applied sciences

MDPI

Article

AI-Enhanced Disaster Management: A Modular OSINT System for Rapid Automated Reporting

Klaus Schwarz [1,2,*], Kendrick Bollens [2], Daniel Arias Aranda [1] and Michael Hartmann [2]

[1] Department of Business and Economics, University of Granada, 18071 Granada, Spain
[2] School of Technology and Architecture, SRH University of Applied Sciences Heidelberg, 12059 Berlin, Germany; kendrick.bollens@srh.de (K.B.)
* Correspondence: kschwarz@correo.ugr.es

Abstract: This paper presents the Open-Source Intelligence Disaster Event Tracker (ODET), a modular platform that provides customizable endpoints and agents for each processing step. ODET enables the implementation of AI-enhanced algorithms to respond to various complex disaster scenarios. To evaluate ODET, we conducted two case studies using unmodified AI models to demonstrate its base performance and potential applications. Through our case studies on Hurricane Harvey and the 2023 Turkey earthquake, we show how complex tasks can be quickly broken down with ODET while achieving a score of up to 89% using the AlignScore metric. ODET enables compliance with Berkeley protocol requirements by ensuring data privacy and using privacy-preserving processing methods. Our results demonstrate that ODET is a robust platform for the long-term monitoring and analysis of dynamic environments and can improve the efficiency and accuracy of situational awareness reports in disaster management.

Keywords: open-source intelligence (OSINT); artificial intelligence; disaster management; zero-shot classification; situational awareness reporting; automated intelligence extraction

Citation: Schwarz, K.; Bollens, K.; Arias Aranda, D.; Hartmann, M. AI-Enhanced Disaster Management: A Modular OSINT System for Rapid Automated Reporting. *Appl. Sci.* **2024**, *14*, 11165. https://doi.org/10.3390/app142311165

Academic Editor: Luis Javier Garcia Villalba

Received: 20 October 2024
Revised: 27 November 2024
Accepted: 27 November 2024
Published: 29 November 2024

1. Introduction

Natural and man-made disasters pose increasingly complex challenges to disaster management. With the increasing frequency and intensity of such events, rapid and accurate information gathering is critical for timely decision-making and action [1,2]. In this context, microblogging platforms such as Twitter/X have proven to be valuable sources of real-time information, often reacting to unfolding situations faster than traditional news sources by broadcasting eyewitness observations [3–6]. However, these platforms present challenges [4,5]. Large amounts of unstructured data must be processed, and relevant information must be filtered from a multitude of irrelevant or erroneous posts [7,8].

Automated systems using artificial intelligence (AI) offer a promising approach to accelerate and optimize the analysis of such data volumes. In particular, techniques such as text embedding, community-based clustering, and zero-shot classification make it possible to identify semantically relevant content and group events and present them in a structured way without the need for models that have been specifically trained for the respective event [7,8]. These techniques have the potential to form the basis for efficient and rapid reporting of disasters.

This paper presents the Open-Source Intelligence Disaster Event Tracker (ODET), a modular platform that provides customizable endpoints and agents for each processing step. ODET enables the implementation of AI-enhanced algorithms to respond to various complex disaster scenarios. To evaluate ODET, we conducted two case studies using unmodified AI models to demonstrate its base performance and potential applications. Through our case studies on Hurricane Harvey and the 2023 Turkey earthquake, we show how complex tasks can be quickly broken down with ODET while achieving

a score of up to 89% using the AlignScore metric [9]. ODET enables compliance with Berkeley protocol requirements [10] by ensuring data privacy and using privacy-preserving processing methods.

Disaster management includes prevention, preparedness, response, and recovery, which together form the disaster management cycle [11]. ODET can be used in the response and recovery phases to provide quick and accurate situational reports. In addition, ODET offers potential for the training phase by providing realistic scenarios and data-driven analysis. This not only improves the preparation and training processes but can also be used to simulate real disasters and fine-tune models.

This paper contributes to the advancement of OSINT methods in the field of disaster management by introducing ODET, a modular platform that can be used to enable faster situational awareness in disaster scenarios.

The remainder of this paper is organized as follows: In Section 2, the current state of research on open-source intelligence (OSINT) and artificial intelligence in disaster management is discussed in detail. Section 3 details the materials and methods used. Section 4 discusses the execution of case studies of Hurricane Harvey and the 2023 Turkey earthquake using ODET. Finally, in Section 5, we present and discuss the main results.

2. Literature Review

Our research builds on earlier work on open-source intelligence (OSINT) for disaster management, sentence embedding, community detection-based clustering, and zero-shot classification using large language models (LLMs).

2.1. Open-Source Intelligence in Disaster Management

Social media, particularly Twitter/X, is a valuable tool for gaining situational awareness during disasters by aggregating information from numerous sources in real time [3]. This information can be crucial for crisis responders but may not be presented in a useful format, necessitating platforms such as SMDRM (social media disaster risk management) for streamlining processing [3]. In disaster-prone countries, such as Indonesia, India, or the Philippines, Twitter/X reactions can provide insights into community responses, with sentiment analysis and classification methods aiding in understanding these reactions [12]. During specific disasters, such as earthquakes, Twitter/X discussions can offer vital knowledge for rescue planning, with multimodal analysis helping identify important disaster topics [13]. Moreover, Twitter/X data can be used to monitor disasters and predict hazards using machine-learning techniques that aid in filtering and categorizing related tweets [14].

Arapostathis et al. [15] developed a method to rapidly process and visualize Twitter/X data linked to fire events, therefore reducing the processing time to less than 2 h. Lorini et al. [3] created a platform called SMDRM (social media disaster risk management) to streamline the near real-time processing of text and images from Twitter/X during specific events. Yahya et al. [12] analyzed Twitter/X sentiments related to disaster management in Indonesia and found approximately 2081 positive and 605 negative sentiments with an accuracy of 84%. Can et al. [13] investigated Twitter/X discussions related to the 2020 Izmir earthquake by examining the spatiotemporal distribution of earthquake rescue and non-rescue terms. Baig et al. [14] explored the use of Twitter/X in disaster research by focusing on recent machine-learning, deep-learning, and disaster-prediction techniques.

In summary, this research has demonstrated the utility of Twitter/X as a real-time source of information during disasters, with applications ranging from situational awareness and community response analysis to hazard prediction and rescue planning. Studies have employed methods such as sentiment analysis, multimodal analysis, and machine-learning techniques to process and extract valuable insights from Twitter/X data, with notable success in reducing the processing time and improving the accuracy. However, there appears to be a gap in the research that focuses on the dynamic evolution of Twitter/X discussions throughout the entire disaster life cycle.

2.2. Sentence Embeddings, Community Detection-Based Clustering, and Zero-Shot Classification Using Large Language Models

Sentence embeddings capture the semantic meaning of sentences and are crucial in entity matching by transforming semantic meaning into a high-dimensional vector space [16]. Community detection-based clustering is a fast method for grouping related entities represented by their vector embeddings [16,17]. In the context of open-source investigations, sentence embeddings, and community detection-based clustering are used to extract and analyze information, with the goal of automating the process and maintaining a high alignment with reality [18]. Zero-shot classification using large language models (LLMs) has been applied across various domains to address the challenges associated with limited annotated datasets. Mugeni et al. [16] addressed entity matching in record linkage using context-aware sentence embeddings and graph clustering. Liu et al. [17] employed node embeddings and infection subgraphs for real-time rumor containment. Moura et al. [19] found that transformer-based models outperformed classical approaches for dialog data annotation, whereas Liang et al. [20] proposed an unsupervised Graph Clustering Network (GCN)-based community detection method. Kalra et al. [21] improved user clustering by integrating social context features into transformer models. In the zero-shot classification domain, Galeano et al. [22] explored LLM-based tasks, such as summarization and classification, whereas Arco et al. [23] combined LLM predictions with supervised models for better classification outcomes. Dr'apal et al. [24] developed an LLM framework for legal experts to assist with data coding and classification. Wang et al. [25] validated the performance of GPT models in text classification, and Raja et al. [26] automated article classification using LLMs.

In summary, recent research has demonstrated the utility of sentence embedding, community detection-based clustering, and LLMs in various applications, including entity matching for data quality improvement, real-time rumor containment in social networks, and open-source investigations. In particular, zero-shot text classification allows rapid adaptation to new tasks without the need for task-specific training data. Studies have explored LLM-based subtasks, aggregated predictions, and collaboration with other models while also validating and comparing their capabilities with state-of-the-art methods across a range of tasks.

2.3. Related Works

This section summarizes recent related works and evaluates them based on their methodologies, performance metrics, and datasets used. We evaluated each study based on several criteria: the effectiveness of the technology in processing large-scale social media data, the sophistication of the algorithms used for classification or clustering, and the clarity of the reported performance metrics. Additionally, the size of the dataset is considered, as larger datasets generally offer more reliable assessments of a system's performance.

Afyouni et al. [27] developed an approach for event discovery by utilizing unsupervised machine learning and Natural Language Processing (NLP) algorithms to compute events' lifetime and spatial spanning. Their incremental clustering technique employs temporal sliding windows to update the discovered topic clusters with incoming social streams (e.g., tweets). They evaluated their system on a subset of 130,096 tweets and achieved an F1 score of 0.86, with a default number of five clusters.

Karimiziarani and Moradkhani [28] utilized Natural Language Processing (NLP) techniques, including sentiment analysis, topic modeling, and text classification, to process and analyze social media data during Hurricane Ian. Although specific numerical performance scores, such as accuracy, F1 score, and precision, were not mentioned, the system was applied to 21 million tweets collected between 24 September 2022 and 6 October 2022, during the hurricane.

Ni et al. [29] developed a system using Natural Language Processing (NLP) techniques such as Bi-LSTM and Conditional Random Fields (CRF) for extracting structural information from textual emergency plans. The system incorporates scenario matching and

semantic relevance matching to generate appropriate emergency plans (EPs) from historical data, specifically for unconventional emergencies. It achieved an F1 score of 0.941 and was tested on 9891 emergency plans.

Afyouni et al. [30] proposed a system for event detection from social media. Their system integrated a semantic keyword generation tool using KeyBERT for dataset preparation. Event detection is performed by CNN and bidirectional LSTM, whereas hierarchical density-based spatial clustering is used for location-inference of events. They conducted experiments over a streamed batch of 4703 tweets, achieving an F1 score of 0.785 in their supervised approach and 0.7 in their unsupervised approach.

Huang et al. [31] developed a system integrating Natural Language Processing (NLP), the Analytic Hierarchy Process (AHP), the Entropy Weight Method (EWM), and the Grey TOPSIS method to quickly assess the relief needs of disaster areas and accurately distribute relief materials. The evaluation focused on relief demand urgency rather than specific numerical performance metrics, such as F1 or accuracy. The system was tested on 251,547 posts collected from disaster-affected areas.

Contreras et al. [32] evaluated a pre-trained sentiment analysis (SA) model developed by the no-code machine-learning platform MonkeyLearn. The model was tested on 695 tweets using hashtags related to the Albanian Earthquake posted between 26 November 2019 and 3 February 2020. The model achieved an accuracy of 63%, indicating that it is acceptable for quick estimation of text polarity during the emergency and early recovery phases after an earthquake.

Zhou et al. [33] compared several NLP models, including BERT, RoBERTa, DistilBERT, XLNet, and ALBERT, to identify which model performs best in classifying rescue request tweets. The best-performing model was BERT with a CNN classifier, achieving an F1-score of 0.919. The evaluation was conducted on 3191 manually labeled tweets from Hurricane Harvey.

Wahid et al. [34] employed Latent Dirichlet Allocation (LDA) for topic modeling and BERT embeddings for feature extraction, followed by deep-learning models (ANN, CNN, LSTM) to classify social media data for crisis response. The system achieved F1 scores between 77% and 83.1% across the different deep-learning models. The datasets included 70,000 disaster-related tweets across seven crises and 10 million COVID-19 pandemic tweets collected between 1 March and 30 April 2020.

Contreras et al. [35] performed sentiment analysis (SA) to assess post-disaster recovery on the 10th anniversary of L'Aquila's earthquake using Twitter. Polarity was first defined using a supervised classification based on experts' rules and Grammarly tones and then compared to the outcome of an unsupervised classification using the pre-trained SA machine-learning algorithm developed by MonkeyLearn. The analysis, conducted on 4349 tweets collected from 4 April to 10 April 2019, achieved an overall accuracy of 57%, with a misclassification rate of 43%. The authors suggest that these results can serve as a benchmark for comparing other post-disaster recovery processes using the same Twitter-based sentiment analysis on future anniversaries.

