NLP for Disaster Message Classification and Analysis

Leo Davidov | Bharathi Sekar | Chamudi Vidanagama

ABSTRACT

Project focuses on the development of an end-to-end NLP pipeline for analyzing disaster-related messages to support emergency response coordination.

The system integrates text preprocessing, multi-label text classification, named entity recognition (NER), topic modeling, and sentiment analysis. It identifies key entities, themes, and emotions within social media data to enhance situational awareness.

Our findings show that lightweight linear models perform reliably on short, noisy texts, while topic modeling (LDA, BERTopic) reveals coherent themes such as aid, logistics, weather, and infrastructure. The pipeline can deliver actionable insights for disaster management agencies, improving information flow and decision support during crises.

TOPIC INVESTIGATION

1. Multi-label text classification for disaster response
2. Traditional ML vs. Deep Learning vs. Transformer performance comparison
3. Named Entity Recognition for critical information extraction
4. Topic modeling for thematic analysis of disaster communications
5. Sentiment analysis across disaster categories
6. Information extraction for emergency response optimization

RELATED WORK

Twitter activity during natural hazards provides real-time, ground-level situational information, establishing the foundation of crisis informatics research [1].

AIDR [2] introduced a real-time system for the automatic classification of disaster-related tweets, using crowdsourced annotations and classical machine-learning models to identify categories such as requests, damage reports, and aid activities. CrisisLex [3][4] extended this effort by developing large annotated corpora and a domain-specific lexicon of crisis-related language, enabling consistent tweet filtering and classifier training. The CrisisNLP initiative [3] later integrated these resources into a shared research platform supporting classification, information extraction, and summarization across multiple disaster events.

Together, these works established the methodological foundation for automated crisis-message analysis, combining annotated data, linguistic resources, and supervised learning to enhance disaster-response coordination.

[1] Social Media in Emergency Management - Survey of computational methods (Vieweg et al., 2010)
[2] AIDR (Artificial Intelligence for Disaster Response) - Automatic classification of disaster-related messages (Imran et al., 2014)
[3] CrisisNLP - Research on NLP for crisis response and monitoring (Olteanu et al., 2014)
[4] CrisisLex - A lexicon for collecting and filtering microblogged communications in crises (Olteanu et al., 2014).

TASK DISTRIBUTION

- All the tasks were distributed evenly
- Everyone worked on all the coding tasks
- Everyone worked on creating slides

Data Source

Disaster Response Messages Dataset (Kaggle)

https://www.kaggle.com/datasets/sidharth178/disaster-response-messages

Provider: Figure Eight

Public, clean, and well-documented real disaster data which supports building and testing response systems.

- ~26,000 messages from real-world disasters (e.g., earthquakes, floods, storms)
- Each message is tagged under multiple categories (e.g., *aid_related*, *infrastructure*, *weather*).
- Fields includes: ID, message text, genre, and category labels.
- Messages translated to English and anonymized.

TECHNOLOGIES

Programming Languages: Python

Core Libraries: NumPy, Pandas

ML Frameworks: Scikit-learn, TensorFlow, PyTorch, Transformers

NLP Libraries: NLTK, spaCy, Gensim, TextBlob, VADER

Visualization: Matplotlib, Seaborn, PyLDAvis, BERTopic, plotly

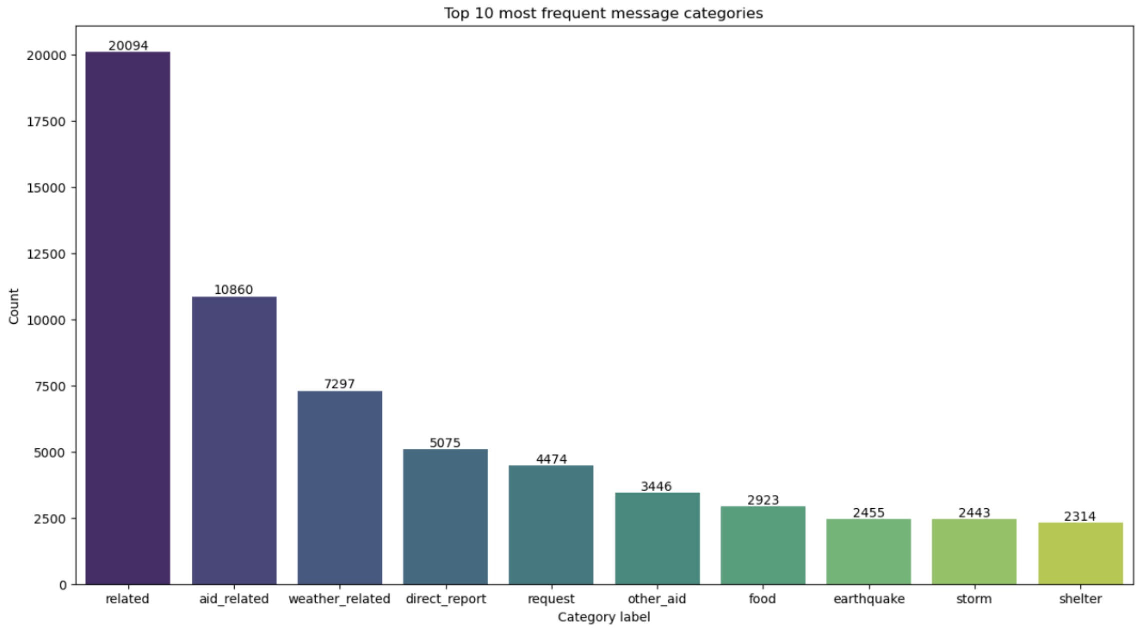
Development: Jupyter Notebook, Joblib for serialization

