Supplementary Material for

"A Generic Ontology-Driven Information Extraction and Impact Analysis of Hazards: A Case Study on the 2024 Wayanad Landslides"

Lakshmi S. Gopal[1], Hemalatha Thirugnanam[1], Tejal Shah[2], Maneesha Vinodini Ramesh[1]

1. Center for Wireless Networks & Applications (WNA), Amrita Vishwa Vidyapeetham, Amritapuri, India

2. School of Computing, Newcastle University, Newcastle Upon Tyne, UK

Contents Page No.

1. **Natural Hazards Insight Ontology Overview**

Ontology plays a critical role in structuring domain knowledge for intelligent information extraction systems. In the context of disaster impact analysis, an ontology provides a formal representation of key concepts, including hazards, affected entities, impacts, locations, timeframes, and response resources, along with the relationships among them. This structured vocabulary enables consistent annotation of data, supports reasoning, and facilitates semantic querying.

The ontology described in this work, referred to as the ***Natural Hazards Insight Ontology***, has been designed with an event-centric, modular approach, supporting n-ary relations that capture complex disaster scenarios. This section outlines the core classes, object and data properties, modeling patterns, and sample assertions used to represent multi-dimensional disaster information.

1.1. **Classes and Sub-Classes**

In an ontology, classes represent the domain's primary concepts, defining key entities within a specific context. Each class contains sub-classes that represent keywords related to their respective class concepts. In the case of the proposed ontology, which focuses on disaster impact analysis, the classes are designed to capture critical components relevant to disaster events and their aftermath.

The ontology includes a ***Disaster*** class representing natural hazard events, such as landslides, floods, and storms, which serve as root triggers in the event modeling framework. The ***Aftermath*** class explores the consequences of these hazards, encompassing impact-related concepts such as casualties, injuries, damage, and rescue. This separation allows for a clear distinction between the cause and its effects, enabling the accurate linkage of event triggers to their observed outcomes in real-world scenarios.

One of the key classes in the ontology represents impacts and the ***AffectedEntity***, capturing entities related to people and animals affected by disasters. This includes information about affected entities such as animals, cattle, and people, including men, women, children, and the elderly, which is crucial for understanding the scale of human and animal impact during and after a disaster event.

Another important category focuses on ***Damage***-related classes, specifically designed to capture various types of damage. This includes infrastructure damage, covering structural impacts on buildings, bridges, roads, power lines, and other critical facilities, as well as agricultural damage, which represents the impact on crops, farmland, livestock, and agricultural produce. Additionally, a class dedicated to ***Resource*** requirements helps identify essential resources needed during and after disasters, such as food, water, medical supplies, and shelter materials.

The ontology also incorporates classes related to ***SpatialData*** information to capture location-based data that aids first responders. This includes details like place names (such as cities, districts, villages, and landmarks) and geographical coordinates, including latitude and longitude information, which are critical for precisely identifying affected regions. To handle

quantitative data, a dedicated class for quantitative measurements is included, namely, ***Quantity***, representing measurements such as the area affected (e.g., hectares of agricultural land damaged), and count data (e.g., the number of casualties, injured, rescued individuals, houses destroyed).

Recognizing the role of support organizations in disaster response, the ontology includes a class titled ***Support***, which represents organizations and agencies involved in disaster management and recovery. This class encompasses entities such as government bodies like the National Disaster Management Authority (NDMA), rescue forces such as the army, police, and fire departments, healthcare services including hospitals and emergency medical teams, and non-governmental organizations (NGOs) that provide humanitarian aid and relief services. In addition to these, the ontology features classes to capture ***TemporalData*** entities, recognizing the importance of time-related information in disaster management. Temporal classes cover aspects such as the event date when the disaster occurred, the duration of the disaster, and its aftermath.

The ontology supports both binary and n-ary modeling patterns. While binary relations are used to directly associate two classes, such as linking a Disaster to a Location or an Affected entity to a Resource, more complex situations require an n-ary structure. To address this, the ***Event*** class is introduced as a central node that brings together multiple related aspects of a disaster occurrence, including the type of disaster, its location, time of occurrence, observed impacts (e.g., casualties, damages), response entities, and quantitative measures. This enables richer semantic representation and supports flexible querying over multi-faceted data using SPARQL.

1.2. **Object Properties**

In an ontology, object properties define the relationships between different classes and their instances, enabling the representation of meaningful associations within the domain. In the proposed Natural Hazards Insight Ontology, both ***binary*** and ***n-ary*** object properties are defined to capture simple and complex relationships, respectively. Binary properties link two entities directly, for example, a Disaster and its Location, while n-ary relations are modeled using an intermediate ***Event class*** that connects multiple entities, such as Disaster, Date, Location, Affected Population, and Response Actions.