Bashir et al. [36] conducted a sentiment analysis of tweets manually, categorizing them into eight broader sentiment categories. Orange Data Mining Software was used in an undisclosed version to visualize positive, negative, and neutral sentiments, whereas VOSviewer in an undisclosed version was employed to visualize word frequency in the tweets. This study analyzed 13,156 English tweets posted during the first 27 days of the attack. Performance metrics, such as F1 scores, were not measured because the analysis was performed manually.

Behl et al. [37] compared supervised learning approaches for multi-class classification of Twitter data, categorizing tweets into three classes: resource need, resource availability, or others. The system achieved a macro average F1 score of 0.61 and a weighted average F1 score of 0.85. The dataset comprised 2274 tweets, with 194 classified as resource needs, 125 as resource availability, and 1955 as others. The study noted a limitation in that the classification accuracy depended heavily on the type of resource being asked for, as

demonstrated by the model's poor performance when trained on earthquake data and tested on COVID-19 data for the availability class.

Lian et al. [38] proposed a system using a density-based K-means algorithm for information extraction, ROST Emotion Analysis for sentiment recognition, MS-DICA for opinion classification, and a multi-agent system (MAS) to control false online information during natural disasters. The system achieved a reduction in false information by 2.05% and 3.02%, respectively, using different strategies. This evaluation was conducted using 42,557 samples.

Karami et al. [39] developed a framework called Twitter Situational Awareness (TwiSA) that leverages text mining methods, including sentiment analysis and topic modeling, to enhance situational awareness for disaster preparedness, response, and recovery. While the paper does not provide specific F1 scores or accuracy metrics, it demonstrates effectiveness in identifying relevant public concerns during disasters. The system was tested using 217,074 negative tweets.

Farnaghi et al. [40] introduced dynamic spatiotemporal tweet mining as a method for dynamic event extraction from geotagged tweets in large study areas. This approach employs a modified version of the ordering point clustering algorithm to address the heterogeneity of Twitter data and uses Word2Vec, GloVe, and FastText embeddings to capture both syntactic and semantic similarities. This method was used to monitor the current state of disasters through spatiotemporal and textual analyses of geotagged tweets. The system was evaluated using tweets collected during Hurricane Florence from 12 to 19 September 2018, covering North and South Carolina. The performance, measured using the silhouette coefficient, yielded 0.561 for GloVe, 0.531 for FastText, and 0.516 for Word2Vec, indicating moderate clustering quality.

Gulnerman and Karaman [41] explored the best combination of filtering, mapping, and spatial accuracy methods for social media data, particularly in emergency situations. The goal was to develop a system capable of creating an incidence map shortly after a disaster. The system was evaluated on 4395 manually labeled tweets and achieved accuracies ranging from 36% to 49% for mapping disaster-related incidents.

Fan et al. [42] proposed a hybrid machine-learning pipeline that uses Named Entity Recognition (NER) for detecting locations mentioned in posts, a location fusion approach to extract coordinates and remove noise, and a fine-tuned BERT model for classifying posts into humanitarian categories. In addition, graph-based clustering is used to identify credible situational information. The system, which aims to uncover disaster events across various locations from social media posts, was tested on 95,606 relevant tweets from Hurricane Harvey and achieved an accuracy of 75.37%.

Singh et al. [43] utilized Natural Language Processing (NLP) techniques, including topic modeling with Latent Dirichlet Allocation (LDA) and Bayesian change point detection, to analyze emotions over time in disaster-related tweets. Although the study did not provide specific numerical metrics such as F1 scores or accuracy, the system was applied to 32,909 tweets.

3. Materials and Methods

This chapter describes the materials and methods used in this study. The first section provides a detailed overview of the ODET platform, the AI models used, and the hardware and computational resources required. The following section describes the processes and procedures necessary for analyzing and processing data. The aim is to present the approach transparently and understandably to provide a basis for the implementation and results presented later. Accordingly, data preparation and anonymization are first described, followed by text moderation and filtering procedures. In the next section, the techniques used for event detection are explained, followed by a description of the zero-shot classification process. Finally, the concepts and construction of knowledge graphs and the process of retrieval-augmented generation (RAG) are explained. This chapter concludes by explaining how compliance with the Berkeley protocol was achieved.

3.1. ODET

The Open-Source Intelligence Disaster Event Tracker (ODET) is based on a modular architecture that allows for flexible adaptation to different scenarios. The system is divided into endpoints and agents. ODET supports specific endpoints, such as API endpoints for connecting external data sources, classification endpoints for filtering irrelevant or inappropriate content, text inference endpoints for semantic analysis, and embedding endpoints for creating textual representations. ODET also provides defined agent types: the API agent for data collection, the classification agent for categorization and filtering, the embedding agent for semantically grouping content, the knowledge graph agent for extracting relationships, and the summary agent for generating detailed reports. These agent types can be flexibly combined and linked in the processing pipeline to satisfy specific requirements. The modular design makes it easy to replace or extend individual components, keeping ODET future-proof and compatible with new technologies or models.

In the case studies, unmodified AI models were used to show that ODET is powerful, even without additional training steps. The GIST-all-MiniLM-L6-v2 model was used for sentence embedding to generate high-dimensional text representations and capture semantic relationships between the analyzed data. The quantized model Mistral-Nemo-Instruct-2407-Q6_K_L was used for zero-shot classification and other agents, such as the knowledge graph agent and the summary agent. This model is not a pre-trained classifier but a basic Instruct model that is dynamically adapted by ODET using GGBNF grammar and specially defined prompts to generate zero- or multi-shot classifiers and other specialized outputs. The context window size was 65,536 tokens (n_ctx = 8192 $\times$ 8) with a batch size of 512 tokens (n_batch = 512), and 10 GPU layers (n_gpu_layers = 10) were used. For hate-speech detection, the KoalaAI TextModeration model was used, which analyzes content with a batch size of 64 tokens (batch_size = 64) while considering truncations (truncation = True).

Computing requirements vary depending on the ODET deployment scenario. In the case studies, the system successfully ran on a MacBook Air with an Apple Silicon M3 processor and 24 GB of shared RAM, although this configuration was not optimal. Using the Mistral-Nemo-Instruct-2407-Q6_K_L quantized model with a context window of up to 65,536 tokens and a batch size of 512 tokens typically requires at least 10 GB of RAM for model inference alone. Additional memory is required to process the input data and to perform parallel tasks, such as creating embeddings or performing classifications, bringing the total memory requirement to approximately 16–20 GB of RAM. On the CPU side, at least 8 cores are recommended to efficiently support multi-threaded processing. Running on an A100 GPU in the Google cloud proved to be much more powerful, processing the entire pipeline at a much higher speed and with lower latency.

The ability to run ODET on both consumer hardware and cloud-based endpoints renders the system highly scalable and flexible. This scalability not only allows the system to adapt to varying resource availability but also provides the basis for rapid deployment in real-world disaster situations where time and availability are critical.

3.2. Data Preparation and Anonymization

The dataset used for the case study on Hurricane Harvey was obtained from Kaggle and includes tweets posted during the disaster between 20 and 30 August 2017. The dataset includes information such as tweet ID, number of likes, replies, retweets, timestamps, and tweet content, and was published by the original creator on Kaggle under the CC0 license [44].

For the 2023 Turkey earthquake case study, a dataset shared on Kaggle was used. According to the original author, the dataset was collected using the Tweepy software in an undisclosed version and the Twitter/X API on a daily basis during the disaster event. Tweepy is an open-source Python package that facilitates access to the Twitter/X API. The dataset consists of public tweets containing information about the earthquake, including fields such as ID, text, hashtags, source, retweets, and retweet status [45].

To protect user privacy, all identifiable information, such as author names and mentions, was removed from the tweets. This anonymization ensures that the analysis is based solely on the content of the tweets and that no personal data are revealed. The datasets contained no specific user data or account information other than mentions or usernames. Accordingly, the anonymization procedure can be limited to the removal of mentions and author names. Subsequently, the data preparation process first converts the timestamps into a uniform date format to ensure that all tweets have a standardized time reference. Care was taken to remove invalid entries, such as tweets without a timestamp or tweets without text content. The next step is to clean the data. This involves using regular expressions to remove various unwanted strings, including HTML entities, URLs, and the mentions of other users. The goal of this step is to reduce the text to core content and avoid possible disturbances during the analysis. Furthermore, the use of the upper and lower cases is standardized to ensure that the same words are not recognized as different terms if they are written differently.

3.3. Text Moderation and Filtering

In disaster management, it is crucial that the analyzed data are free of offensive or irrelevant content, as such content can negatively affect the accuracy and reliability of situation reports. For this purpose, the KoalaAI text moderation model was integrated into the preprocessing pipeline [46]. The KoalaAI classifier was chosen primarily for its speed and efficiency, as it allows us to quickly filter out unwanted content, such as hate speech, which could otherwise corrupt final reports. Although a more comprehensive text classification model can be used for filtering, the performance of the KoalaAI classifier is sufficient for the task at hand, particularly given the time-critical nature of disaster management.

Filtering out hate speech is not only a technical necessity but also an ethical imperative. The KoalaAI text classifier was configured to filter out tweets identified as hate speech. This filtering step is critical because hate speech, if included in the final reports, could undermine the credibility of situational awareness reports, potentially leading to distrust and misinformation. In the broader ODET platform, the filtering model can be swapped or even omitted, depending on the specific requirements of the case. In other studies, such as those on the conflicts in Ukraine, it was necessary to filter additional content types. This makes it possible to adapt it to complex scenarios. Even in the Hurricane Harvey case study, additional hate-speech filtering was necessary, as the moderation measures implemented by Twitter in 2017 were not sufficiently effective.

In ODET, text moderation and filtering were performed in batches to process large numbers of tweets efficiently. The Huggingface transformer framework was used to create a pipeline for text classification [47]. Each tweet was submitted to the model for classification, and a label was assigned as a result, indicating whether the tweet contained hate speech. Consequently, the hate-speech-labeled tweets were removed from the dataset. This ensured that the dataset was cleaned from irrelevant and disrespectful content for analysis.

3.4. Semantic Analysis and Clustering

The GIST-all-MiniLM-L6-v2 model [48] was integrated into the data pipeline by directly applying it to the pre-filtered and moderated data. After preprocessing, which involves filtering out hate speech, the tweets are converted into high-dimensional vector representations and stored in a vector database. These vectors store the semantic relationships between the tweets. In addition, a client for a persistent vector database was created to efficiently store the generated vectors and enable access for further steps.

ODET begins to process data in chunks to divide large amounts of data into manageable sections. These chunks represent time slots that can be preselected. This makes it possible to view data from historical records as live data by examining only ever looking at the currently relevant time slot. The preprocessed, filtered, and vectorized tweets were stored in a vector database. Each chunk is added to a separate collection, with the name of

the collection being generated dynamically (e.g., “collection0”, ”collection1”). If a collection does not yet exist, it is created by the client.

This process involves adding vectors and IDs to the collections in the database to enable efficient and rapid searches for semantically similar content. This mechanism ensures that the data are structured and prepared for the subsequent semantic analysis. Finally, the file management process (handle_list_file) ensures that the information in the collections remains consistent and available for further analysis.

After transforming the tweets into embeddings, a community detection algorithm was used to identify semantically similar groups of tweets. This is performed by calculating the cosine similarity between the vectors, which measures the degree of content similarity between tweets. Tweets with high content similarity generate a similarity value close to one, whereas those with less similarity generate lower values. To ensure that only relevant semantic relationships are captured, similarity values below a threshold of 0.1 are set to zero.

This process is performed in small steps to keep the calculations memory-efficient. The calculated similarity matrix was converted into a sparse matrix and stored in a compressed format. This data structure allows efficient processing of large amounts of data because only the relevant connections between tweets are stored.

A similarity matrix serves as the basis for the community detection algorithm, which groups semantically similar tweets and identifies and eliminates overlapping communities. Thus, significant topics in the data can be identified and used to generate situation reports. The minimum cluster size depends on the number of tweets analyzed. For smaller datasets, the minimum size can be reduced accordingly, whereas larger datasets require larger clusters.

3.5. Zero-Shot Classification and Information Extraction

For the case studies presented, zero-shot classification was realized using a 6-bit quantized version of the Mistral-Nemo-Instruct-2407 model that was specifically optimized for instructional tasks by its creators [49]. This model is a fine-tuned version of Mistral-Nemo-Base-2407, which was jointly developed by Mistral AI and NVIDIA. It is designed to ensure both high accuracy and efficiency and can be easily replaced by more advanced models in future studies. We used this model in a quantized but unmodified version. The quantized version of this model was specifically chosen for efficient use of memory and computing power. The model required approximately 10 GB of main memory with a near-perfect quality, making it particularly suitable for use in a wide range of hardware configurations.