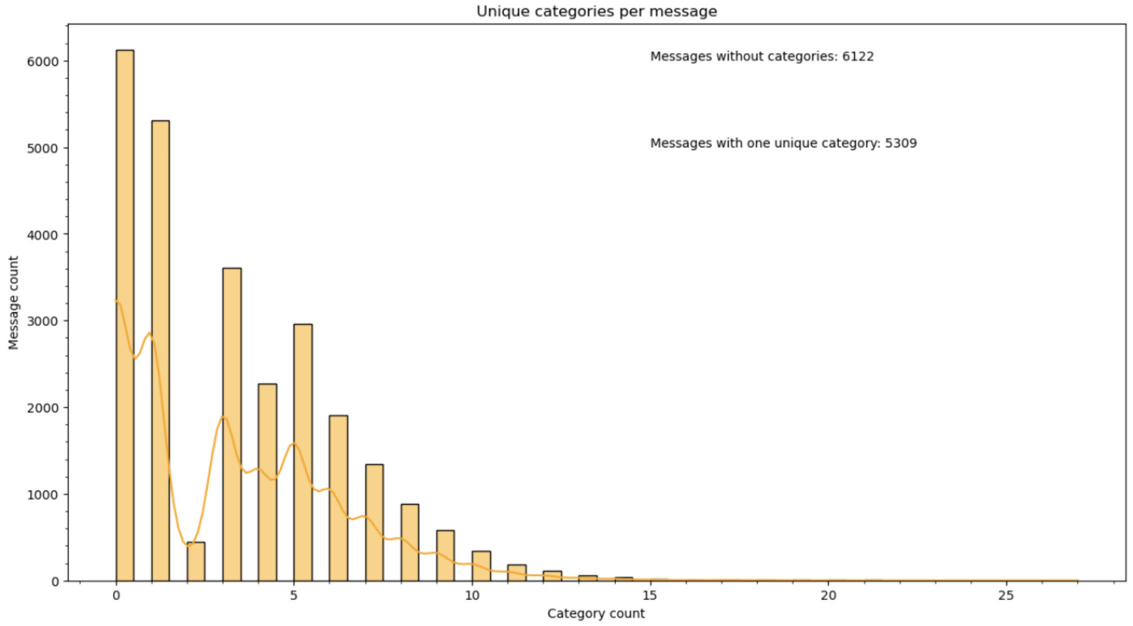
IMPLEMENTATION

Data Loading & Exploratory Data Analysis

To load, inspect, and understand the dataset's structure, quality, and distribution of the disaster message dataset.

- Dataset Loaded & Cleaned: Merged messages and categories, removed duplicates, handled missing original text, and corrected invalid labels (e.g., related=2 mapped to 1).
- Data Structure: Final dataset contains 26,216 messages, each classifiable into one or more of 35 categories (after removing the constant child_alone column).
- Severe Class Imbalance: The bar plot shows a highly skewed distribution. Top categories (related, aid_related) dominate, while critical classes (search_and_rescue, fire) are rare.
- Multi-label Analysis: The histogram reveals the multi-label nature of the problem, with many messages assigned to 0 or 1 category and a significant number of "vaguely assigned" messages (marked only as related).





IMPLEMENTATION

Text Cleaning and Preprocessing

To transform raw message text into a cleaned, tokenized, and lemmatized format suitable for NLP modelling.

Steps:

- Lowercasing: To convert all text to lowercase for consistency.
- Garbage Removal: To use a single regex pattern to remove URLs, hashtags, user mentions, numbers, punctuation, and underscores.
- Whitespace Handling: To collapse multiple spaces and trim trailing/leading whitespace.
- Tokenization: Split text into individual words (tokens).
- Stopword Removal & Lemmatization: Filtered out common English stopwords and reduce words to their base dictionary form (e.g., "updating" → "update").

Sample Output:

- Original: "Weather update - a cold front from Cuba..."
- Cleaned: ['weather', 'update', 'cold', 'front', 'cuba', ...]

	id	message	genre	related	request	offer	aid_related	medical_help	medical_products	search_and_rescue	...	other_in	clean_text
0	2	Weather update - a cold front from Cuba that c...	direct	1	0	0	0	0	0	0	...		[weather, update, cold, front, cuba, could, pa...
1	7	Is the Hurricane over or is it not over	direct	1	0	0	1	0	0	0	...		[hurricane]
2	8	Looking for someone but no name	direct	1	0	0	0	0	0	0	...		[looking, someone, name]
3	9	UN reports Leogane 80-90 destroyed. Only Hospi...	direct	1	1	0	1	0	1	0	...		[un, report, leogane, destroyed, hospital, st,...
4	12	says: west side of Haiti, rest of the country ...	direct	1	0	0	0	0	0	0	...		[say, west, side, haiti, rest, country, today,...