The following are the binary object properties:

- "***Causes***" - The causes property establishes a causal relationship between the "*Disaster*" class and the "*Aftermath*" class. This property is used to describe the impact or consequence of a disaster event. For example, in the sentence, "The landslide causes death," the object property connects the "landslide" (an instance of the Disaster class) to "death" (an instance of the Aftermath class). This relationship helps in identifying and quantifying the direct consequences of disasters.
- "***isQuantifiedBy***" - This property links the "Aftermath" class to the "Quantity" class. It is used to represent how specific impacts, such as casualties or damages, are quantified. This property captures both exact and approximate numerical values

associated with disaster impacts. For example, "Death “isQuantifiedBy” ExactCount (10)" or "Death “isQuantifiedBy” ApproxCount (thousands)." Here, "death" is connected to its corresponding quantity, whether it is an exact number or an approximate estimate.

- "***locatedIn***" - This property connects the "*Disaster*" class to the "*SpatialData*" class. This property is used to specify the geographical location where a disaster event occurred. For example, "The landslide “locatedIn” district (Wayanad)". In this case, the disaster event "landslide" is linked to the spatial entity "Wayanad", indicating the affected location. This property is vital for spatial analysis and helps first responders identify impacted areas.
- "***occurredOn***" - The “occurredOn” property represents the temporal relationship between the "*Disaster*" class and the "*TemporalData*" class. It is used to capture the date, time, or duration of a disaster event. For example, "The landslide “occurredOn” date (10-12-2024)". Here, the disaster event "landslide" is associated with the date it occurred, enabling temporal analysis and tracking of events over time.
- "***reliefBy***" - The “reliefBy” property connects the "*Affected*" class to the "*Resource*" class. This property represents the resources provided to affected individuals or communities during disaster response and recovery efforts. For example, "People reliefBy shelter". In this instance, the affected "people" are linked to the resource "shelter", indicating the type of relief provided. This property is essential for understanding the distribution of resources during disasters.
- "***resourceRequiredBy***" - The “resourceRequiredBy” property is the inverse of the “reliefBy” property. It connects the "*Resource*" class to the "*Affected*" class, representing the specific resources required by affected individuals. For example, "Clothes resourceRequiredBy people". Here, the resource "clothes" is linked to the "people" who require it. This relationship is crucial for identifying resource needs during disaster situations, aiding in effective resource allocation.
- "***rescuedBy***" - The “rescuedBy” property links the "*Affected*" class to the "*Support*" class. This property represents the entities or organizations responsible for rescuing affected individuals during disaster events. For example, "People ‘rescuedBy’ Army". In this case, the affected "people" are associated with the "Army", indicating the organization that conducted the rescue operations. This property helps in tracking the involvement of various support agencies during disaster response activities.

While binary object properties suffice for modeling simple pairwise relationships, ***n-ary object properties*** are essential when a relationship involves more than two entities. In the *Natural Hazards Insight Ontology*, such complex relationships are modeled through the introduction of an intermediary class called ***Event***. This design pattern allows multiple related components, such as Disaster Type, Location, Date, Death Count, Support Agency, Action Taken, and Resources Deployed, to be linked together as part of a single, coherent event instance. The following are the n-ary object properties:

- "***hasAction***" - Links an *Event* to the *Aftermath* or action taken during or after the disaster (e.g., recovery, evacuation, rescue operations). It captures what was done in response to the event.
- "***hasActor***" - Connects an *Event* to the *Support* class, indicating the agencies or organizations (like NDRF, Army, or Volunteers) that participated in handling the disaster event.
- "***hasDate***" - Associates an *Event* with a *TemporalData* instance to specify when the event occurred, such as a specific date or period.
- "***hasDisaster***" - Links the *Event* to a *Disaster* class individual (e.g., landslide, flood), specifying what type of disaster the event represents.
- "***hasLocation***" - Connects an *Event* to a *SpatialData* individual (e.g., Wayanad, Chooralmala), representing where the event took place.
- "***hasQuantity***" - Points from an *Event* to a *Quantity* individual (e.g., death count, injured, rescued), helping quantify the impact of the disaster.
- "***hasRecipient***" - Associates the *Event* with an *Affected* class individual (e.g., women, children), indicating who was impacted by the disaster.
- "***hasResource***" - Connects an *Event* to a *Resource* individual (e.g., food, medicine), specifying what aid or materials were deployed or required during the event.