A key aspect of ODET is its ability to freely exchange models and adapt them to specific requirements. The goal was to develop a system that does not rely on specially trained models, and can, therefore, be flexibly applied to different future disasters. This architecture allows for the exchange of AI models, embedding models, and classifiers to ensure that researchers can benefit from new developments or integrate their own models without being tied to specific technologies.

Another central component is the ability to customize the prompts and grammar used to control the model output. The flexibility of the system allows researchers to define specific output requirements realized through JSON schema-based grammar. The schema used for the case studies consisted of disaster-event-related, location-related, and headline-related components. These components ensure that the model’s output always includes information on whether a cluster is related to an event of interest, where the event is occurring, and which headline is the most appropriate. Using this structured grammar, we can ensure that the results are consistent and analyzable. The flexibility of this methodology allows us to define different questions and output formats for different use cases.

A specially developed prompt template that builds upon the grammar utilized earlier was used for the classification. As an example, the template for the Hurricane Harvey case study contains specific instructions that prompt the model to provide concise answers to questions such as, “Does the tweet refer to a hurricane?”, “Where is the event located?”

"What would be an appropriate headline?". This structured approach can be changed to specifics regarding the event of interest and ensures that the classified data are consistent and easy to analyze. The model was dynamically generated and adapted to a specific use case to provide accurate and meaningful results.

During the analysis, all clusters were classified thematically. Clusters that were classified as irrelevant were discarded to ensure the quality and meaningfulness of the final situation report. The cluster formation and processing method helps to increase the efficiency of data processing by processing large numbers of tweets in a shorter time and with fewer computational resources. The results of this process were stored in Python dictionaries and were subsequently used for further analysis. Data were filtered based on the output JSON values, excluding records classified as irrelevant.

Occasional misclassifications do not have a significant impact on the overall context of the data analysis. The clustering process in ODET consists of two distinct phases. First, the vector embeddings of the tweets are generated using a sentence embedding model that transforms the data into a high-dimensional vector space in which semantic relationships are preserved. These embeddings are then sorted and clustered using a cosine similarity threshold such that only semantically relevant tweets are part of the same cluster.

Choosing an appropriate threshold for cosine similarity is a trivial procedure performed empirically by the user. This is performed by taking a set of samples of the computed similarities between vector representations generated with the desired sentence embedding model. Based on these similarity values, a threshold is chosen to ensure that only semantically related content is grouped into a cluster.

ODET supports all sentence embedding models supported by the SentenceTransformers library. This wide selection provides flexibility for different use cases and allows the user to choose models that are best suited to a particular problem.

Even a threshold that is not optimally chosen has little impact on the final result in the ODET pipeline. This is because the threshold is only used to cluster similar tweets.

After clustering, a zero-shot classification process is performed to assign thematic labels to each cluster. This comprehensive approach ensures that every sample within a cluster is processed, therefore minimizing the effects of misclassification. Given the large amount of data and robust clustering methodology, uncertainties, including hallucinations and misclassifications, must exceed a very high threshold to have a negative impact on the final results. In addition, a knowledge graph is generated in the further course of data processing, and retrieval-augmented generation is used, which provides an additional structure and context and further increases the robustness of the results.

3.6. Knowledge Graph Construction

To extract the relationships between entities, a similar methodology was used, in which the AI model was again supported by a specially defined grammar. This grammar specifies that for each relationship between entities, three key elements are extracted:

- the 'subject'—which represents the main entity,
- the 'relation'—which describes the relationship between the entities,
- and the 'object'—which represents the target of the relationship.

Each element is defined in the schema as a string-based property. Grammar ensures that the model structures the relevant information and outputs it as triples, which is the basis for creating a knowledge graph.

These triples are the building blocks of a knowledge graph based on extracted entities and their relationships. In a knowledge graph, an entity such as "Hurricane Harvey" is linked as a subject to other entities such as "flooding" (object) via a relationship such as "caused". This triple forms the basis of the semantic representation of the data. These links make it easy to represent the complex relationships within tweets. The knowledge graph connects these relationships to create a holistic understanding of events that go beyond mere data points.

Next, the previously classified clusters were divided into smaller groups to allow more efficient processing. The tweets contained in each group were combined into a single string by continuously adding the content of the tweets. The batch size can be defined freely. As soon as a group reached the defined batch size, the block was passed to the model for analysis. This structure facilitates the processing of large amounts of data into manageable batches.

The extracted data were returned as structured JSON responses, which were then organized in a Python dictionary. Each cluster was analyzed using the model, and the resulting triples were stored in the dictionary under the appropriate index. This process makes it possible to systematically document the relationships between the extracted entities and map them in a knowledge graph.

3.7. RAG and Summarization

The retrieval-augmented generation (RAG) process is critical for generating accurate and comprehensive summaries from large amounts of text [50–52]. This process combined the extraction of relevant information with the generation of new content to create meaningful and contextually relevant summaries. ChromaDB plays a central role in storing and querying the generated vector embeddings [53]. Previously generated vector embeddings were stored in the ChromaDB database, and relevant documents were efficiently retrieved through targeted queries. These queries allowed the collection of contextually relevant information for the subsequent summary. The headings from the zero-shot classification were used as inputs for the queries. At any given time, ChromaDB contains data only from the current hour. Agents are unable to look into the past or future.

The summarization process was performed using specialized agents that summarized the retrieved information. These agents used previously created knowledge graphs and contextual data to generate concise and consolidated summaries. The summarization agents interpret the knowledge graphs for each cluster (topic) to provide detailed situational analysis. These summaries have been consolidated into comprehensive reports.

3.8. Berkeley Protocol Conformance

This section explains how our technical process conforms to the standards and principles of the Berkeley protocol for open-source digital investigations [10].

3.8.1. Anonymization and Privacy

We anonymized the tweets in our data-processing process by removing all user-related information. This includes the deletion of usernames, locations, and other identifying features. This ensured that the results could not be traced back to the authors of the tweets. Our process processes tweets mechanically, without storing or mining user information.

3.8.2. Methodological Principles of the Berkeley Protocol

The Berkeley protocol emphasizes the importance of accuracy, data minimization, and data security in open-source digital investigations. Our approach meets these criteria in several ways.

- Accuracy:
  - Using advanced NLP techniques such as sentence embedding and community detection, we ensured that the data were accurately analyzed, and relevant events were correctly identified.
  - Using ChromaDB to store and query vector data ensures that only the most relevant information is used for analysis.
- Data minimization:
  - We processed only the necessary information by filtering out irrelevant and potentially harmful content, such as hate speech, using a classification model from KoalaAI [46].

  - Our approach limits the analysis to hourly intervals, with only the currently relevant hourly data stored in ChromaDB. This minimizes the amount of data and increases the focus on the current events.
- Security:
  - By isolating data on an hourly basis, we prevent our agents from looking into the past or future, ensuring the integrity of the data and the accuracy of the analysis.
  - Anonymizing the data and using only machine processing protects user privacy and ensures that no personal data are stored or analyzed.

4. Case Studies

This section explains the implementation of the Open-Source Intelligence Disaster Event Tracker (ODET) for the case studies of Hurricane Harvey and the 2023 Turkey earthquake. This describes how ODET was applied step-by-step to the respective datasets to extract relevant information and create automated reports.

Hurricane Harvey, which hit the coastal regions of Texas and Louisiana hard in August 2017, caused significant damage and economic losses. The storm caused widespread flooding and took a heavy toll on human lives, making it one of the most devastating natural disasters in US history [54]. By comparison, the earthquake in Turkey in February 2023 presents a very different scenario: measuring 7.8 on the Richter scale, it devastated large parts of the country, causing severe building damage and thousands of deaths [55]. The two events present different but equally challenging scenarios for applying ODET, demonstrating how flexibly the system can be adapted to analyze different disaster events and efficiently provide relevant information.

This section explains the basic structure of ODET in the first step, starting with setting up the endpoints for data collection, including configuring an API endpoint for the Hurricane Harvey dataset. In the second step, various agents are configured in detail. The third step describes the structure of the data-processing pipeline and explains how to execute it to enable continuous data analysis. Finally, benchmarking of the reports is described, including establishing a basis for comparison and selection of suitable metrics to evaluate the generated content. Figure 1 illustrates each step of ODET's structure.

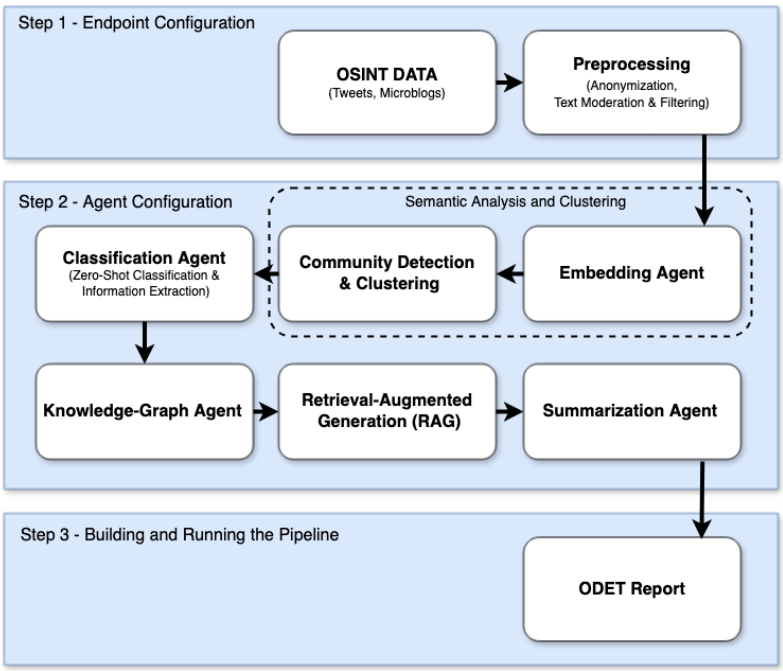

Figure 1. Structure of ODET for automated disaster event tracking and reporting.

4.1. Step 1: Endpoint Configuration

The endpoints form the basis of ODET's modular architecture and enable researchers to create their own modules and flexibly adapt ODET to new technologies or data sources. The ability to create custom endpoints makes ODET future-proof. For example, if a new powerful large language model (LLM) is released, a custom endpoint for text inference can be implemented. Similarly, the structure allows for easy integration of new datasets into the ODET. To integrate a new dataset, a REST API endpoint that provides specific data in the desired format can be made available. ODET supports four main types of endpoints:

1. API endpoints—Enables the connection of external data sources (e.g., web APIs such as Twitter/X or self-hosted datasets).
2. Classify endpoints—Responsible for classification models, for example, filtering irrelevant or inappropriate content.
3. Text Inference Endpoints—Use inference models to derive specific information from text.
4. Embedding endpoints—Creation of text embeddings to enable semantic similarities and clustering.

As shown in Figure A1 we created an API endpoint for the Hurricane Harvey dataset that makes tweets accessible during hurricanes. These endpoints allow flexible querying of data at defined time intervals and are further processed by downstream agents. Aligned with this example, the API endpoint was implemented for the 2023 Turkey Earthquake study.

4.2. Step 2: Agent Configuration

In ODET, agents are used as central building blocks to perform various data-processing tasks. Each agent performs a specific task, such as classifying data, creating text embeddings, or creating knowledge graphs. These agents operate sequentially, with each agent processing the results of the previous agent. ODET's modular design allows agents to be flexibly adapted, replaced, or extended to integrate future requirements and developments in AI technology.

4.2.1. API Agent Configuration

The API agent is the starting point of the data pipeline in ODET and is used to connect and retrieve data. As shown in Figure A2, several parameters that influence the query and timing of datasets can be specified. While parameters can be set individually, for the case studies presented, the parameters start, offset, and interval were used. The ability to define parameters freely allows researchers to work with a wide range of time-series data and datasets from microblogging platforms.

- Start Time—This determines when the analysis of the dataset begins. For the Hurricane Harvey case study, 3:00 p.m. on 25 August 2017 was chosen to synchronize the start of reporting with the initial impact of the storm.
- Interval—This interval divides the data into regular hourly slices (60 min), resulting in hourly reports.
- Offset—This parameter determines the time of day the analysis begins. In this case, the offset was set to 3:00 p.m. The first data considered by the agent was the data created after 3:00 p.m.
- Input and Output—The input and output options of the API agent indicate that it is at the beginning of the pipeline and has no previous input data.