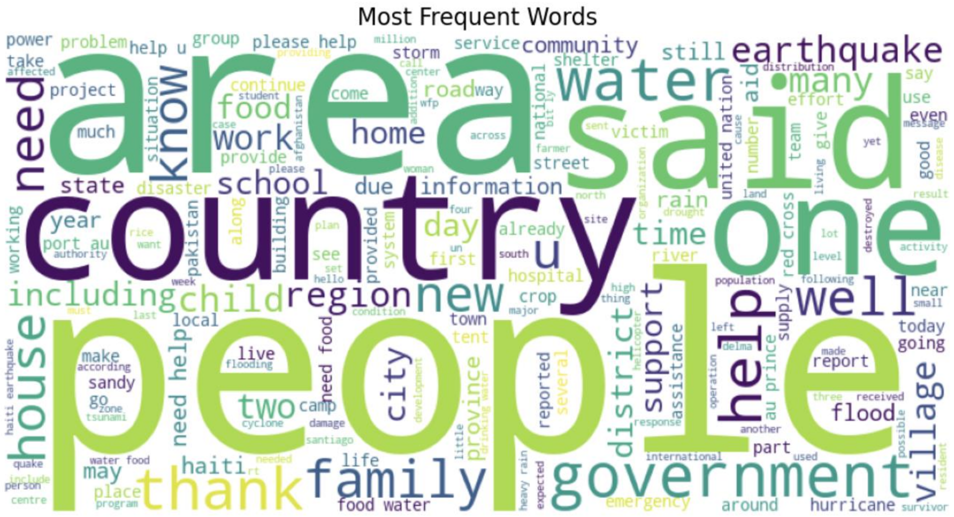
IMPLEMENTATION

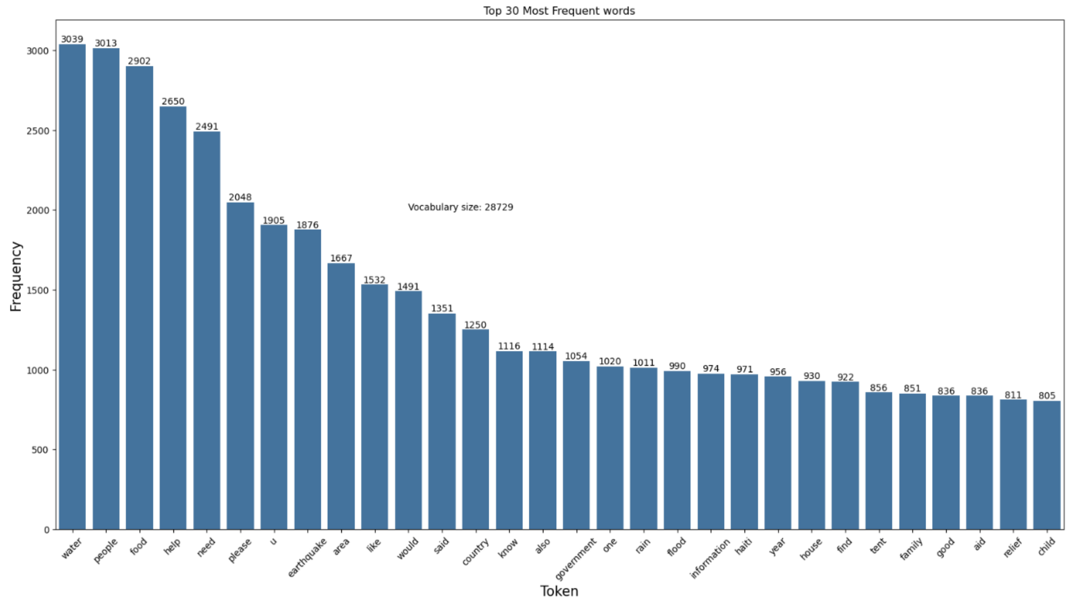
Text Representation Using Classical Methods

To convert cleaned text into numerical features for machine learning models using various vectorization techniques.

Methods & Results:

- **Word Cloud & Frequency Analysis:** Visualized the most frequent words (e.g., 'water', 'people', 'food', 'help'), confirming the disaster context.





IMPLEMENTATION

Text Representation Using Classical Methods

– **Bag-of-Words (BOW) & TF-IDF:** Created sparse vector representations. Limited vocabulary to 6,224 features by filtering rare and overly common words.

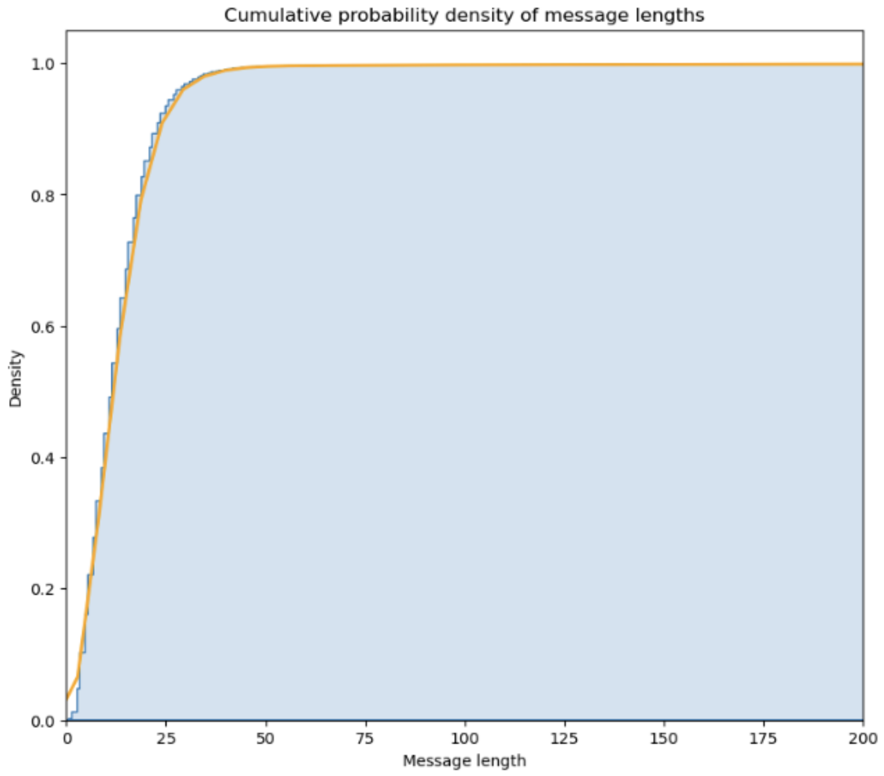
```
Bow train sample
  (0, 350)      1
  (0, 5253)     1
  (0, 2756)     1
  (0, 4144)     1
  (0, 3461)     1
  (0, 308)      1
Bow test sample
  (0, 1707)     1
  (0, 3387)     1
  (0, 3829)     1
  (0, 4775)     1
  (0, 4825)     1
  (0, 5915)     1
Vocabulary size: 6224
Train shape: (20972, 6224)
Test shape (5244, 6224)
```