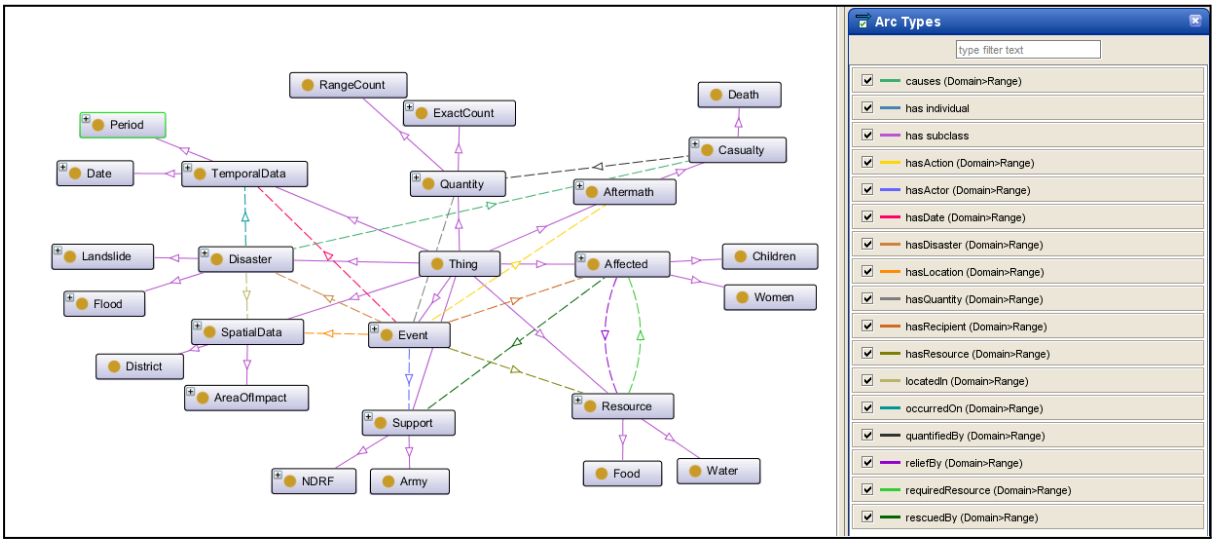

Figure 1: Main classes, sub-classes, and object properties of Natural Hazards Insight Ontology

1.3. Data Properties

Data properties in an ontology are attributes that link classes or instances to specific data values, such as numerical counts, textual descriptions, or temporal information. These properties are essential for capturing quantifiable and descriptive information associated with various disaster-related entities. In the proposed ontology, data properties are defined to handle integer, float, and string data types, enabling the representation of numerical counts, measurements, and descriptive details relevant to disaster impact analysis.

The following are the data properties defined:

- "***agricultureDamage***" - Represents damage to agricultural areas caused by the disaster, often quantified in hectares or descriptive terms (e.g., "30 hectares" or "major crop loss").
- "***deathCount***" - Stores the number of fatalities resulting from the event. Typically, an integer value (e.g., 70).
- "***disasterEvent***" - Captures the label or name of the specific disaster (e.g., "landslide"), stored as a string for identification.
- "***environmentDamage***" - Describes environmental impact due to the disaster, such as deforestation, water pollution, or ecosystem loss. Can be a qualitative or quantitative string.
- "***eventAction***" - Indicates specific actions taken during or after the disaster (e.g., "evacuation", "rescue"), describing the type of aftermath operation.
- "***eventDate***" - Stores the exact date the disaster occurred, formatted as a date (e.g., "2024-07-29").
- "***eventPeriod***" - Represents a period associated with the event, such as "last week" or "July 2024", stored as a string.
- "***infrastructureDamage***" - Describes the extent of damage to physical infrastructure (e.g., "bridge collapse", "200 houses affected"), can be stored as an integer or a string.
- "***injuredCount***" - Number of individuals injured due to the disaster. An integer value.
- "***missingCount***" - Captures the number of people reported missing after the event, also an integer.
- "***placeName***" - Stores the name of the location affected (e.g., "Wayanad", "Chooralmala"), a string.
- "***rescueBy***" - Names the organization or actor responsible for rescue operations (e.g., "NDRF"), stored as a string.
- "***rescueCount***" - Indicates how many individuals were rescued. An integer value.

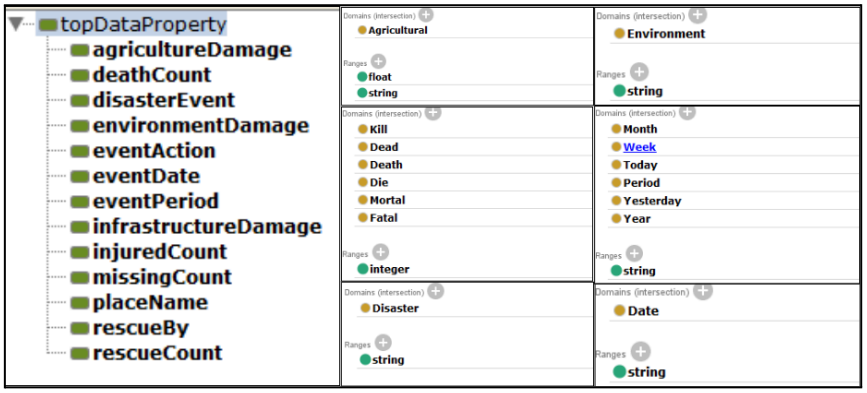

Figure 2: Data Properties and examples of Natural Hazards Insight Ontology

1.4. **Annotation Properties**

Annotation properties in an ontology are used to add metadata or supplementary information to classes, subclasses, properties, or instances. Unlike object or data properties, which establish formal relationships between entities or link entities to data values, annotation properties provide descriptive information that enhances the semantic understanding of ontology elements. This additional layer of information is crucial in improving the performance of IE systems, as it aids in maintaining contextual accuracy during data extraction.