4.2.2. Hate-Speech Filter

The configuration of the classification agent is the next step in the pipeline, as shown in Figure A3. This filter is a key part of data processing to remove inappropriate content. Enabling the hate-speech filter ensures that any inappropriate content that is not relevant to the analysis of the disaster event is removed. This helps to ensure the accuracy and reliability of the reports. This step is not mandatory, and researchers may choose not to

include it. However, for the Hurricane Harvey case study, as well as the 2023 Turkey earthquake case study, a hate-speech filter was enabled.

- Filter label—The filter is set to remove all tweets classified as "hate speech" by labeling them with an "H". ODET supports several different filters and labels through settings such as these.
- Input options—The hate-speech filter uses the API agent's output. ODET allows the data to be seamlessly transferred from the API agent using the "Map from previous" configuration option.
- Output options—The output data are passed to the next agent in the pipeline after filtering. It is decidable to either silently pass on the data or output debug information.

4.2.3. Embedding Agent

The embedding agent is a central element of the pipeline and is used to semantically group tweets and organize them for further processing. As shown in Figure A8, the agent uses vector embeddings to group tweets based on their similarity. The input data for this agent comes from a previously set up hate-speech filtering agent.

- Similarity Threshold—The cosine similarity threshold between the tweets was set to 0.84. This means that only tweets with a high degree of content similarity were grouped into clusters.
- Minimum Cluster Size—To ensure that only relevant groups were formed, a minimum of 100 tweets per cluster was set. The cluster size allows for the identification of trends. A relevant event is expected to generate numerous similar messages.
- Grouping of Cluster Elements—Large clusters can contain thousands of similar tweets. To further compress them, tweets from a cluster can be grouped. This has the advantage of minimizing the number of queries to computationally intensive AI models and increasing efficiency. This is particularly useful for processing large amounts of similar content in a cluster as a single unit because fewer queries need to be made.

4.2.4. Classification Agent

The classification agent is responsible for filtering irrelevant clusters and assigning an appropriate label to each relevant cluster. As shown in Figure A5, this agent allows the configuration of a zero-shot classifier based on an LLM. The classification agent is critical for further processing of the clusters. It uses a pre-trained language model driven by JSON-based grammar. This grammar defines the output parameters and allows the model to determine the relevance of the clusters based on prompt instructions.

- Grammar and JSON schema—Grammars define the outputs of the language model. They rely on a prompt template adopting its fields. Grammars were created as JSON schema and translated by ODET into a format that can be processed by an AI model. Both case studies used a schema that included a Boolean value for detecting disaster relevance. An example is shown in Figure A4.
- Prompt template and objective—This template provides specific instructions for creating the headline, rating the relevance, and identifying the location. The goal of the agent is to determine the relevance of the clusters by answering the question of whether the content is disaster-related. It is possible to answer multiple questions simultaneously to save steps. In the context of the case studies presented in this paper, the agent was also asked for a possible location for the described context as well as for a headline summarizing the events. The results are output in a predefined format and structured based on the defined JSON grammar. An example is shown in Figure A9.
- Mapper and object filter—The classification agent has a mapper that maps objects and a filter that removes irrelevant clusters based on the classification results. If a cluster was marked as irrelevant ("false"), it was removed from the pipeline. An example is shown in Figure A7. The mapper and object filters can be used to create complex filter rules that remove unwanted objects or process them in a targeted manner. On

the other hand, objects can be grouped or processed in a targeted manner. This was made possible by the generated JSON output.

- Input and Output—The agent receives its input data from the embedding agent and outputs filtered and classified clusters with associated headings. The data are then passed to the next agent.

4.2.5. Knowledge Graph Agent

The knowledge graph agent is a central component of the ODET pipeline that not only identifies relationships between entities but also refines them through retrieval-augmented generation (RAG). RAG makes it possible to extract relevant information directly from the original data, which improves the accuracy of the reports because the generated summaries are supported by real data. Furthermore, the knowledge graph agent makes it possible to capture the dynamics of events by representing not only static but also changing relationships. This feature helps to track the progression of a disaster event in real time and to analyze the impact of decisions or external factors on the event. Combining the extracted knowledge graphs with RAG allows ODET to create more accurate and up-to-date reports that are invaluable to decision-makers in crisis situations. The input for this agent comes from the results of the classification agent, and the output is processed in the last step of the pipeline. This agent uses the same text inference endpoint as the classification endpoint and differs only in the grammar, prompt, and RAG settings. An example of the agent configuration is shown in Figure A6.

- Grammar and JSON schema—The agent analyzes the semantic relationships between entities based on the context of tweets. ODET allows the definition of a JSON grammar that specifies the structure of the output data as triples (subject, relation, and object). These triples are supported by a prompt similar to that used by the classification agent. An example is shown in Figure A10.
- Retrieval-augmented generation (RAG)—The knowledge graph agent combines the generated graphs with RAG using the generated headlines to access the original data set and collect further information. This ensures higher accuracy and increases the validity of the reports.
- Grouping of entities—Since it can be assumed that messages with similar content and a similar creation time refer to a single event, the agent groups such messages. This reduces the number of requests for the model and optimizes processing speed. Grouping similar data points increased the efficiency of knowledge graph generation.
- Visualization of results—The knowledge graph agent creates the results as a structured network of entities in the form of triples, which supports the creation of reports and simplifies data analysis. The generated knowledge graphs can be displayed graphically in the report if desired, making it easier to identify correlations and validate relationships.

4.2.6. Summarization Agent

The summarization agent is the final step in the pipeline for Hurricane Harvey and the 2023 Turkey earthquake case studies, where the summary reports are generated based on the processed data. As shown in Figure A11, the agent can efficiently summarize relevant information using previously generated knowledge graphs and retrieval-augmented generation (RAG) data in the system. Although the summarization agent represents the final step in the case studies presented in this paper, it can be complemented by additional post-processing agents in other applications. The modularity and flexibility of ODET allow researchers to adapt the structure of their pipelines to the specific needs of their projects.

- Objective and reasoning—Researchers can provide detailed instructions to the agent and specify what information is included in the report and how it should be structured. As shown in Figure A12, the agent was instructed to summarize the content from the knowledge graph and create an appropriate headline for each report.

- Markdown output—The agent outputs the reports in the Markdown format, which allows for flexible integration with external systems and reporting formats.

4.3. Step 3: Building and Running the Pipeline

The third and final step in ODET involves building and running the report generator, which allows researchers to configure agents in a specific order and generate consecutive reports. These reports are defined in what is called the "report area," where researchers can name the report, add a detailed description, and specify the exact order of agents to be processed. The description clarifies the focus of the report to ensure that it meets the objectives of the research project.

A key aspect of report creation in ODET is the ability to select and sequence agents flexibly. This is presented in a user-friendly interface, where researchers can drag agents from the list of available agents into the selected agent box. Multiple agents can be used in a pipeline or multiple times in a report to account for different aspects of data evaluation. This provides a high degree of flexibility as researchers can use the order and selection of agents to perform various analyses tailored to the specific needs of the investigation.

Another important feature of ODET is its ability to customize the frequency of report execution. The system allows researchers to repeat reports at regular intervals, which is particularly useful when processing time-series data. The execution frequency can be set to a minute, hour, day, or month. This makes ODET useful for both analyses that run through historical data and real-time analyses, in which it is important to constantly integrate and analyze new data. This ensures that researchers have up-to-date information and can make quick decisions based on the latest data.

In addition to flexibility in execution frequency, the ODET also provides complete interactive control over the agents during reporting. Any agent that is part of the pipeline can be paused, restarted, or re-executed to ensure an optimal operation. This allows researchers to adjust and try agents until they produce the desired results. This interactive capability makes the ODET a valuable tool for iterative report development and improvement. Another useful feature is the ability to regenerate the entire report, which is especially helpful when changes have been made or when the report needs to be re-run after agent customization.

The generated reports provide detailed outputs, organized in a way similar to Jupyter notebooks. This means that each step of the pipeline is represented in a report, and researchers can review the output of each agent before proceeding to the next step. These reports can include both text-based and graphical outputs, and researchers have the option to view results in a Markdown format or other suitable formats to prepare and visually display the data for analysis.

Overall, this step in ODET enables the creation of flexible and scalable reports designed for both one-time analysis and continuous real-time monitoring of disaster events. The ability to interactively customize agents and continuously refresh reports ensures that ODET remains a powerful tool that meets the changing needs of disaster researchers. An example of the reported setup is presented in Figure A13.

5. Results and Discussion

This section presents and discusses the results of the case studies performed to evaluate ODET's performance in terms of consistency with established reference sources. In addition, we discuss implications for disaster management. The central evaluation approach is based on measuring the content consistency between reports generated by ODET and English-language Wikipedia articles on Hurricane Harvey and the 2023 Turkey earthquake using the AlignScore metric. Finally, the strengths and weaknesses of the system were analyzed to identify opportunities for improvement and future applications.

5.1. AlignScore

To evaluate the quality of the reports generated by ODET in our case studies, the metric AlignScore was used to show the validity of the results. This metric makes it possible to compare machine-generated reports with an established reference source and to measure the alignment of the content elements. AlignScore provides a systematic and objective metric for evaluating the accuracy and completeness of automatically generated content compared to a reliable source [9] by measuring its structural and semantic alignment. The advantage of this metric is that it is possible to measure both content scope and semantic accuracy, which provides a more comprehensive insight into the system's performance. AlignScore considers not only the surface area but also the depth and context of the information. AlignScore is mathematically described as:

$$\text{ALIGNSCORE}(o, l) = \underset{j}{\text{mean}} \max_{i} \text{alignment}(o'_i, l'_j), \tag{1}$$

where o refers to the context, l represents the claim, o'_i is the collection of context chunks, l'_j is the collection of claim sentences, and alignment$(\cdot)$ denotes the likelihood that the model assigns to the Aligned label in a 3-class classification setup [9].

The goal of ODET is to generate situational awareness reports from unstructured disaster data. This task requires a robust evaluation platform that can assess how well generated outputs reflect real-world events. Compared to traditional metrics such as F1 score, AlignScore is well suited for ODET evaluation because it focuses on factual consistency rather than discrete classification accuracy. AlignScore addresses this need by holistically evaluating the semantic fidelity of the generated text to the reference material to identify nuanced discrepancies such as omissions, factual distortions, or misrepresentations [9].

Throughout the case studies of the 2023 Turkey Earthquake and Hurricane Harvey, Wikipedia was a valuable source for evaluating reports. A direct timeline of events was not available; therefore, Wikipedia pages had to be relied upon as a comprehensive source of information. These pages contain well-researched and detailed neutral information that has been edited and added by numerous authors over the years. Although Wikipedia contains retrospective insights and information that only became available after the event, it provides a reliable basis for comparison.

To evaluate the case studies, each of the 58 ODET-generated reports for the Hurricane Harvey case study and 131 reports for the 2023 Turkey Earthquake case study were analyzed in a final summary report and then compared to the corresponding Wikipedia article. The high level of agreement indicates that the ODET-generated reports were largely accurate despite the limitations of live data and the limited database.

A final result of 0.89 (Hurricane Harvey) and 0.84 (2023 Turkey earthquake) was achieved, matching the generated ODET reports and the corresponding Wikipedia articles. These values confirm a remarkable agreement between the two sources, especially considering the different natures of the data sources. While Wikipedia is based on retrospective research and extensive source studies, ODET reports reflect information gathered in real time during the event.

5.2. Summary of Key Insights and Implications

In this paper, we present ODET, a modular platform that allows both research and disaster management communities to experiment with different algorithms and develop innovative solutions. Our case studies show that ODET is not only a flexible platform but also capable of generating highly accurate and structured reports that enable rapid situational assessment.

ODET serves as a platform for researchers to deploy and customize AI-based algorithms to generate reports tailored to specific scenarios. We demonstrated that an appropriate algorithm based on text embedding, community clustering, and zero-shot classification can be efficiently integrated into the ODET. Our case studies on Hurricane Harvey and the 2023 Turkey earthquake show that ODET also works reliably under the

challenges of real-world data sources, such as tweets. With AlignScore values of 0.89 and 0.84, respectively, the system demonstrated its ability to extract relevant and consistent information in a dynamic environment.

ODET's modular design allows for easy replacement of individual components or models, making the platform not only future-proof but also highly adaptable. This allows researchers to integrate new technologies or models and to test their own approaches without being tied to specific technologies. This flexibility means that ODET can run on standard hardware, as well as in high-performance cloud environments, making it suitable for both small and large datasets.