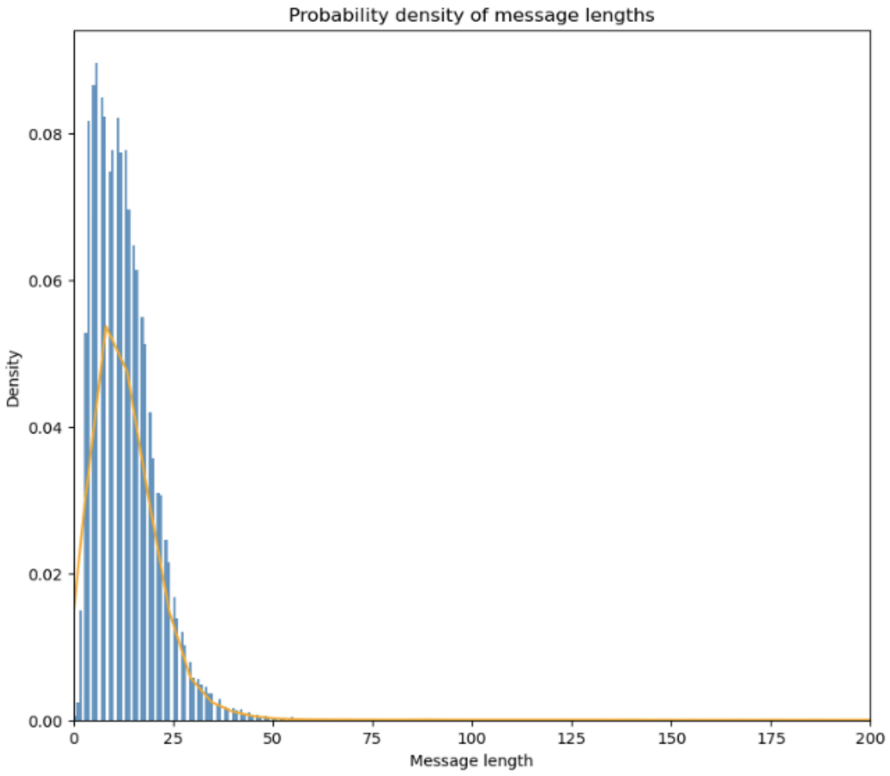
```
TF-IDF train sample
  (0, 350)      0.3956350220587632
  (0, 5253)     0.3206350006124153
  (0, 2756)     0.40639210211615395
  (0, 4144)     0.39606849218740015
  (0, 3461)     0.5941356728524809
  (0, 308)      0.25621111759832194
TF-IDF test sample
  (0, 1707)     0.2328380203498406
  (0, 3387)     0.36323175860598983
  (0, 3829)     0.27190922442845744
  (0, 4775)     0.4243354406964296
  (0, 4825)     0.5664106019216554
  (0, 5915)     0.488909995080891
Vocabulary size: 6224
Train shape: (20972, 6224)
Test shape (5244, 6224)
```

IMPLEMENTATION

Text Representation Using Classical Methods

– **Message Length Analysis:** Found that 99% of messages are shorter than 50 tokens, informing sequence padding for deep learning.