In the *Natural Hazards Insight Ontology*, two key annotation properties are incorporated to support the IE system:

- "***hasPoS***" - This annotation property represents all possible Part-of-Speech (PoS) tags that a particular class or subclass can have. By associating PoS tags with ontology concepts, the IE system can accurately identify and classify words based on their grammatical roles within sentences. For instance, the subclass "dead" from the "casualty" class is assigned with "NN" (Common Noun), "VB" (Verb, base form), "ADJ" (Adjective Phrase), "JJ" (Adjective), and "VBD" (Verb, past tense) PoS tags using the "hasPoS" annotation property.
- "***hasDep***" - This property captures all possible syntactic dependency relations associated with a class or subclass. Dependency relations define how words are connected in a sentence, which helps the IE system understand the grammatical structure and semantic dependencies of key terms. This enables the system to extract relevant information while preserving the original text's context. For instance, the subclass "dead" from the "casualty" class is assigned with "nsubj" (nominal subject), "amod" (adjectival modifier), "dobj" (direct object), and "nmod" (modifier of nominal) dependencies using the "hasDep" annotation property.

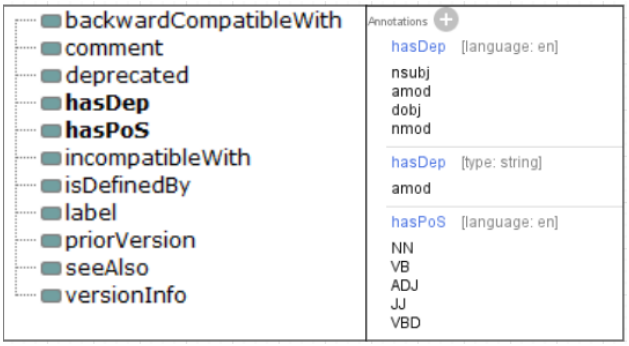

Figure 3: Annotation Properties with Example of the class “dead”.

1.5. **Individual Assertions and Sample Data**

In the ontology, individuals represent real-world entities such as events, people, organizations, or places, while classes group these individuals based on shared characteristics. To associate real-world knowledge with these classes, object and data

properties are used; object properties link individuals to other individuals, and data properties associate individuals with literal values such as numbers or dates.

The following example illustrates a sample individual assertion modeled in the *Natural Hazards Insight Ontology* to demonstrate how event-level relationships are represented.

In the *Natural Hazards Insight Ontology*, each sentence from the dataset is modeled as a distinct event individual (e.g., Event001, Event002, ..., Event00n). Each of these event individuals forms the central node in an n-ary relationship, connected to multiple other individuals via object properties like hasDisaster, hasAction, hasLocation, hasDate, and hasQuantity. Each of these object properties is assigned values that originate from the NLP-based information extraction module (detailed in Section III of the paper). The resulting structure captures rich semantic relationships for each news sentence.

Example:

For the sentence "A landslide killed 70 people in Wayanad on July 30th", the individual Event001 is created with the following assertions:

- *hasDisaster* → *disasterEvent001* (whose *disasterEvent* is "landslide")
- *hasAction* → *action001* (whose *eventAction* is "kill")
- *hasLocation* → *placeName001* (whose *placeName* is "Wayanad")
- *hasDate* → *date001* (whose *eventDate* is "July 30")
- *hasQuantity* → *quantity001* (whose *deathCount* is 70)

Each referenced individual (e.g., action001, quantity001) is further described using data properties to hold specific values.

This n-ary modeling enables the capture of multiple aspects of an event in a structured, queryable format, which is not possible with binary relations alone.

In the current prototype, these individuals are manually added within Protégé to facilitate SPARQL query testing and validation. Moving forward, this process will be automated, with individuals being programmatically created based on NLP outputs. These assertions will be stored in the ontology and linked with external systems like a dashboard, supporting dynamic and seamless data visualization across systems.

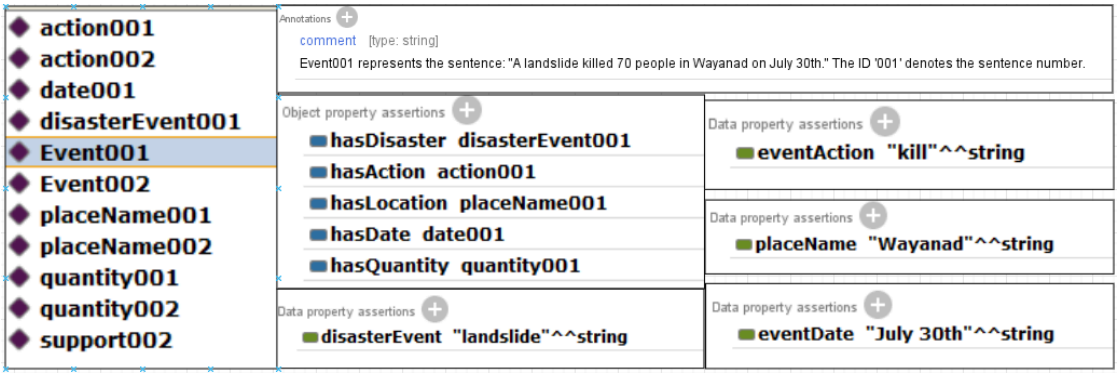

Figure 4: Individual Assertions Example - Sample sentence from dataset created as 'Event001' along with information extracted as the object properties

Note: The ontology is continuously being refined as part of ongoing work to automate individual creation based on NLP-derived outputs. As a result, the examples and instances presented in this section represent the earlier prototype version and may differ from the most recent ontology updates.

