuned separately for each of the 13 supported languages and the best performing models were selected based on recall. Recall was chosen as the primary metric as the article classifier functions as a soft filter to remove majority of the irrelevant articles. Allowing some of the irrelevant articles to pass through is not a major concern, but it is essential to maximize the number of relevant articles retained at this stage. The best performing models and their metric scores are shown in Table 1.

A.2 Event Extraction

In this section we provide the details on questions and hypothesis formulation for QA and NLI based pipeline, as well as prompt design for LLM based pipeline.

A.2.1 Question formulation for the QA model

The QA model is used to extract numbered events from the articles. We use the templates shown in Table 5 to formulate questions based on previously extracted entities: disease and location. The questions contain the combination of entities: Incident and Incident type, while the remaining entity Number is extracted by the QA model. If the model returns similar values of number across different question categories, the one with the highest confidence is considered.

Category	Questions
new_cases	• How many new DISEASE cases were reported in LOCATION? • How many new DISEASE cases were reported in LOCATION in the last 24 hours? • How many fresh DISEASE cases were reported in LOCATION? • How many fresh DISEASE cases were reported in LOCATION in the last 24 hours? • How many new DISEASE infections were reported in LOCATION? • How many fresh DISEASE infections were reported in LOCATION? • How many DISEASE cases were reported in LOCATION in 24 hours?
new_deaths	• How many new DISEASE deaths were reported in LOCATION? • How many new DISEASE deaths were reported in LOCATION in the last 24 hours? • How many fresh DISEASE deaths were reported in LOCATION? • How many fresh DISEASE deaths were reported in LOCATION in the last 24 hours? • How many new deaths due to DISEASE were reported in LOCATION? • How many DISEASE deaths were reported in LOCATION in 24 hours?
total_cases	• How many total DISEASE cases were reported in LOCATION? • What is the total number of DISEASE cases reported in LOCATION? • How many total cases of DISEASE were reported in LOCATION? • What is the total tally of DISEASE cases reported in LOCATION?
total_deaths	• How many total DISEASE deaths were reported in LOCATION? • How many total deaths due to DISEASE were reported in LOCATION? • What is the total number of deaths due to DISEASE in LOCATION? • How many total deaths of DISEASE were reported in LOCATION? • What is the total tally of DISEASE deaths in LOCATION?

Table 5: Question templates for different combinations of the entities 'Incident type' and 'Incident'. The disease and location values extracted in earlier stages of the pipeline are inserted into these templates to generate specific questions.

A.2.2 Hypothesis formulation for the NLI model

Articles where the QA model does not extract any events, are processed by the NLI model for num-berless event extraction. In a manner similar to the questions, we construct hypotheses showcasing the presence of relevant health events, which are then validated by the NLI model with the article serving as premise. The hypothesis templates are shown

in Table 6. If a hypothesis is validated with an entailment score greater than 0.5, the corresponding event is generated.

A.2.3 Prompt Designing for LLM

For the task of event extraction from articles, precise and well-structured prompts are essential for obtaining reliable outputs from LLMs. In our experiments, we tested various prompts to identify the most effective one. One such prompt is shown in Table 7. Following the guidelines on prompt engineering by OpenAI[14], the LLM is assigned the persona of an Event Extractor, to align its output with the specific requirements of our task.

The prompt is carefully designed to focus on extracting infectious disease outbreaks from Indian news articles. It provides clear definition of each entity present in the event list and explicitly differentiates between relevant health events and general health information. Additionally, the LLM is instructed to exclude events related to international locations.

To guide the extraction process, the prompt is supplemented with few-shot examples. These examples include both relevant articles with actionable events and irrelevant general health information, helping LLM to distinguish between relevant and irrelevant content, enhancing the overall accuracy of event extraction. To showcase the effectiveness of few-shot approach, we compare it with zero-shot approach in Table 8. Few-shot examples lead to a significant improvement in both precision and recall.