The results presented here show that ODET is a promising platform for quickly generating situation reports that provide decision-makers with the information they need. At the same time, ODET underscores its importance as a research platform that enables the development and testing of innovative AI-based algorithms in a controlled environment.

Future work should focus on further improving ODET, particularly on how to deal with misinformation and integrate new approaches to increase the accuracy of reports. Another well-known unsolved problem is hallucinations, which we are currently trying to minimize in the reports generated using a multi-step clustering and labeling approach as well as RAG and knowledge graphs, as these technologies provide a stronger link between content and underlying data. However, hallucinations cannot be eliminated at this time because they are a known unsolved problem in AI-based text generation. Further research is needed to develop more effective mechanisms for detecting and avoiding hallucinations. This is crucial to meet the growing demands of disaster management systems and to realize the full potential of ODET as a flexible and scalable platform.

Author Contributions: Conceptualization, K.S.; methodology, K.S.; validation, K.S., K.B. and M.H.; writing—original draft, K.S.; writing—review and editing, K.B., D.A.A. and M.H.; supervision, D.A.A. and M.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received external funding from Universidad de Granada and the European Union's Horizon 2020 research and innovation programme under grant agreement No [823759] called REMESH.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. These data can be found here: https://www.kaggle.com/datasets/gpreda/turkey-earthquake-tweets and here: https://www.kaggle.com/datasets/dan195/hurricaneharvey (accessed on 30 June 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ANN	Artificial Neural Network
API	Application Programming Interfaces
BERT	Bidirectional Encoder Representations from Transformers
CNN	Convolutional Neural Network
CRF	Conditional Random Fields
EWM	Entropy Weight Method
GCN	Graph Clustering Network
JSON	Java Script Object Notation
LDA	Latent Dirichlet Allocation
LLM	Large Language Model
LSTM	Long Short-Term Memory
MAS	Multi-Agent System
ML	Machine Learning

NER	Named Entity Recognition
NLP	Natural Language Processing
ODET	Open-Source Intelligence Disaster Event Tracker
OSINT	Open-Source Intelligence
RAG	Retrieval-Augmented Generation
REST	Representational State Transfer
SA	Sentiment Analysis
SMDRM	Social Media Disaster Risk Management
TwiSA	Twitter Situational Awareness

Appendix A

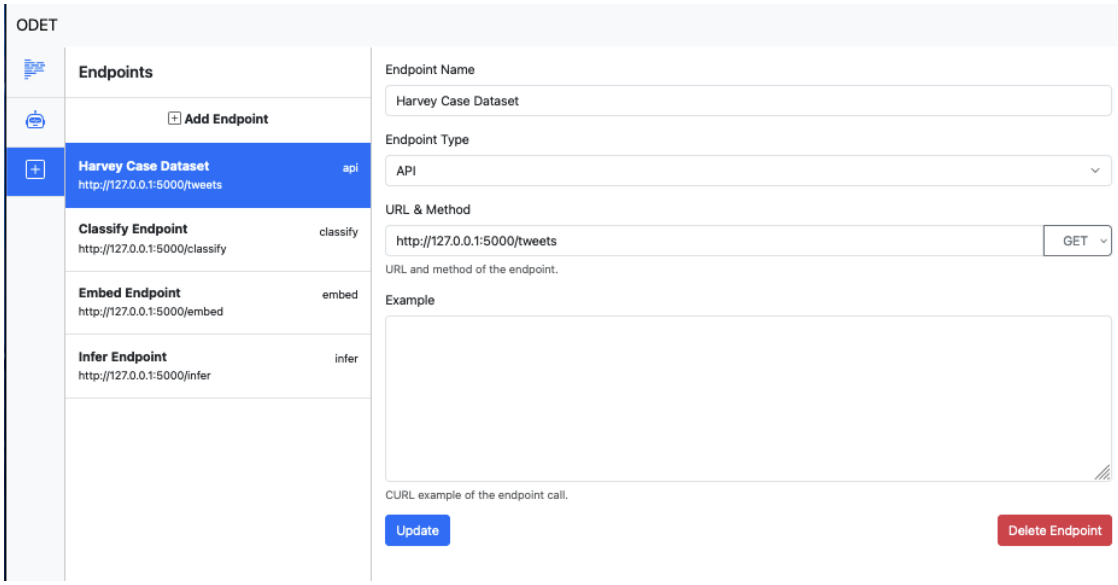

Figure A1. Overview of the endpoints user interface for the Hurricane Harvey case study in ODET.

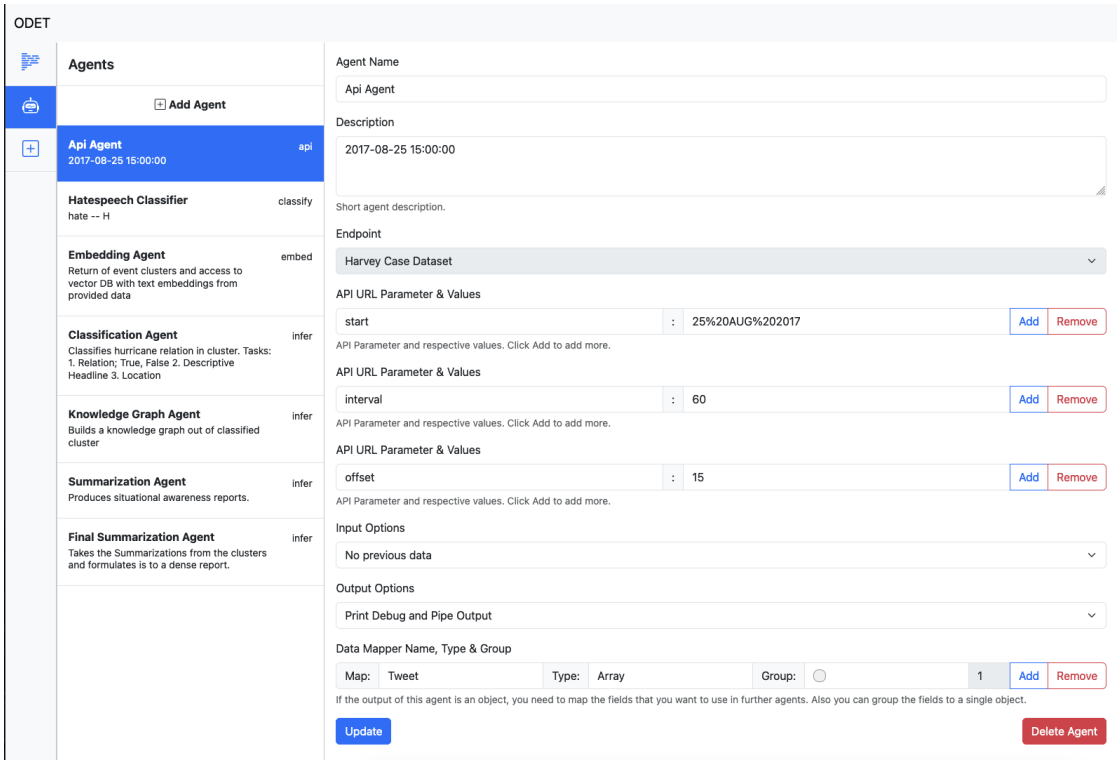

Figure A2. Overview of the API agent user interface for the hurricane Harvey case study in ODET.

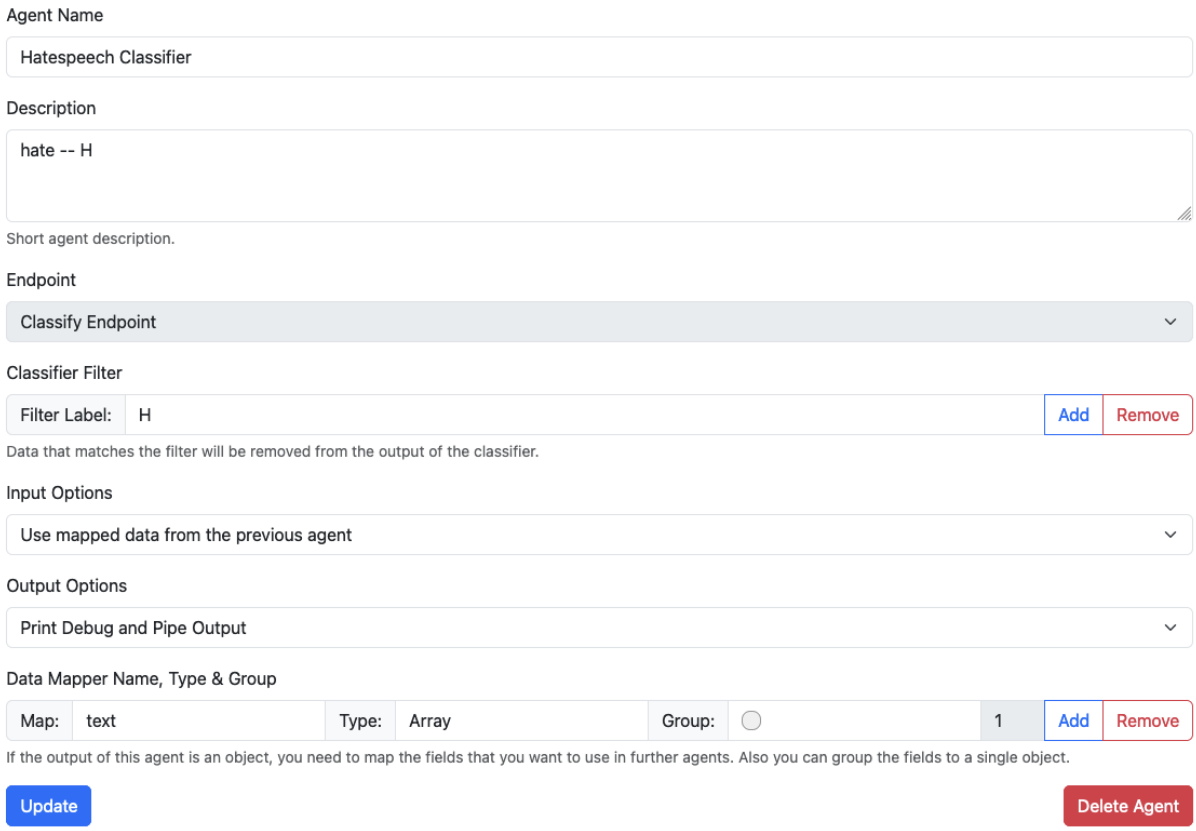

Figure A3. Overview of the hate-speech filter agent user interface for the hurricane Harvey case study in ODET.

Figure A4. Classification Agent—Grammar and JSON schema example.