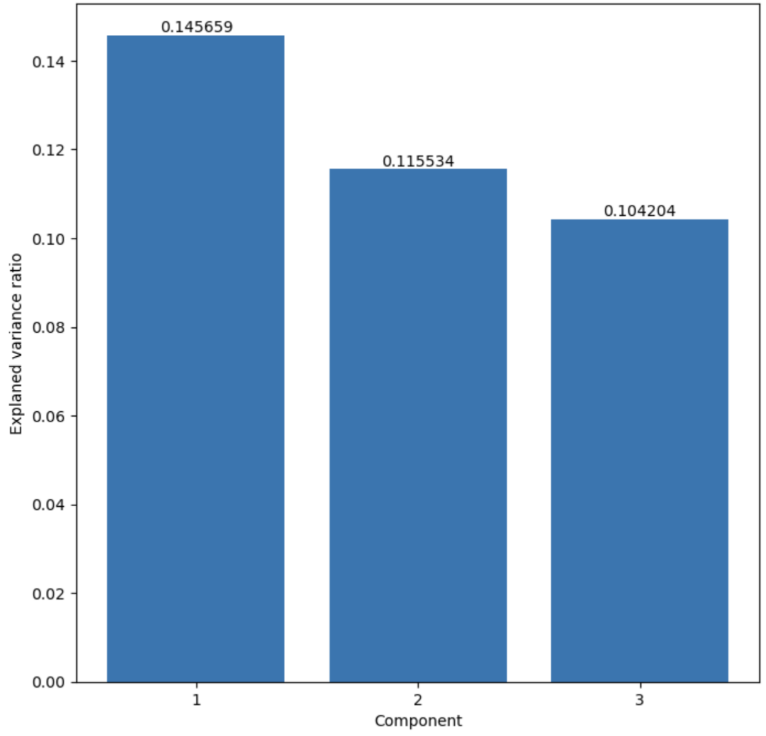




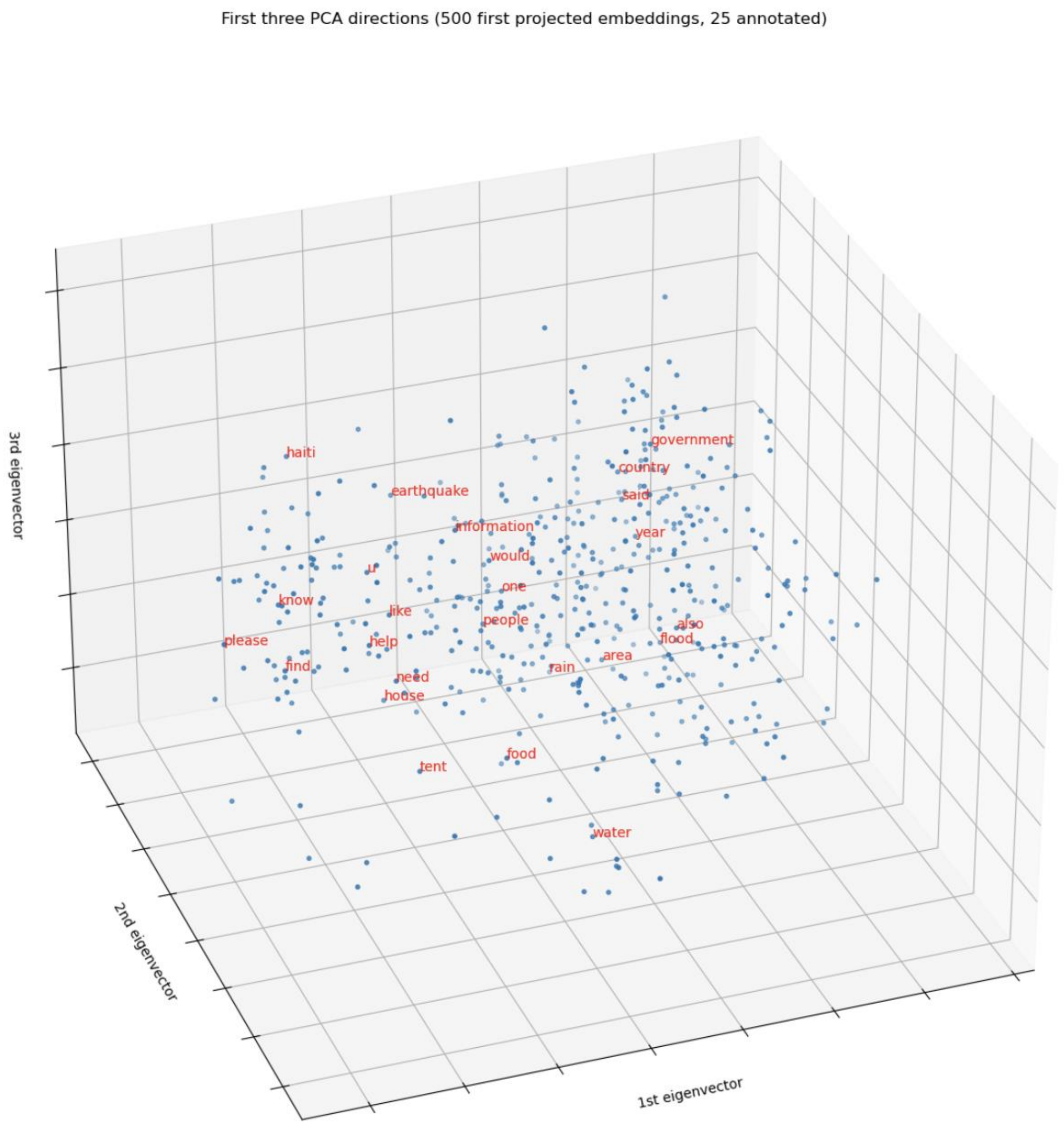
IMPLEMENTATION

Text Representation Using Classical Methods

- **Word2Vec Embeddings:** Trained a Skip-gram model to create 50-dimensional word vectors. Used PCA to project and visualize the semantic relationships between words in 3D space.