1.6. **SPARQL queries and results**

To retrieve structured information from the ontology, SPARQL (SPARQL Protocol and RDF Query Language) is used. It allows querying RDF data by matching patterns in the underlying graph. Since the proposed ontology models both binary and n-ary relationships through well-defined object and data properties, SPARQL enables flexible retrieval of complex event-related insights.

Each sentence from the news corpus is represented as an individual event, connected to other concept individuals (e.g., location, quantity, date) through object properties such as *hasDisaster*, *hasQuantity*, *hasDate*, and *hasLocation*. These relationships are then queried using SPARQL for analysis, validation, or visualization.

The examples below demonstrate how such queries are executed within Protégé to retrieve values like death counts, rescue agencies, locations, and event-specific actions:

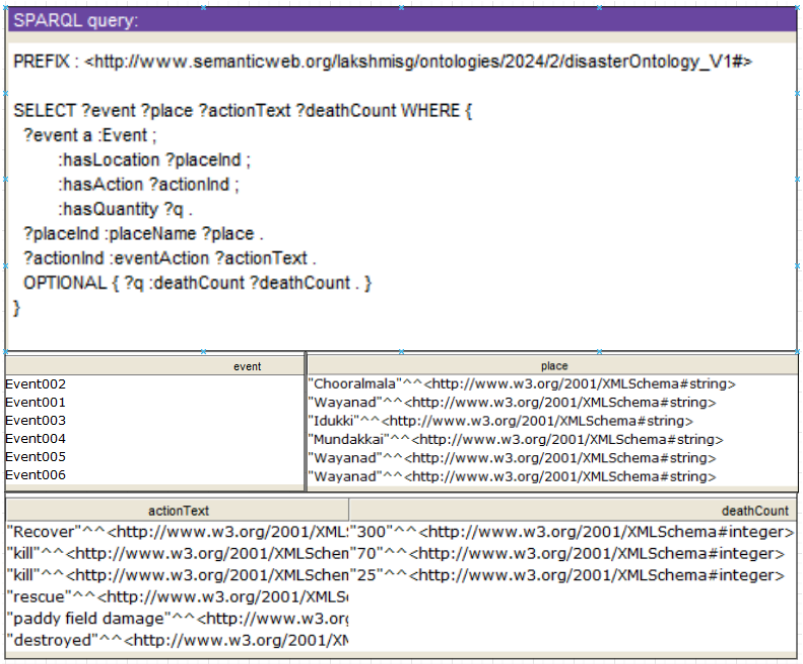

Figure 5: SPARQL query to extract the location-wise action and death count

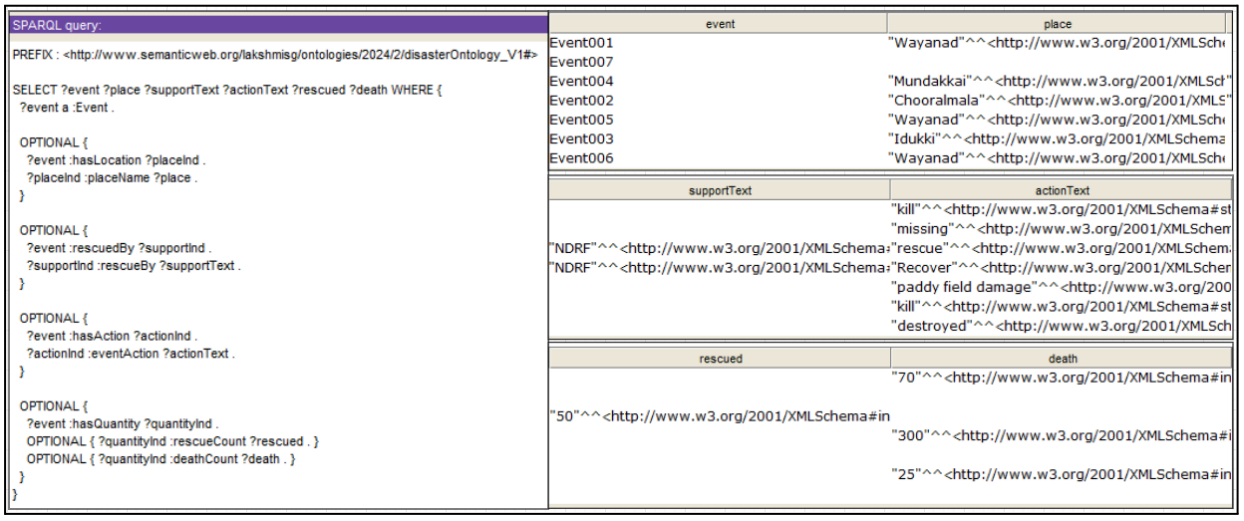

Figure 6: SPARQL query to extract the location-wise action, support information, rescue, and death count

Note: The SPARQL queries and outputs shown here correspond to an earlier version of the ontology. Since the ontology is being continuously updated through automated individual creation and refinement, query results may vary in the latest version.