A.3 Mapping of Disease and Location

A.3.1 Disease Mapping

In the extracted events, diseases are often present in colloquial or media-specific terms rather than the standardized nomenclature used by health authorities. For example, disease 'Pneumonia' is sometimes referred to as 'Lung Fever'. Additionally, disease extracted using LLMs may include extra words with the disease name, such as 'Cholera Infection' or 'Cholera Infectious Disease' instead of just 'Cholera'. Since our solution is tailored for use by health authorities, it's important to map these terms to the standardized disease names. The mapping process is performed in two stages:

1. **Synonym Mapping**: To standardize the name

---
14 https://platform.openai.com/docs/guides/prompt-engineering

of extracted diseases, we use a dictionary that contains a mapping of common disease synonyms and media terms to standard names. This dictionary is curated and verified by public health experts.

2. **Synonym Expansion Using LLM:** To make the synonym dictionary comprehensive, we prompt an LLM to map the un-mapped diseases from the previous stage to the nearest name in the list of 122 standard diseases within reason. Any new synonym identified is added to our dictionary, following verification by experts. If an appropriate mapping is not present from the 122 standard diseases, the disease is mapped to a miscellaneous category called "Others". The prompt used for this mapping can be found in Table 9.

A.3.2 Location Mapping

The extracted location data may include information such as the names of villages, districts, and states. However, it is essential to map these to the appropriate administrative levels, such as State, District, or Sub-district, so that the relevant health authorities can be prompted to take action. Similar to disease mapping, this process is performed in two parts:

1. **Logic-Based Mapping:** We use a standard hierarchical dictionary of States → Districts → Sub-districts → Urban Local Bodies (ULBs) and their synonyms to assign appropriate values to the extracted location data. The logic first assigns each individual location to a state, then to a district, and finally to a sub-district or ULB. Additionally, backward mapping— such as from district to state or sub-district to district is performed to handle cases where where direct mapping is not possible. A visual representation of this mapping process is show in Figure 3.

2. **LLM-Based Mapping:** For locations that cannot be mapped using the logic-based approach, we use an LLM to assist in identifying the correct administrative levels. The LLM is tasked with extracting the Indian state and district from the given article, identifying any international locations, or returning an empty result if the location cannot be mapped. Due to the extensive knowledge embedded within LLMs, they often perform accurate state-level

Category	Hypotheses
cases	• DISEASE is spreading in LOCATION
	• DISEASE was spreading in LOCATION
	• DISEASE has been spreading in LOCATION
	• Cases of DISEASE increased in LOCATION
	• Cases of DISEASE are increasing in LOCATION
	• Cases of DISEASE have risen in LOCATION
	• Cases of DISEASE are rising in LOCATION
	• A person is infected by DISEASE in LOCATION
	• A person was infected by DISEASE in LOCATION
	• A person was diagnosed with DISEASE in LOCATION
	• A person was affected by DISEASE in LOCATION
	• People are infected by DISEASE in LOCATION
	• People were infected by DISEASE in LOCATION
	• People are suffering from DISEASE in LOCATION
	• People are sick with DISEASE in LOCATION
	• A DISEASE outbreak was reported in LOCATION
deaths	• People died due to DISEASE in LOCATION
	• Deaths were reported in LOCATION due to DISEASE
	• Deaths are reported in LOCATION due to DISEASE
	• People are dying of DISEASE in LOCATION
	• Deaths have been reported in LOCATION due to DISEASE
	• Deaths have occurred due to DISEASE in LOCATION

Table 6: Hypothesis templates for different event categories ('cases' and 'deaths'). The table presents various templates used to generate hypotheses based on the extracted values of disease and location. Given an article as the premise, these hypotheses are validated by the NLI model. If the hypothesis is entailed, the corresponding event is generated.

mapping. However, it is prone to hallucinations in case of district mapping. To mitigate this, we prompt the LLM to perform the mapping multiple times and only accept it if all the outputs are consistent. The prompt used can be seen in Table 10. If an appropriate mapping cannot be found, the corresponding fields are left blank (" ").

A.4 Clustering

Once all events are extracted and processed, they are clustered together to perform de-duplication. This enables us to deal with multiple media outlets covering the same occurrence of an event and isolating all the unique health events. We achieve this by combining an ML based approach with a finely curated set of rules. Clustering is performed day-wise to categorize unique events based on their occurrence date.