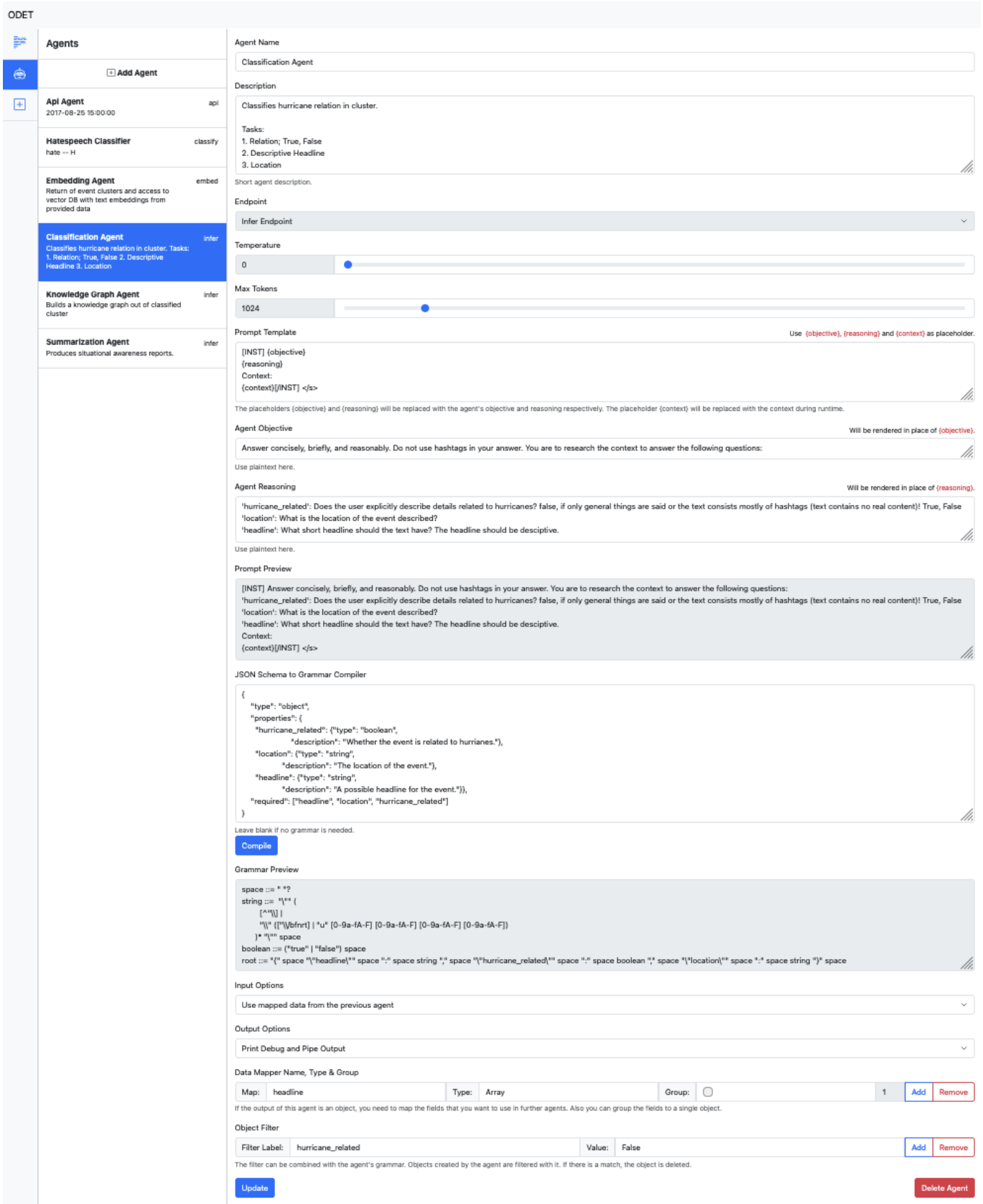

Figure A5. Overview of the classification agent user interface for the Hurricane Harvey case study in ODET.

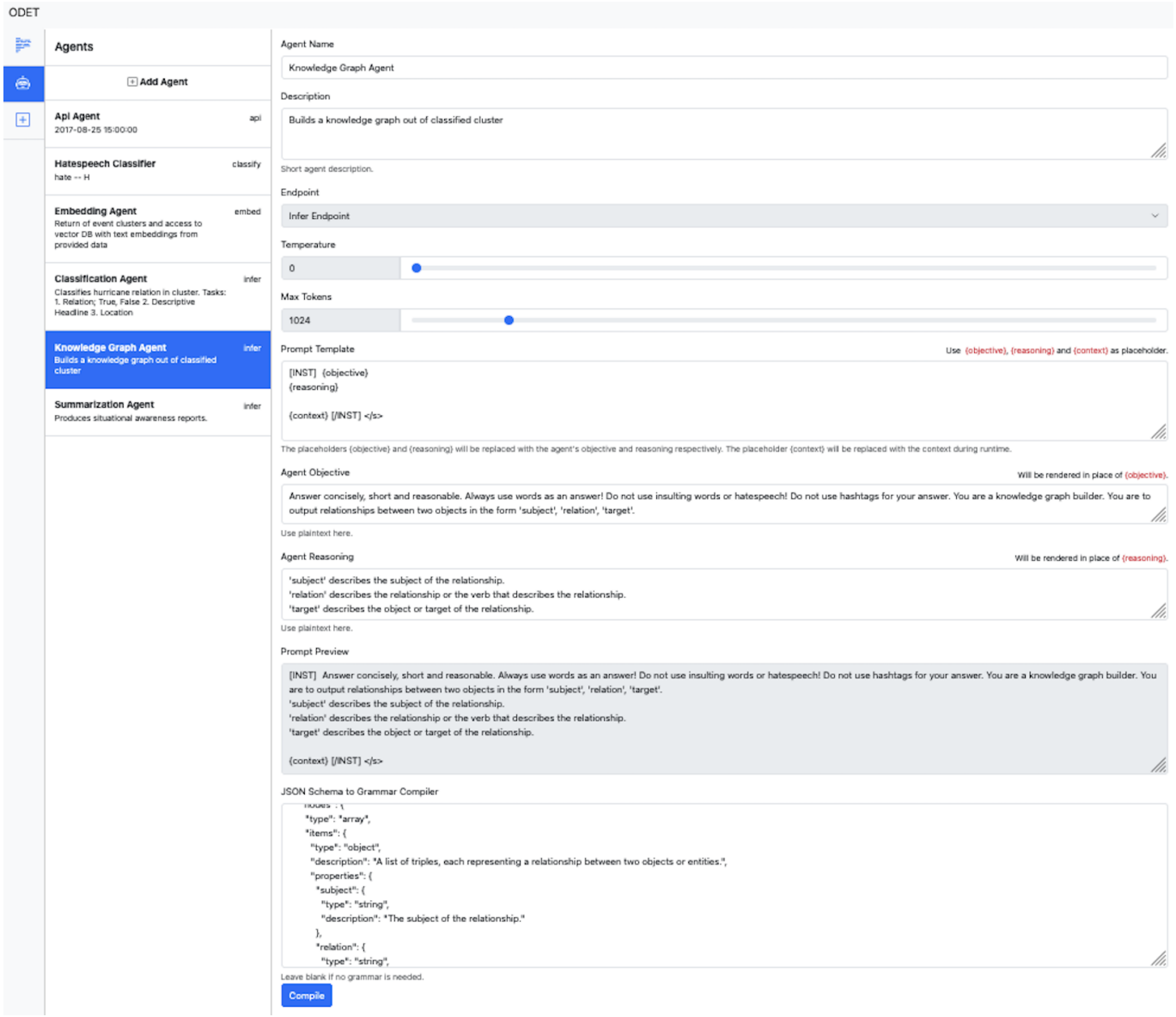

Figure A6. Overview of the knowledge graph agent user interface for the Hurricane Harvey case study in ODET.

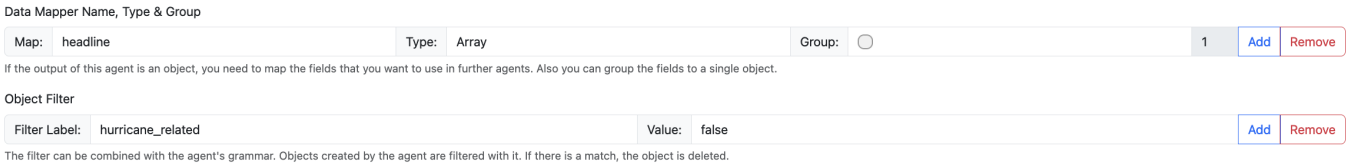

Figure A7. Classification agent—mapper and object filter example.

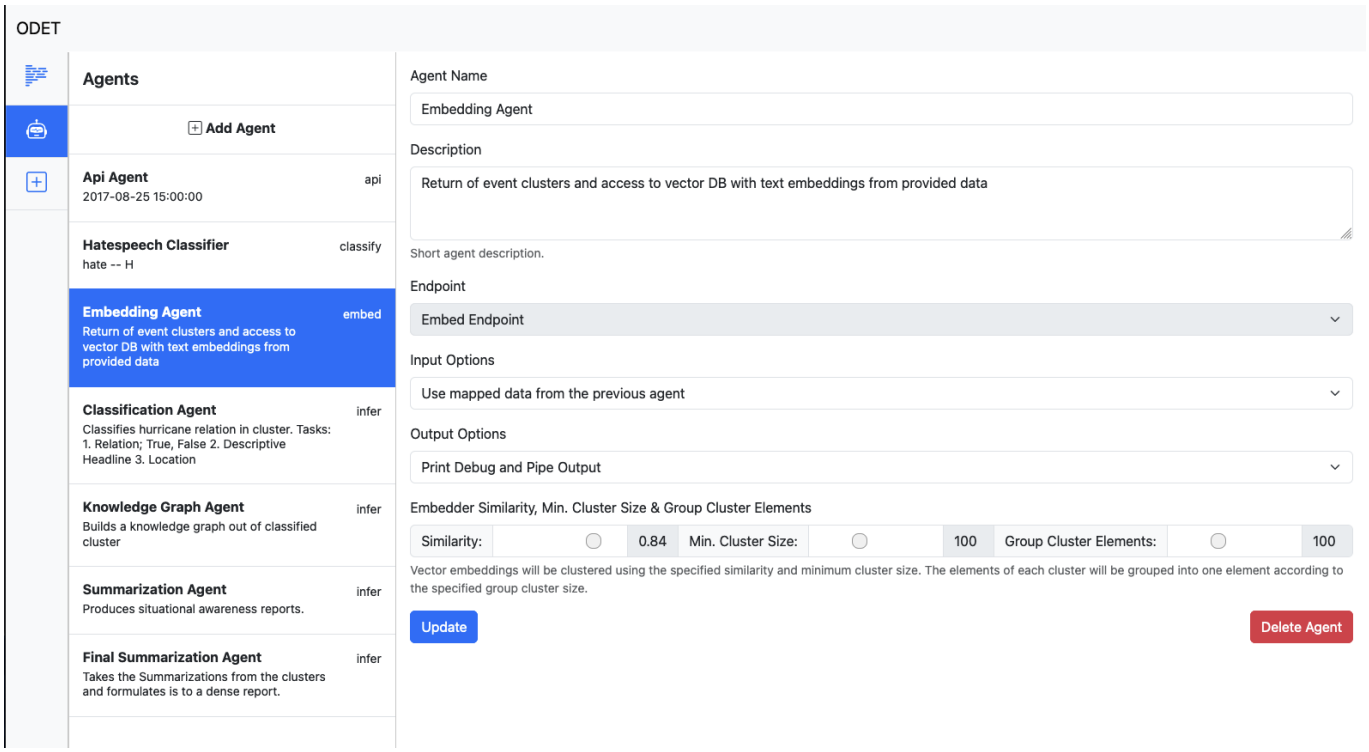

Figure A8. Overview of the embedding agent user interface for the hurricane Harvey case study in ODET.

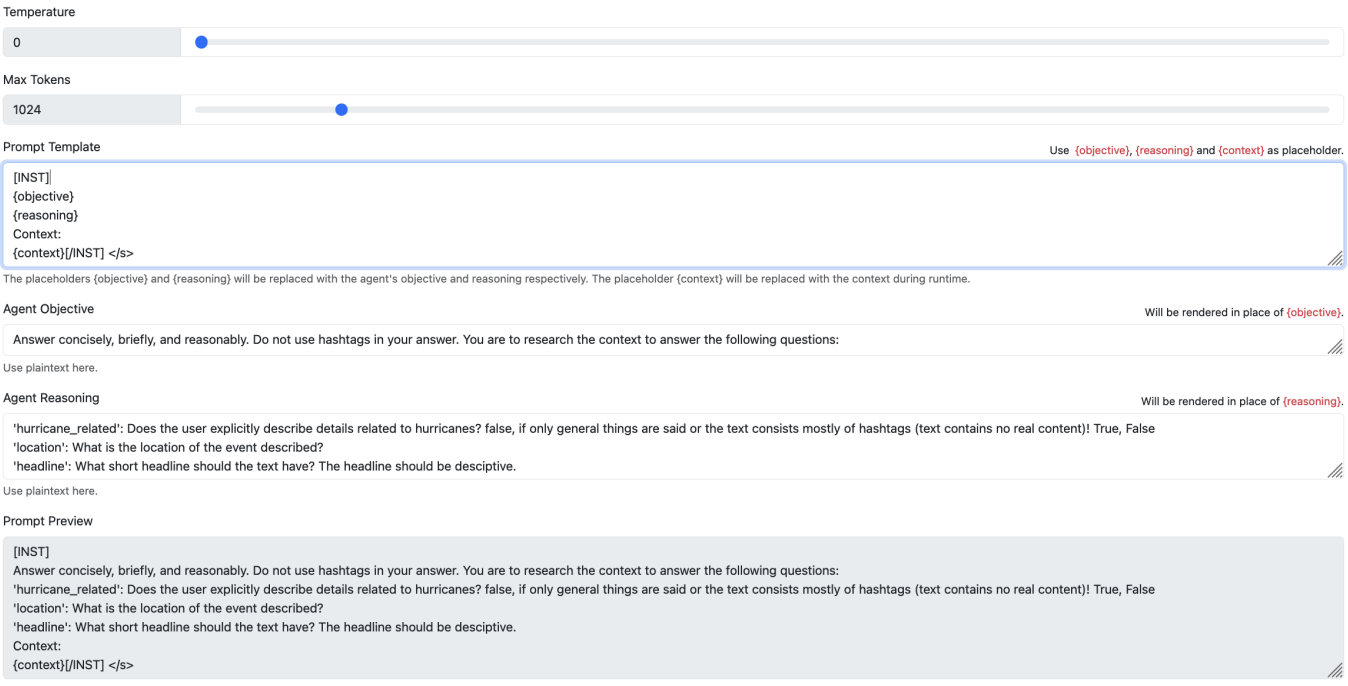

Figure A9. Classification Agent—Prompt template and objective example.