1.7. Comparative Evaluation of Static and Dynamic Ontology

Table 1 presents a comparison of the static and dynamic versions of the developed ontology in the context of their application across various system modules. The comparison is structured around key operational factors such as relevance to web crawling and scraping, triple and relation extraction, scalability, and reusability.

The static ontology is manually curated and performs well in terms of precision and reliability, especially when used in controlled or domain-specific contexts. However, it lacks adaptability and requires expert intervention for updates, which limits its scalability to new or evolving disaster scenarios.

In contrast, the dynamic ontology is designed to self-update by incorporating new keywords and syntactic patterns from input data, thereby improving recall and reducing manual maintenance. While it introduces a higher dependency on NLP accuracy and may require validation for newly learned entries, its adaptability makes it suitable for long-term and cross-domain use cases.

Factors	Static Ontology	Dynamic Ontology
Usage in Web Crawling & Scraping	May not target new or unexpected keywords in the URL or page content	Tracks new emerging terms
Usage in IE System (Triple Extraction)	High precision due to predefined PoS and dependency patterns	Better recall as new patterns/terms get incorporated, but precision may vary
Usage in IE System (Relation Extraction)	Extracts only known and pre-defined relation types and their structures	Can learn and integrate new structures dynamically, but needs verification
Scalability to New Events	Limited; must be manually updated with terms from new disaster scenarios	Scales better with evolving news content by learning new disaster-specific terms and relations
Accuracy	High	Moderate to High; depends on NLP model accuracy
Dependency on NLP	Low - human-defined PoS and dependencies	High - relies on accurate PoS tagging and parsing
Maintenance Effort	High - requires expert	Low - minimal manual input

	intervention for updates	once deployed
Reusability	Easy to reuse with documentation, but domain-bound	Requires interpretation of dynamic terms, but more adaptable for future or other domain use cases

Table 1: Comparison of Static and Dynamic Ontology in System Modules

2. Web-based news data collection

To collect real-time disaster-related news articles, a web crawler and scraper were developed as part of the system architecture. These components form the data acquisition layer of the pipeline, responsible for sourcing and filtering relevant content from the web. The crawler initiates from a set of seed URLs derived from domain-relevant news portals. It traverses hyperlinks recursively up to a user-defined depth and maintains a frontier queue to manage URL discovery. A filtering step based on ontology keywords in page titles and snippets ensures irrelevant links (e.g., ads, unrelated blogs) are discarded early. The scraper operates on URLs approved by the crawler. It uses HTML parsing libraries (e.g., BeautifulSoup) to extract the Article title, full text content, publication date, source name, and media links (if any).

For further technical and implementation details, please refer to the main manuscript titled "Machine learning based classification of online news data for disaster management" (https://ieeexplore.ieee.org/abstract/document/9342921).

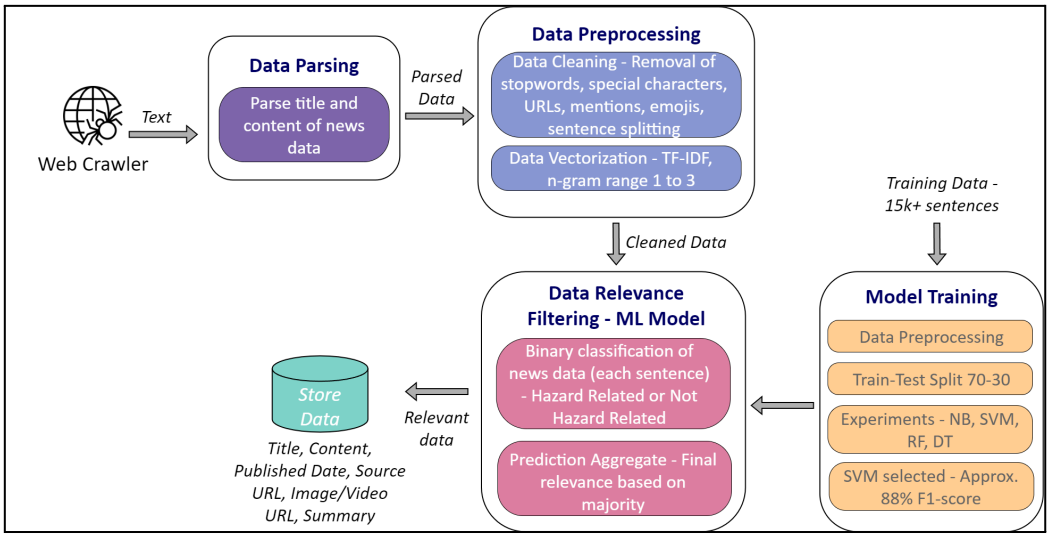

Figure 7: Overall Architecture of the Web Crawler and Scraper Module

3. IE System Algorithm

Algorithm 1 (Section III-D)

Figure 8 presents the flowchart of the ontology-driven actor-action-value extraction module used in the IE system. This component processes each sentence in a disaster-related news

report to identify triples that denote the actor (affected entity), action (impact), and associated value.