System Prompt
You are a renowned event extractor specializing in identifying disease outbreaks within Indian news articles. Your expertise lies in meticulously pinpointing health events with high accuracy. Your task is to analyze an English article as input, carefully extract health events, and provide them in a structured format. A health event is an unusual occurrence in a specific area that could potentially threaten the health of people. This includes: **Unusual sickness in people:** This could be one case of a rare disease, a sudden increase in cases of a common illness, or people falling ill due to some unidentified reason. **Animals getting sick:** If there's a sudden jump in animals getting a particular disease, especially one that can spread to people, it's a health event. **Animal Bites:** Incidents of animals biting humans in a specific area. These health events serve as signals for public health officials to promptly investigate and take necessary actions to safeguard public health. Note that it is critical to differentiate between health events and health information. Health information consists of a broad spectrum of info related to human health and well-being. This includes: disease prevention, medical research, public health initiatives, guidelines, and action plans developed to combat infectious diseases. The extracted event should be presented in the following JSON schema: [{'Disease': 'The name of the disease mentioned in the article', 'Location': 'The most local geographical level affected (e.g., state, city, or district)', 'Incident (case or death)': 'Indicates whether the article discusses cases of illness or deaths', 'Incident Type (new or total)': 'Specifies if the article refers to new or total cases/deaths', 'Number': 'The numerical value associated with the incident'}] STRICTLY focus on events occurring in India, disregarding any news outside the country.

Few-Shot Examples	
Input Article	**Output**
E1: Ambikapur News: Four people of the same family fell ill after eating putu. Four members of the same family fell ill after consuming wild puttu on Sunday night in Parpatia village of Mainpat development block of Chhattisgarh.	[{'Disease': 'ill after consuming food', 'Location': 'Ambikapur', 'Incident (case or death)': 'case', 'Incident Type (new or total)': 'new', 'Number': '4'}]
E2: 8 laborers died when the truck overturned. Bihar Accident: 8 people died when a truck carrying a load of pipes overturned. Some others were seriously injured. This incident happened in Purnia, Bihar.	[]
E3: 3,353 vaccinated against rabies in government hospitals. Coimbatore: During the current year, 2,539 rabies cases and 814 cases of dog bites have been reported in Government Hospitals.	[]
E4: 906 new cases of Covid-19 were reported in India, the number of patients under treatment decreased to 10,179. India In Hindi	According to the updated data released by the Union Health Ministry at eight o'clock on Thursday morning, after the death of 20 more patients from Covid-19, the total number of people who lost their lives due to coronavirus infection in the country has increased to 5,31,814.	[{'Disease': 'Corona', 'Location': 'India', 'Incident (case or death)': 'case', 'Incident Type (new or total)': 'new', 'Number': '906'}, {'Disease': 'Corona', 'Location': 'India', 'Incident (case or death)': 'death', 'Incident Type (new or total)': 'new', 'Number': '20'}, {'Disease': 'Corona', 'Location': 'India', 'Incident (case or death)': 'death', 'Incident Type (new or total)': 'total', 'Number': '531814'}]

Table 7: System prompt with few-shot examples for extracting health events from news articles.

Model	Precision	Recall	F1	Exact Match	Detection Rate
GPT-4o-Mini (Zero-Shot)	0.60	0.58	0.59	0.52	0.91
GPT-4o-Mini (Few-Shot)	**0.70**	**0.67**	**0.68**	**0.61**	**0.92**