Figure A10. Knowledge Graph Agent—Grammar and JSON schema example.

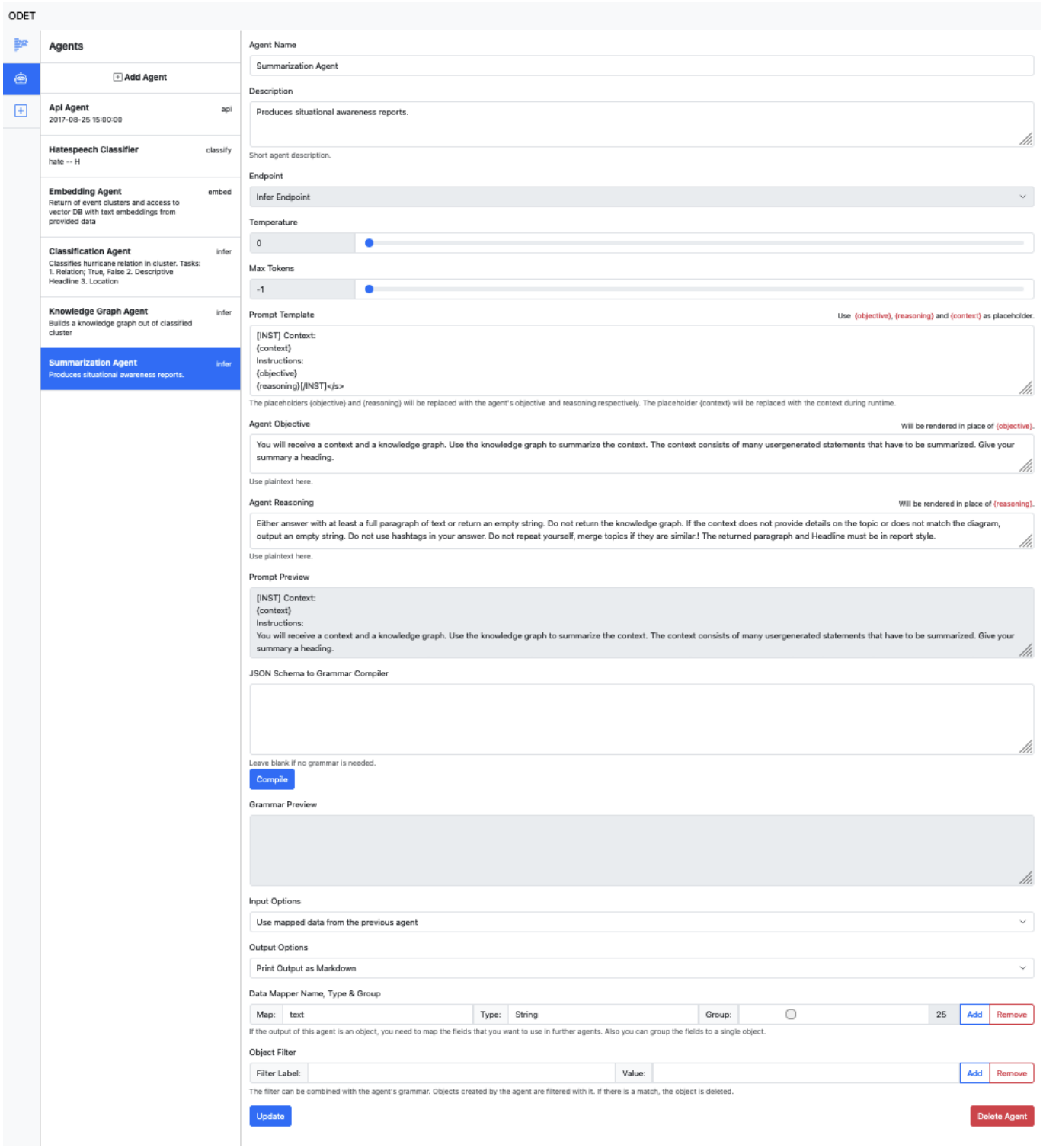

Figure A11. Overview of the summarization agent user interface for the Hurricane Harvey case study in ODET.

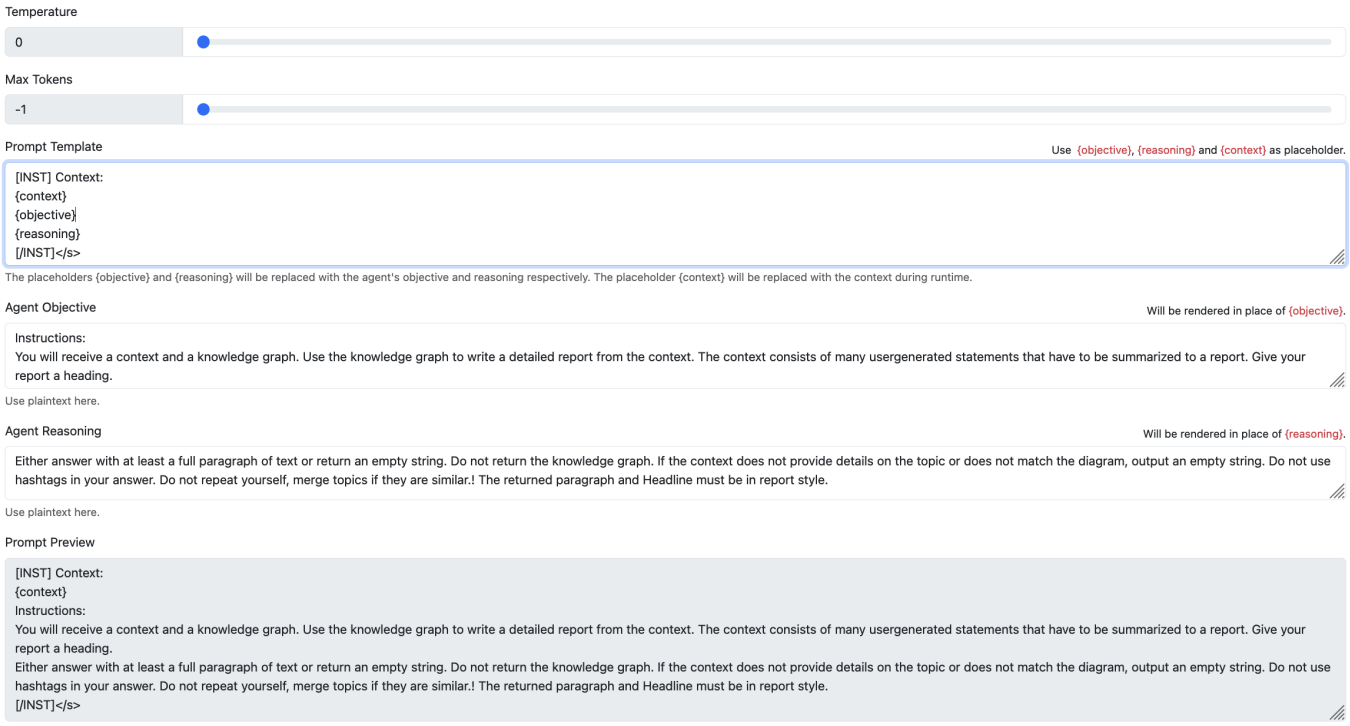

Figure A12. Summarization Agent—Prompt example.

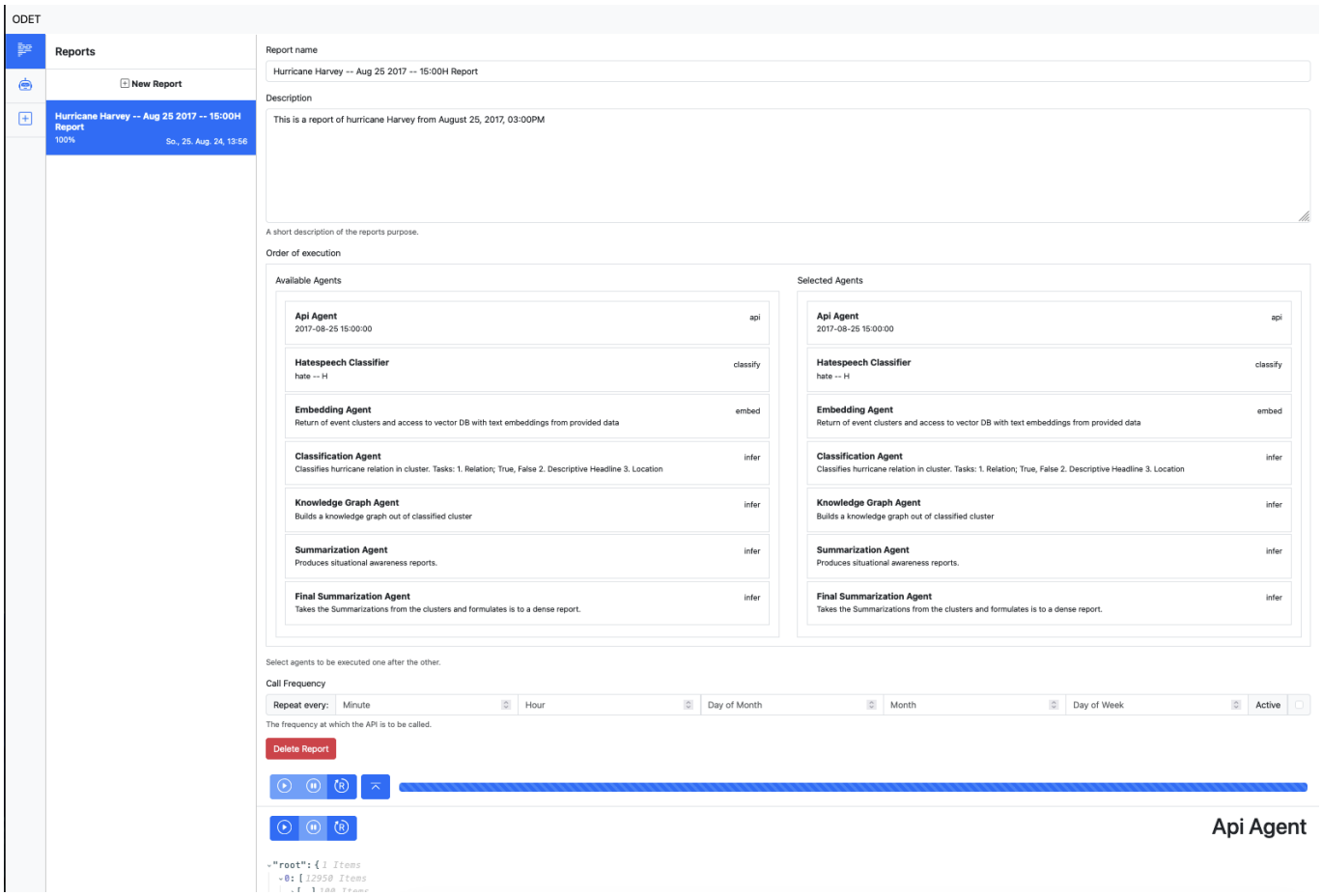

Figure A13. Overview of the report configuration user interface for the Hurricane Harvey case study in ODET.