The process begins with keyword detection, where terms from the sentence are checked against the ontology (O) for matches with predefined action or actor keywords. If no exact match is found, the system dynamically updates the ontology with semantically similar terms.

Two types of associations are considered:

- Direct Associations: The head and child nodes of the numeric value are inspected. If these nodes contain actor or action terms from the ontology, a base score of 12 is assigned. A word similarity score and an LCS (Lowest Common Subsumer) boost are then added to compute the final score.
- Indirect Associations: If direct links are absent, the system applies Breadth-First Search (BFS) starting from the numeric value node. During traversal, proximity penalties are applied to the base score (set to 10), with additional contributions from word similarity and LCS boost.

All possible (actor, action, value) combinations are computed and stored with their respective scores. The final output is the combination with the highest score, which is stored as a structured triple in the database.

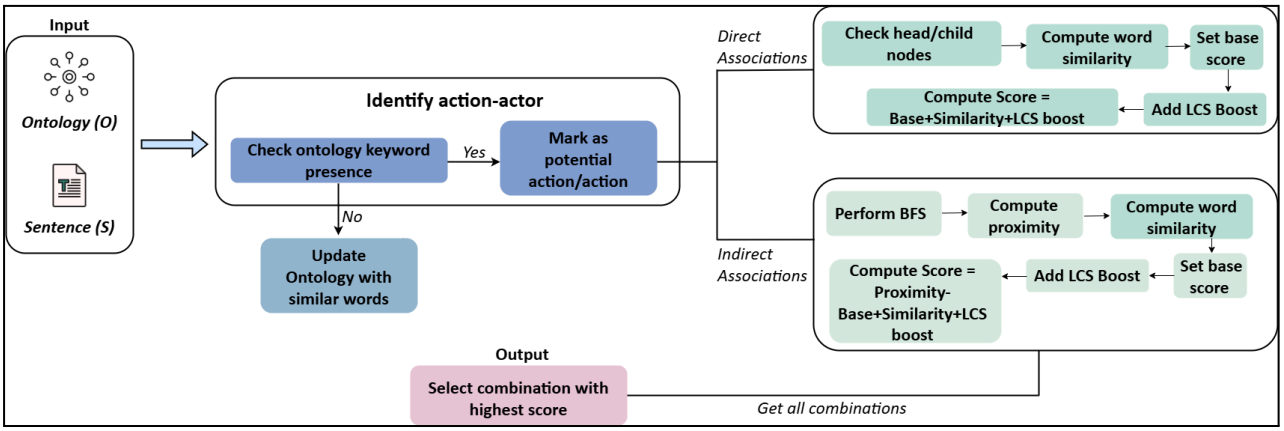

Figure 8: Flow diagram of Algorithm-1

Algorithm 2 (Section III-D)

The process begins by parsing each input sentence and checking for the presence of ontology terms. If a match is found, the word is marked as a potential resource or actor. Otherwise, semantically similar words are evaluated and optionally added to the ontology.

Two linguistic structures are evaluated to establish a valid actor-resource association:

- Preposition-Based Association: The algorithm checks whether the actor and resource are connected via prepositional constructs (e.g., “need of,” “require”). It also verifies the presence of requirement-indicative terms between the two. If both conditions are satisfied, proximity between the entities is computed.

- Subject-Verb-Object (SVO) Pattern: The algorithm checks if the sentence structure aligns with an SVO format (e.g., “People need water”) and then calculates the proximity score.

All valid combinations are generated, and the one with the highest proximity score is selected as the final association. This structured output is stored in the database and used to assess and visualize resource needs through the IE system dashboard. Figure 9 illustrates the flowchart of the algorithm.

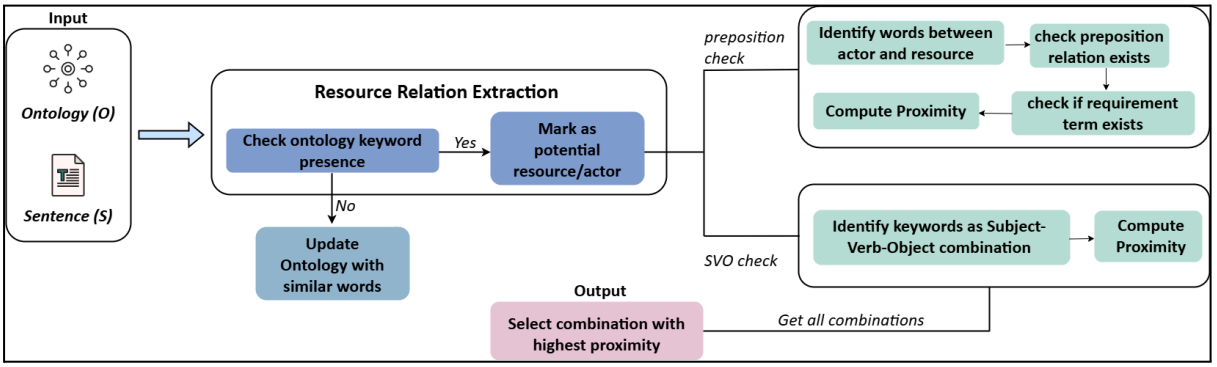

Figure 9: The flowchart of Algorithm 2.