Table 8: Comparison of GPT-4o-Mini performance for zero-shot and few-shot prompt. The few-shot prompt include four examples from Table 7, leading to significant improvement in precision, recall, and exact match over zero-shot method.
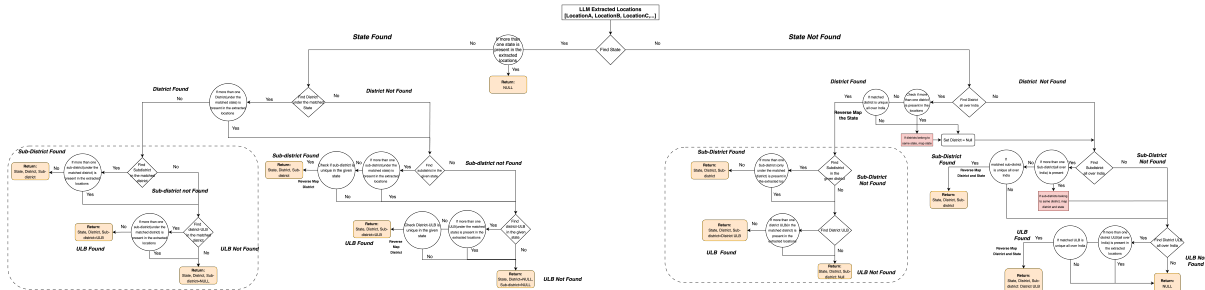

Figure 3: Logic for mapping extracted locations to appropriate State, District, Sub-district, and Urban Local Bodies (ULBs). First, individual locations are extracted from the comma separated values. The process starts with assigning a state if present, followed by assigning a district and sub-district/ULB. If a state is not identified, the logic tries to assign a district or sub-district/ULB and then reverse maps to determine the corresponding state. If multiple values are found during assignment, the location is not mapped.

System Prompt
As a renowned disease mapper, you are tasked with mapping a given disease to the nearest standard disease name in the provided list. You should only map diseases where you are certain of a close similarity, otherwise, label it as 'Others'. Your disease list is $\{DiseaseList\}$.
Few-Shot Examples
Input: "sick after eating contaminated food"   **Output:** "Food Poisoning infection"
Input: "Diarrhoea outbreak"   **Output:** "Acute Diarrhoeal Disease"
Input: "Bird flu (H5N1)"   **Output:** "Bird flu"
Input: "Cricket Fever"   **Output:** "Others"

Table 9: System prompt and few-shot examples for mapping diseases to standard names using an LLM.

A.4.1 Methodology

The following steps are undertaken to create clusters with the ideal compositions and thus identify all unique events on an daily basis:

- All the events extracted for the present day are collected.

- Every event has an associated article from which it was extracted. We use a sentence transformer, *paraphrase-distilroberta-base-v2* (Reimers and Gurevych, 2019) to generate embeddings for all the articles associated with each of the extracted events.

- We use cosine similarity to compute pairwise similarity scores for each events' associated articles. An example of how this matrix looks like can be seen in Table 11.

- Following this, a rule based approach is taken to fix the threshold that needs to be applied for each pair of events to determine if they are a match. The logical flow of the rules used for determining these thresholds are shown in Figure 4.

- After the thresholds are determined based on the rules and applied to the similarity score

39

System Prompt
You are an expert in extracting locations of occurrence of health events, with a capability of distinguishing Indian locations with international, and providing precise details down to state and district levels within India. Your task is to analyze the provided English article and identify the event's location and classify it accordingly. 1. For Indian events: Return the state and district (only if mentioned in the article). 2. If the event relates to India but cannot be pinpointed to a specific state or district, return blank values for state and district. 3. For international events: Return the output as 'International'. 4. For events discussing locations in both India and outside India: Return a blank value for the state and district.
Few-Shot Examples
Input: 'Four people of the same family fell ill after eating putu. Four members of the same family fell ill after consuming wild puttu on Sunday night in Parpatia village of Mainpat development block of Chhattisgarh.' **Output:** [{{'State': 'Chhattisgarh', 'District': 'Surguja'}}]
Input: '906 new cases of Malaria were reported from Gaya, the number of patients' deaths has reached 50. Bihar In Hindi.' **Output:** [{{'State': 'Bihar', 'District': 'Gaya, Darbhanga'}}]
Input: 'Bird flu hits Northwest Iowa dairies - Storm Lake Times Pilot.' **Output:** [{{'State': 'International', 'District': ''}}]
Input: 'Signs of bird flu in 4 states - Government of India Signs of bird flu in 4 states.' **Output:** [{{'State': '', 'District': ''}}]

Table 10: System prompt and few-shot examples for extracting state and district.

of each pair of events, we get a match matrix with 1's an 0's. An example of how this matrix looks like can be seen in Table 12.