References

1. Hackathon, M.S. Leveraging AI for Natural Disaster Management: Takeaways From The Moroccan Earthquake. *arXiv* **2023**, arXiv:2311.08999.
2. Kejriwal, M.; Fang, G.S.; Zhou, Y. A Feasibility Study of Open-Source Sentiment Analysis and Text Classification Systems on Disaster-Specific Social Media Data. In Proceedings of the 2021 IEEE Symposium Series on Computational Intelligence (SSCI), Orlando, FL, USA, 5–7 December 2021; pp. 1–8.
3. Lorini, V.; Panizio, E.; Castillo, C. SMDRM: A Platform to Analyze Social Media for Disaster Risk Management in Near Real Time. In Proceedings of the 16th International Conference on Web and Social Media, Atlanta, GA, USA, 6–9 June 2022.
4. Papadimos, T.; Pantelidis, N.; Andreadis, S.; Bozas, A.; Gialampoukidis, I.; Vrochidis, S.; Kompatsiaris, Y. Real-time Alert Framework for Fire Incidents Using Multimodal Event Detection on Social Media Streams. In Proceedings of the International Conference on Information Systems for Crisis Response and Management, Tarbes, France, 22–25 May 2022; pp. 623–635.
5. Abduraimov, M. Crisis Management in the Digital Age. *Qo'Qon Univ. Xabarnomasi* **2024**, *11*, 16–18. [CrossRef]
6. Zhou, X.; Chen, L. Migrating Social Event Recommendation Over Microblogs. *Proc. VLDB Endow.* **2022**, *15*, 3213–3225. [CrossRef]
7. Adelakun, B.O.; Antwi, B.O.; Ntiakoh, A.; Eziefule, A.O. Leveraging AI for sustainable accounting: Developing models for environmental impact assessment and reporting. *Financ. Account. Res. J.* **2024**, *6*, 1017–1048. [CrossRef]
8. Aljebreen, M.; Alabduallah, B.; Asiri, M.M.; Salama, A.S.; Assiri, M.; Ibrahim, S.S. Moth Flame Optimization With Hybrid Deep Learning Based Sentiment Classification Toward ChatGPT on Twitter. *IEEE Access* **2023**, *11*, 104984–104991. [CrossRef]
9. Zha, Y.; Yang, Y.; Li, R.; Hu, Z. AlignScore: Evaluating Factual Consistency with A Unified Alignment Function. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023; pp. 11328–11348.
10. United Nations: Office of the High Commissioner on Human Rights. *Berkeley Protocol on Digital Open Source Investigations*; United Nations: New York, NY, USA, 2022.
11. Al-Madhari, A.; Keller, A. Review of disaster definitions. *Prehospital Disaster Med.* **1997**, *12*, 17–21. [CrossRef]
12. Yahya, R.H.; Maharani, W.; Wijaya, R. Disaster Management Sentiment Analysis Using the BiLSTM Method. *J. Media Inform. Budidarma* **2023**, *7*, 501–508. [CrossRef]
13. Can, A.B.; Parlak, I.B.; Acarman, T. A New Method of Automatic Content Analysis in Disaster Management. In Proceedings of the 2022 10th International Symposium on Digital Forensics and Security (ISDFS), Istanbul, Turkey, 6–7 June 2022; pp. 1–5.
14. Baig, H.F.A.; Islam, N.; Alim, A. Exploring Machine Learning and Deep Learning Approaches for Disaster Prediction and Management: A Survey of Different Approaches. *Pak. J. Eng. Technol. Sci.* **2024**, *11*, 45–73.
15. Arapostathis, S. Utilising Twitter for disaster management of fire events: Steps towards efficient automation. *Arab. J. Geosci.* **2021**, *14*, 667. [CrossRef]
16. Mugeni, J.B.; Amagasa, T. A graph-based blocking approach for entity matching using pre-trained contextual embedding models. In Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing, Virtual, 25–29 April 2022.
17. Liu, W.; Xie, C.; Zong, S. A rumor source identification method based on node embeddings and community detection in social networks. In Proceedings of the 2023 Eleventh International Conference on Advanced Cloud and Big Data (CBD), Danzhou, China, 18–19 December 2023; pp. 104–109.
18. Canon, M.J.P.; Maceda, L.L.; Flores, N.M. Contextual Understanding of Academic-Related Responses Based on Enhanced Word Embeddings, Clustering, and Community Detection. *J. Adv. Inf. Technol.* **2024**, *15*, 693–703. [CrossRef]
19. Moura, A.; Lima, P.; Mendonça, F.; Mostafa, S.; Morgado-Dias, F. On the Use of Transformer-Based Models for Intent Detection Using Clustering Algorithms. *Appl. Sci.* **2023**, *13*, 5178. [CrossRef]
20. Liang, Y.; Zhu, J.; Ye, W.G.; Gao, S. Region2Vec: Community detection on spatial networks using graph embedding with node attributes and spatial interactions. In Proceedings of the 30th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, 1–4 November 2022.
21. Kalra, S.; Sharma, Y.; Agrawal, M.; Mantri, S.; Chauhan, G.S. Impact of Transformer-Based Models and User Clustering in Early Fake News Detection in Social Media. In Proceedings of the International Conference on Pattern Recognition Applications and Methods, Lisbon, Portugal, 22–24 February 2023; pp. 889–896.
22. Galeano, S.R. Zero-Shot Spam Email Classification Using Pre-trained Large Language Models. *arXiv* **2024**, arXiv:2405.15936.
23. Arco, F.M.P.d.; Nozza, D.; Hovy, D. Leveraging Label Variation in Large Language Models for Zero-Shot Text Classification. *arXiv* **2023**, arXiv:2307.12973.
24. Dr'apal, J.; Westermann, H.; Šavelka, J. Using Large Language Models to Support Thematic Analysis in Empirical Legal Studies. *arXiv* **2023**, arXiv:2310.18729.
25. Wang, Z.; Pang, Y.; Lin, Y. Large Language Models Are Zero-Shot Text Classifiers. *arXiv* **2023**, arXiv:2312.01044.
26. Raja, H.; Munawar, A.; Delsoz, M.; Elahi, M.; Madadi, Y.; Hassan, H.A.; Serhan, A.; Inam, O.; Hermández, L.; Tran, S.; et al. Using Large Language Models to Automate Category and Trend Analysis of Scientific Articles: An Application in Ophthalmology. *arXiv* **2023**, arXiv:2308.16688.
27. Afyouni, I.; Khan, A.; Al Aghbari, Z. E-ware: A big data system for the incremental discovery of spatio-temporal events from microblogs. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 13949–13968. [CrossRef]

28. Karimiziarani, M.; Moradkhani, H. Social response and Disaster management: Insights from twitter data Assimilation on Hurricane Ian. *Int. J. Disaster Risk Reduct.* **2023**, *95*, 103865. [CrossRef]
29. Ni, W.; Shen, Q.; Liu, T.; Zeng, Q.; Xu, L. Generating textual emergency plans for unconventional emergencies—A natural language processing approach. *Saf. Sci.* **2023**, *160*, 106047. [CrossRef]
30. Afyouni, I.; Khan, A.; Aghbari, Z.A. Deep-Eware: Spatio-temporal social event detection using a hybrid learning model. *J. Big Data* **2022**, *9*, 86. [CrossRef]
31. Huang, L.; Shi, P.; Zhu, H. An integrated urgency evaluation approach of relief demands for disasters based on social media data. *Int. J. Disaster Risk Reduct.* **2022**, *80*, 103208. [CrossRef]
32. Contreras, D.; Wilkinson, S.; Alterman, E.; Hervás, J. Accuracy of a pre-trained sentiment analysis (SA) classification model on tweets related to emergency response and early recovery assessment: The case of 2019 Albanian earthquake. *Nat. Hazards* **2022**, *113*, 403–421. [CrossRef] [PubMed]
33. Zhou, B.; Zou, L.; Mostafavi, A.; Lin, B.; Yang, M.; Gharaibeh, N.; Cai, H.; Abedin, J.; Mandal, D. VictimFinder: Harvesting rescue requests in disaster response from social media with BERT. *Comput. Environ. Urban Syst.* **2022**, *95*, 101824. [CrossRef]
34. Wahid, J.A.; Shi, L.; Gao, Y.; Yang, B.; Wei, L.; Tao, Y.; Hussain, S.; Ayoub, M.; Yagoub, I. Topic2Labels: A framework to annotate and classify the social media data through LDA topics and deep learning models for crisis response. *Expert Syst. Appl.* **2022**, *195*, 116562. [CrossRef]
35. Contreras, D.; Wilkinson, S.; Balan, N.; James, P. Assessing post-disaster recovery using sentiment analysis: The case of L'Aquila, Italy. *Earthq. Spectra* **2022**, *38*, 81–108. [CrossRef]
36. Bashir, S.; Bano, S.; Shueb, S.; Gul, S.; Mir, A.A.; Ashraf, R.; Shakeela; Noor, N. Twitter chirps for Syrian people: Sentiment analysis of tweets related to Syria Chemical Attack. *Int. J. Disaster Risk Reduct.* **2021**, *62*, 102397. [CrossRef]
37. Behl, S.; Rao, A.; Aggarwal, S.; Chadha, S.; Pannu, H. Twitter for disaster relief through sentiment analysis for COVID-19 and natural hazard crises. *Int. J. Disaster Risk Reduct.* **2021**, *55*, 102101. [CrossRef]
38. Lian, Y.; Liu, Y.; Dong, X. Strategies for controlling false online information during natural disasters: The case of Typhoon Mangkhut in China. *Technol. Soc.* **2020**, *62*, 101265. [CrossRef]
39. Karami, A.; Shah, V.; Vaezi, R.; Bansal, A. Twitter speaks: A case of national disaster situational awareness. *J. Inf. Sci.* **2020**, *46*, 313–324. [CrossRef]
40. Farnaghi, M.; Ghaemi, Z.; Mansourian, A. Dynamic Spatio-Temporal Tweet Mining for Event Detection: A Case Study of Hurricane Florence. *Int. J. Disaster Risk Sci.* **2020**, *11*, 378–393. [CrossRef]
41. Gulnerman, A.G.; Karaman, H. Spatial Reliability Assessment of Social Media Mining Techniques with Regard to Disaster Domain-Based Filtering. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 245. [CrossRef]
42. Fan, C.; Wu, F.; Mostafavi, A. A Hybrid Machine Learning Pipeline for Automated Mapping of Events and Locations From Social Media in Disasters. *IEEE Access* **2020**, *8*, 10478–10490. [CrossRef]
43. Singh, N.; Roy, N.; Gangopadhyay, A. Analyzing The Emotions of Crowd For Improving The Emergency Response Services. *Pervasive Mob. Comput.* **2019**, *58*, 101018. [CrossRef]
44. Gordon, D.S.S.E. Hurricane Harvey Tweets. 2017. Available online: https://www.kaggle.com/datasets/dan195/hurricaneharvey (accessed on 19 October 2024).
45. Preda, G.D.S. Turkey Earthquake Tweets. 2023. Available online: https://www.kaggle.com/datasets/gpreda/turkey-earthquake-tweets/data (accessed on 19 October 2024).
46. KoalaAI. KoalaAI/Text-Moderation. Hugging Face Repository. 2024. Available online: https://huggingface.co/KoalaAI/Text-Moderation (accessed on 25 May 2024).
47. Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; Delangue, C.; Moi, A.; Cistac, P.; Rault, T.; Louf, R.; Funtowicz, M.; et al. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv* **2019**, arXiv:1910.03771.
48. Solatorio, A.V. GISTEmbed: Guided In-sample Selection of Training Negatives for Text Embedding Fine-tuning. *arXiv* **2024**, arXiv:2402.16829.
49. Rebedea, T.; Dinu, R.; Sreedhar, M.N.; Parisien, C.; Cohen, J. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023; pp. 431–445.
50. Zhao, P.; Zhang, H.; Yu, Q.; Wang, Z.; Geng, Y.; Fu, F.; Yang, L.; Zhang, W.; Cui, B. Retrieval-Augmented Generation for AI-Generated Content: A Survey. *arXiv* **2024**, arXiv:2402.19473.
51. Yu, H.; Gan, A.; Zhang, K.; Tong, S.; Liu, Q.; Liu, Z. Evaluation of Retrieval-Augmented Generation: A Survey. *arXiv* **2024**, arXiv:2405.07437.
52. Su, W.; Tang, Y.; Ai, Q.; Wu, Z.; Liu, Y. DRAGIN: Dynamic Retrieval Augmented Generation based on the Real-time Information Needs of Large Language Models. *arXiv* **2024**, arXiv:2403.10081.
53. Gupta, V.; Srinivasa Rao, P. Leveraging open-source models for legal language modeling and analysis: A case study on the Indian constitution. *arXiv* **2024**, arXiv:2404.06751.

54. Blake, E.S.; Landsea, C.; Gibney, E.J. The deadliest, costliest, and most intense United States tropical cyclones from 1851 to 2010 (and other frequently requested hurricane facts). In *NOAA Technical Memorandum NWS NHC-6*; National Oceanic and Atmospheric Administration, National Weather Service, National Hurricane Center: Miami, FL, USA, 2017.
55. Maletckii, B.; Astafyeva, E.; Sanchez, S.; Kherani, E.; De Paula, E. The 6 February 2023 Türkiye earthquake sequence as detected in the ionosphere. *J. Geophys. Res. Space Phys.* **2023**, *128*, e2023JA031663. [CrossRef]