4. Comparative Evaluation with Pre-trained Models

To validate the performance and explainability of the proposed ontology-driven Information Extraction (IE) system, we conducted a comparative analysis against five widely used NLP frameworks: spaCy, HuggingFace Transformers, Flair, AllenNLP, and Stanford Stanza. Each of these frameworks has been previously employed in generic or domain-specific IE tasks, especially for named entity recognition (NER), dependency parsing, and semantic role labeling (SRL).

The comparison was made across key dimensions critical for disaster-related information extraction, including domain adaptability, relation type extraction, explainability, customization flexibility, and triple generation capability. The ontology-based system, unlike the generic models, offers fine-grained control over relation types (causal, spatial, temporal, quantity, response, and resource) and supports transparent reasoning through explicitly defined object and data properties.

Moreover, the system integrates syntactic patterns like Part-of-Speech (PoS) tags and dependency relations as annotation properties, enabling context-aware extraction. In contrast, pre-trained models often function as black-box systems, requiring large annotated datasets and extensive fine-tuning to adapt to disaster-specific tasks. The ontology system also allows dynamic vocabulary updates, enabling it to learn and integrate emerging disaster terms over time.

A detailed comparison is provided in Table 2, highlighting the proposed system’s strengths in comparison to other popular NLP models used for information extraction.

Criteria	Ontology-Driven IE System	spaCy	HuggingFace Transformers	Flair	AllenNLP	Stanford Stanza
Domain Knowledge	Uses a curated ontology	General-purpose	General-purpose	General-purpose	General-purpose	General-purpose
Explainability	High (rule-based, traceable)	Limited	Black-box	Black-box	Moderate	Moderate
Triple Generation	Directly supported	Need fine-tuning	Need fine-tuning	Need fine-tuning	Partial	Need fine-tuning
Data Requirement	Low (ontology-based)	Needs large data	Needs large data	Need fine-tuning	Needs large data	Needs large data

Table 2: Comparison of the proposed IE system with the existing models

5. Interactive Dashboard Overview

The developed dashboard provides a visual interface for exploring the results of the ontology-driven Information Extraction (IE) system. It currently presents summary statistics derived from news reports, including aggregated counts of disaster impact factors such as casualties, rescue efforts, and public needs. In addition to the main summary page, the dashboard includes dedicated sub-pages offering deeper insights into specific components. These include: detailed information about the Wayanad landslide event, an overview of the *Natural Hazards Insight Ontology*, a summary of the research methodology, and comprehensive metadata on the news reports collected. This multi-page layout allows users to navigate through various facets of the system, supporting transparency, validation, and informed decision-making.

The next phase of dashboard development aims to support dynamic querying, where users can input natural language questions such as "How many death events in Wayanad landslides 2025?" These queries will be programmatically translated into SPARQL queries, enabling real-time extraction of relevant information from the underlying ontology. The results will be displayed on the dashboard in an interpretable format, enhancing user interactivity and decision-making capabilities. Once this Phase II development is complete and tested, the dashboard will be made publicly accessible for trials and community feedback.

The following is the link to the present version of the dashboard developed in Google Looker Studio:
https://lookerstudio.google.com/reporting/3015dd47-e21d-4286-9893-1157c4f30d1a/page/p_y6u6yidkvd

6. Closing Remarks

This supplementary material provides detailed insights into the ontology structure, information extraction methodology, SPARQL querying, and dashboard integration presented in the main paper. It includes ontology specifications, implementation algorithms, example queries, and dynamic system components. These details are intended to enhance the reproducibility and clarity of the proposed approach.

7. Abbreviations Used

The following are the abbreviations used in the paper in the order of occurrence:

Abbreviation/Acronym	Description
IE	Information Extraction
UNDRR	United Nations Office for Disaster Risk Reduction
VGI	Volunteer-Generated Information
PoS	Part-of-Speech
NER	Named Entity Recognition
NLP	Natural Language Processing
NGO	Non-Government Organization
ML	Machine Learning
DL	Deep Learning
SRL	Semantic Role Labelling
NN	Noun
VB	Verb
JJ	Adjective
nsubj	Nominal Subject
amod	Adjectival Modifier
dobj	Direct Object
HTML	HyperText Markup Language
SVM	Support Vector Machine
NDRF	National Disaster Response Force

BFS	Breadth First Search
advcl	Adverbial Clause Modifier
cc	Coordinating Conjunction
conj	Conjunct
oprd	Object Predicate
CCONJ	Coordinating Conjunction
NNS	Plural Noun
PROPN	Proper Noun
NUM	Number Modifier
ADJ	Adjective
nummod	Numerical Modifier
LCS	Lowest Common Subsumer
ADP	Adposition
prep	Preposition
pobj	Object of Preposition
IoT	Internet of Things
SVO	Subject-Verb-Object
GPE	Geopolitical Entities
LOC	Locations
nsubjpass	Nominal Subject (passive)
GIS	Geographical Information Systems

***