PCA decomposition of word embeddings, top 3 components

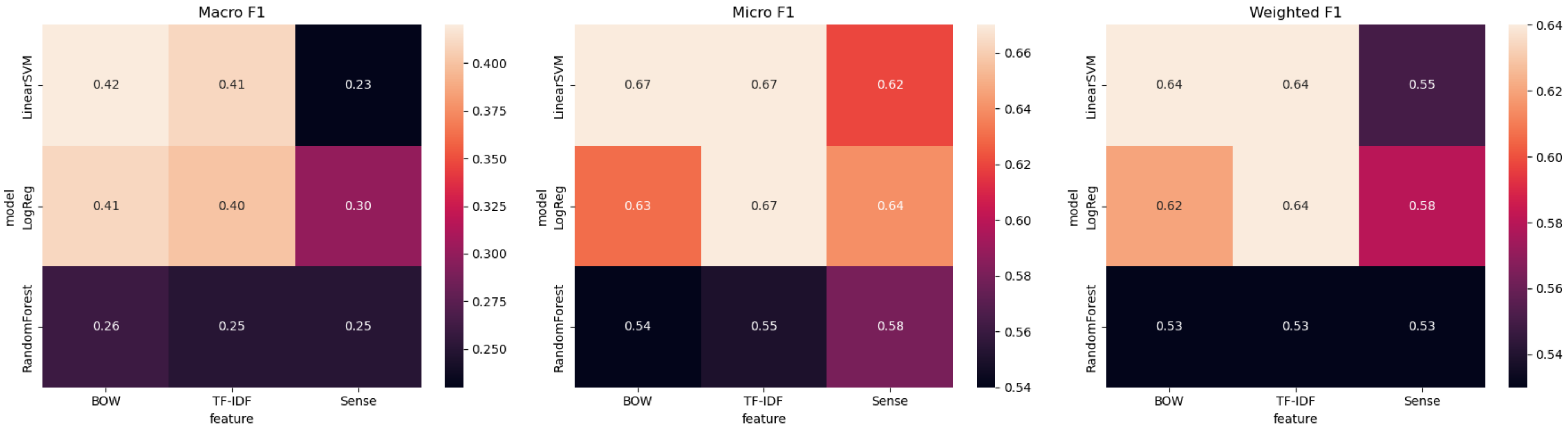


IMPLEMENTATION

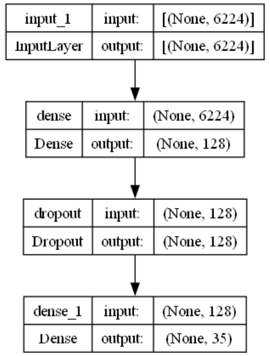
Multi-label Classification Using Traditional ML models (LinearSVM, Logistic Regression, Random Forest 80/20 strain-test split GridSearch)

Three feature sets tested, BOW Model, TF-IDF, and dense Word2Vec sentence embeddings

- Linear models (Logistic Regression, Linear SVM) – show good generalization across classes. Random Forest has difficulty handling high-dimensional sparse features by its nature.
- Linear classifiers are computationally efficient, having strong results on multi-label binary classification 35 of targets.
- TF-IDF features slightly outperformed raw BOW by emphasizing informative terms, while dense Word2Vec sentence embeddings - formed by summing and normalizing word vectors performed poorly, likely since they compress message structure and lose key token-level cues in short texts.



IMPLEMENTATION

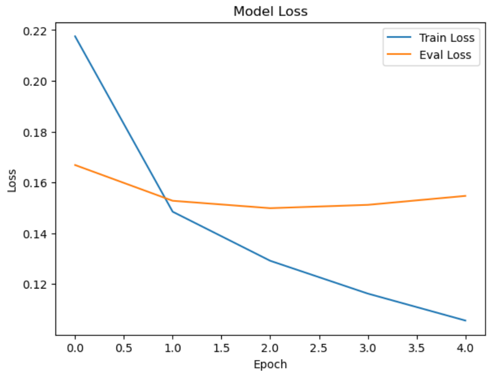


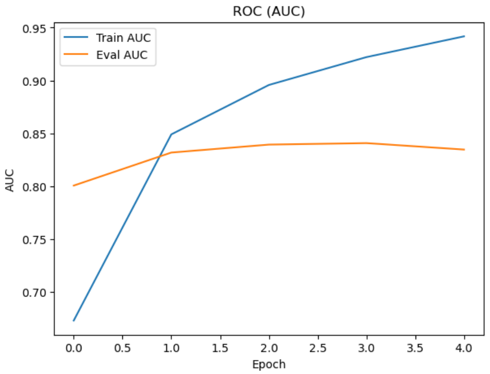
Multi-label classification using Deep Learning methods: Simple Dense

- Two feature sets tested, TF-IDF, and dense Word2Vec sentence embeddings
- Trainable params: 801,315
- Overfits extremely fast
- In general, worse than Classical ML, easily generalizes frequent categories, misses moderate

TF-IDF

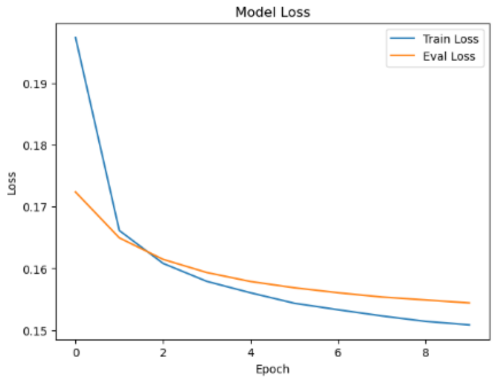
	precision	recall	f1-score	support
micro avg	0.78	0.59	0.67	16821
macro avg	0.70	0.31	0.38	16821
weighted avg	0.75	0.59	0.63	16821
samples avg	0.75	0.66	0.58	16821

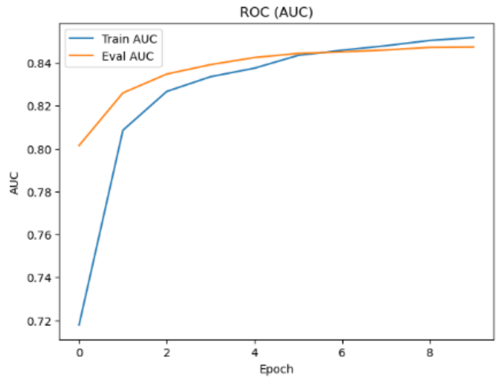




W2V Embeddings

	precision	recall	f1-score	support
micro avg	0.77	0.57	0.65	16821
macro avg	0.71	0.25	0.30	16821
weighted avg	0.75	0.57	0.59	16821
samples avg	0.73	0.66	0.57	16821

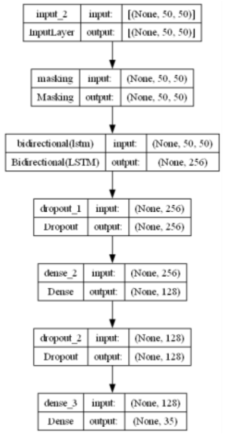




IMPLEMENTATION

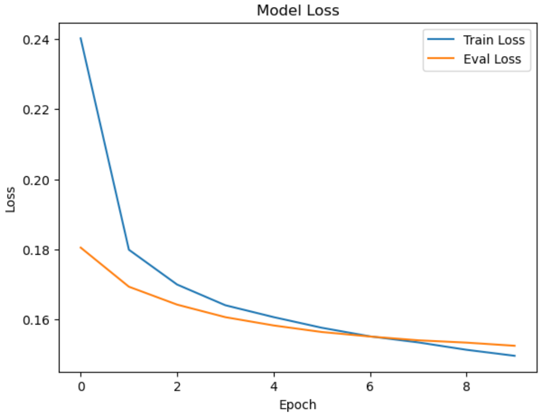
Multi-label classification using Deep Learning methods: Simple biLSTM