- We then treat the match matrix as a graph problem, where a 1 represents the presence of an edge between a pair of events and a 0 represents otherwise. We use a Depth First Search approach to identify all the disjoint graphs from the match matrix and treat each of them as a cluster. For the given example in Table 12, we get two clusters as follows:

    - Cluster A: Event 1, Event 3, and Event 5.
    - Cluster B: Event 2, and Event 4.

While cluster B's formation is straightforward, it must be noted that even though Event 1 and Event 3 are not matched, they can end up in the same cluster as they are chained through Event 5.

- This chaining effect is usually observed due to the presence of events with ambiguous information. This ambiguity can occur in two different ways as follows:

    - Disease ambiguity: This phenomenon is encountered when an event's extracted disease is mapped to "Others".
    - Location ambiguity: This phenomenon is encountered when the mapped state, district, or sub district is blank("").

This sometimes leads to the formation of clusters that have events with conflicting information and are chained through an ambiguous event. Example: An event with a district as "Mallapuram" is clustered with another event with district as "Kozhikode" due to the presence of an ambiguous event with a blank district("").

- We thus have an additional step to detect clusters with conflicting information. On detection, they are further broken down into multiple clusters without any conflicting information.

A.4.2 Evaluation Metrics

To quantitatively evaluate the quality of the clusters formed, we employ three key metrics: the Adjusted

	Event 1	Event 2	Event 3	Event 4	Event 5
Event 1	1.0	0.54	0.23	0.48	0.75
Event 2	0.54	1.0	0.43	0.84	0.16
Event 3	0.23	0.43	1.0	0.38	0.89
Event 4	0.48	0.84	0.38	1.0	0.73
Event 5	0.75	0.16	0.89	0.73	1.0

Table 11: Example of a 2D matrix created by using a sentence transformer followed by cosine similarity computation for 5 events

	Event 1	Event 2	Event 3	Event 4	Event 5
Event 1	1	0	0	0	1
Event 2	0	1	0	1	0
Event 3	0	0	1	0	1
Event 4	0	1	0	1	0
Event 5	1	0	1	0	1

Table 12: Example of a 2D matrix created after applying the corresponding thresholds for each pair of events
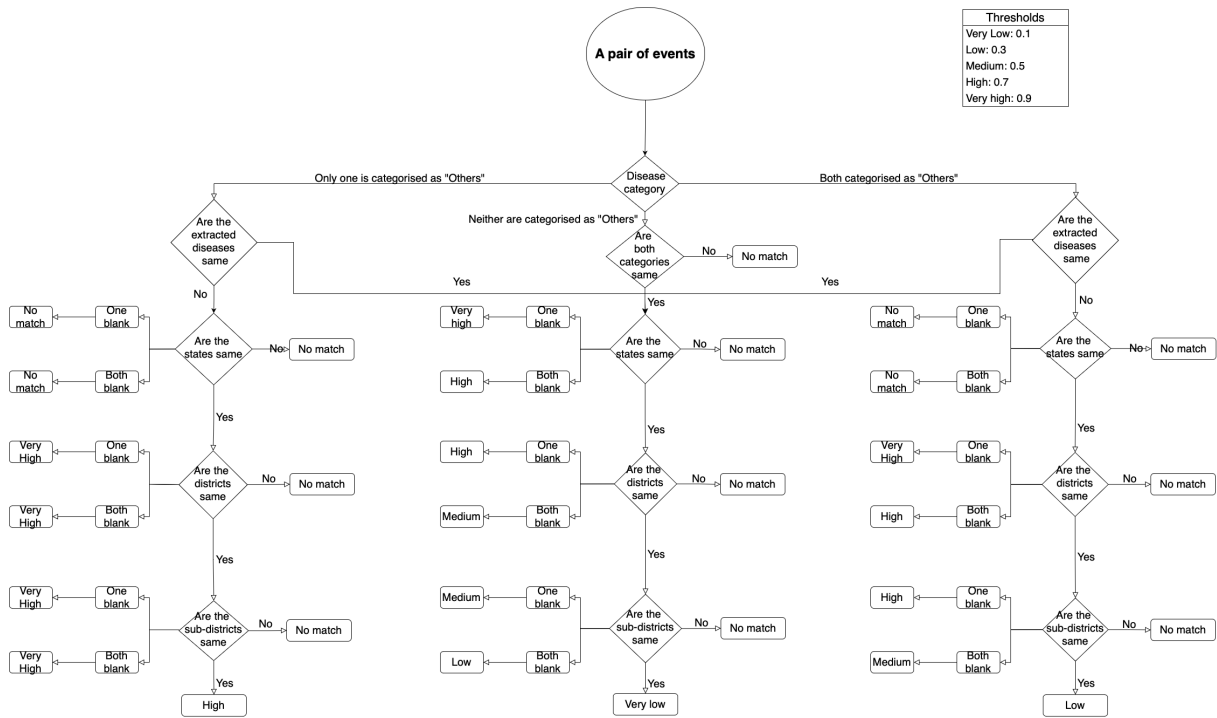