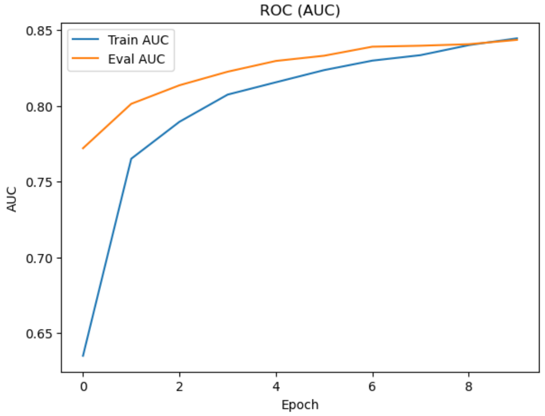
- Two feature sets tested: word2vec embeddings, trainable embedding layer
- Trainable params: 666,147
- Overfits fast
- In general, worse than Classical ML or Simple Dense Model, easily generalizes frequent categories, misses rare



W2V Embeddings

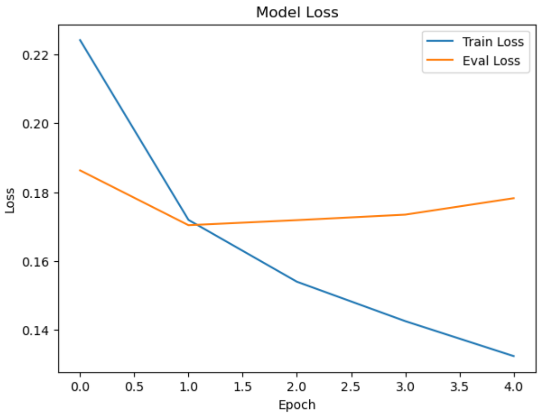
	precision	recall	f1-score	support
micro avg	0.78	0.57	0.66	16821
macro avg	0.67	0.25	0.30	16821
weighted avg	0.75	0.57	0.60	16821
samples avg	0.75	0.66	0.58	16821

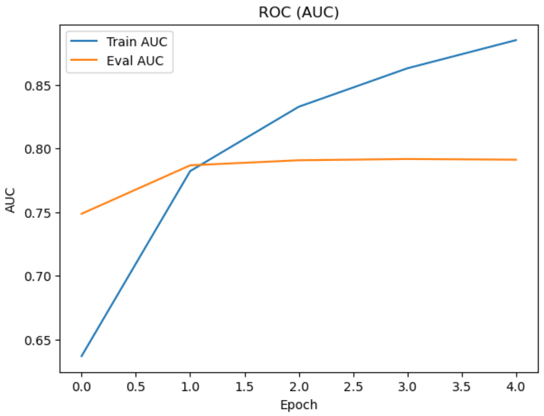




Trainable Embeddings

	precision	recall	f1-score	support
micro avg	0.76	0.56	0.64	16821
macro avg	0.66	0.22	0.24	16821
weighted avg	0.73	0.56	0.57	16821
samples avg	0.74	0.63	0.55	16821



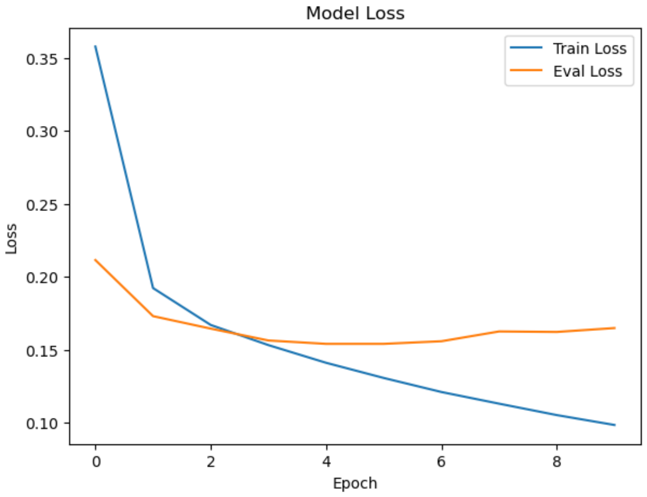


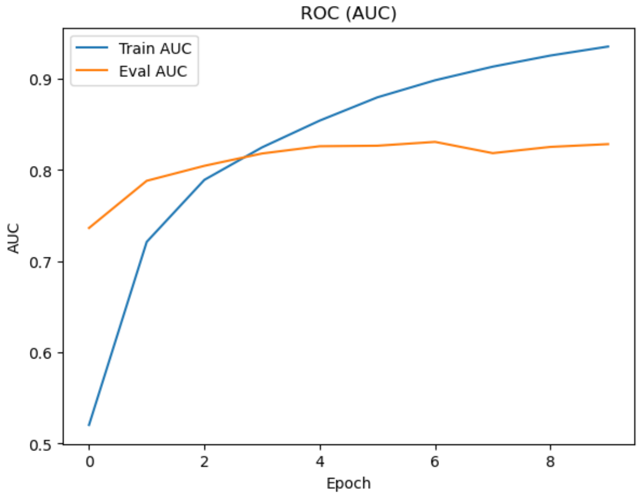
IMPLEMENTATION

Multi-label classification using Deep Learning methods: DistilBERT

- tokenizer = DistilBertTokenizerFast.from_pretrained('distilbert-base-uncased’) on clean texts
- 66 million parameters, overfits fast
- We used aggressive text cleaning. While this benefits classical models, deep models, and transformer models like DistilBERT are pretrained almost raw text and rely on punctuation, casing, and special tokens, which carry semantics BERT can use. This mismatch likely reduced useful signal (and sometimes shortened inputs to 1–2 tokens), which may have hurt transformer performance compared to using semi-raw documents.