Figure 4: Logic flow of the rules that are used to determine the threshold that is applied on the similarity score for a pair of events

Rand Index (ARI) (Hubert and Arabie, 1985), Normalized Mutual Information (NMI) (Strehl and Ghosh, 2002), and V-Measure (Rosenberg and Hirschberg, 2007). These metrics are calculated on a per day basis, allowing for a detailed and dynamic assessment of clustering performance over the different dates in our clustering dataset, as illustrated in Table 1.

Adjusted Rand Index (ARI): The ARI provides a normalized measure of the similarity between two data clustering, after correcting agreements occurring by random chance. It is particularly useful in determining the agreement between the ground truth labels and the clusters generated by our algo-

rithm. The ARI is calculated as follows:

$$ARI = \frac{RI - \text{Expected}[RI]}{\text{Max}[RI] - \text{Expected}[RI]}$$

where RI (Rand Index) measures the agreement of the clustering with the true labels, defined by:

$$RI = \frac{TP + TN}{TP + FP + FN + TN}$$

Here, TP (true positives) and TN (true negatives) are pairs correctly identified as belonging to the same or different clusters. While FP (false positives) and FN (false negatives) are pairs incorrectly identified as belonging together or apart.

Normalized Mutual Information (NMI): NMI is an adjustment of the Mutual Information (MI) score that accounts for the chance grouping of elements, normalized by the entropy of the clusters. This makes it a reliable metric for comparing clustering of different sizes and compositions. It is computed as:

$$NMI = \frac{2 \times I(y; \hat{y})}{H(y) + H(\hat{y})}$$

where $I(y; \hat{y})$ represents the mutual information between the predicted and true labels, indicating the amount of information gained about one through the other. $H(y)$ and $H(\hat{y})$ are the entropy of the true labels and the predicted labels, respectively.

V-Measure: This metric offers a balance between homogeneity (each cluster contains only members of a single class) and completeness (all members of a given class are assigned to the same cluster). The V-Measure is defined as the harmonic mean of these two aspects, providing a single score to assess the effectiveness of clustering without the need for each cluster to be of approximately equal size:

$$V\text{-}Measure = \frac{\text{Homogeneity} \times \text{Completeness}}{\text{Homogeneity} + \text{Completeness}}$$

where Homogeneity and Completeness are calculated based on the distribution of each class within the clusters and the consistency of class labels within each cluster.

A.5 Hardware and Software Configuration

The pipeline presented in this paper runs on a machine with Ubuntu 20.04.6 LTS operating system, with an Intel(R) Xeon(R) CPU @ 2.00GHz and an NVIDIA T4 GPU with 16 GB of GPU RAM. The pipeline is implemented in Python 3.9. We used an OpenAI API[15] for proprietary LLMs such as GPT-3.5-Turbo and GPT-4o-mini. For open source LLMs, we used the latest instruction fine-tuned 4-bit quantized versions of models: `llama3.1-8b` and `gemma2-9b` provided by Ollama[16]. The training of certain models was performed on 32GB V100 GPUs, while all the inferences and evaluation were performed on a 16GB NVIDIA T4 GPU.

---

[15]https://openai.com/index/openai-api/
[16]https://ollama.com/