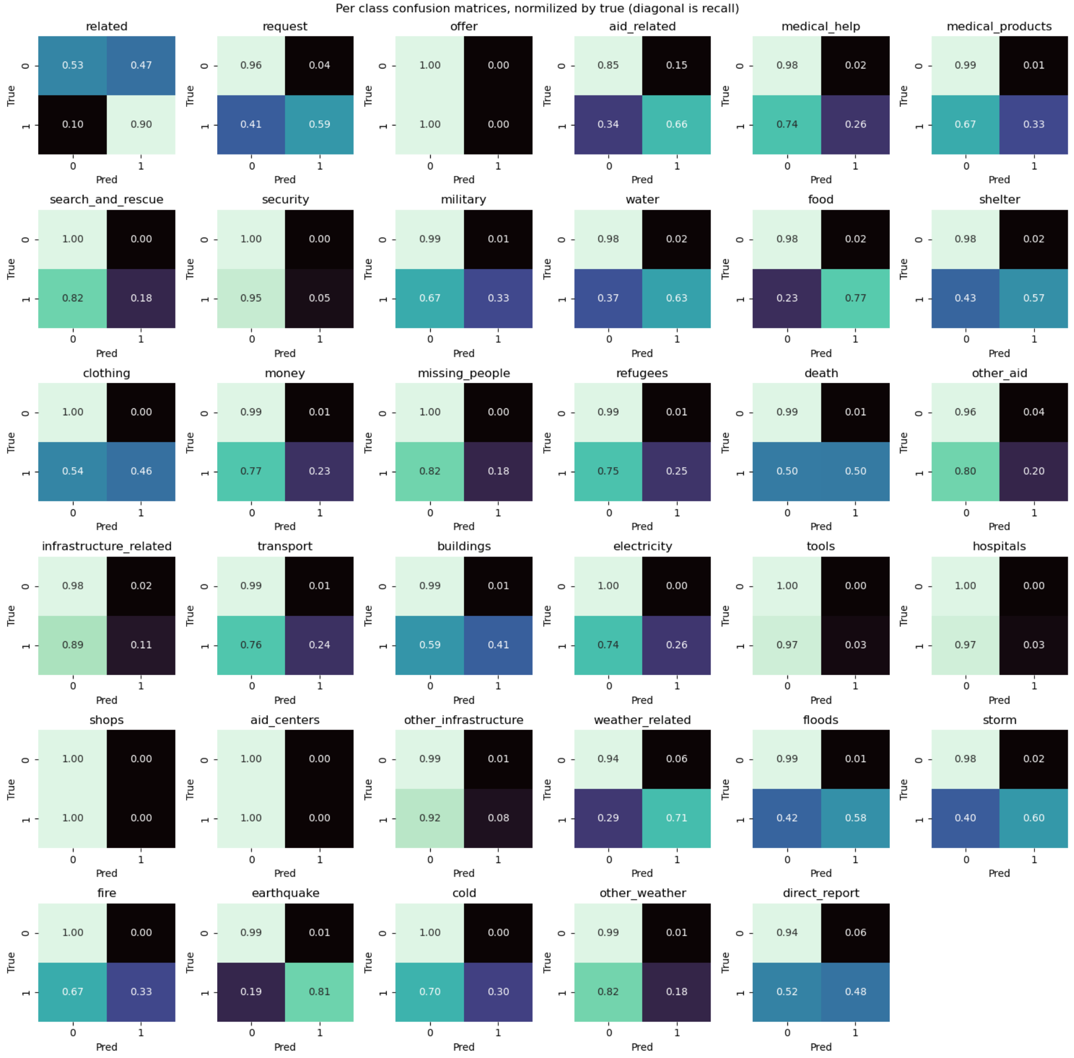
	precision	recall	f1-score	support
micro avg	0.77	0.63	0.69	16821
macro avg	0.66	0.29	0.32	16821
weighted avg	0.73	0.63	0.64	16821
samples avg	0.75	0.69	0.60	16821





IMPLEMENTATION

SVM BOW Confusion matrices



Multi-label classification using Deep Learning methods: Conclusion

- The linear dense network trained on TF-IDF features achieved the most balanced performance, with a macro-F1 score around 0.38 and a micro-F1 near 0.67, followed by DistilBERT.
- Models built on Word2Vec embeddings or trainable embedding layers underperformed, largely due to the dataset's short message lengths, small vocabulary, and class imbalance. The LSTM architecture didn't provide any advantage, as most messages contain few tokens (a lot 2-5 tokens or less after cleaning), offering little sequential information for the recurrent layers, same is applicable to BERT.
- Additionally, training word embeddings from scratch on such a small corpus leads to fast overfitting and weak generalization to rare labels.
- For short, noisy posts, classical linear models over BoW/TF-IDF remain a strong baseline.

IMPLEMENTATION

Named Entity Recognition (NER)

Used spaCy's pre-trained model to automatically extract and analyze named entities (like locations, organizations) from disaster messages to identify critical information for emergency responders.

Automatically extract information from messages to aid emergency response.

Applied spaCy's en_core_web_sm model to all messages.

Results

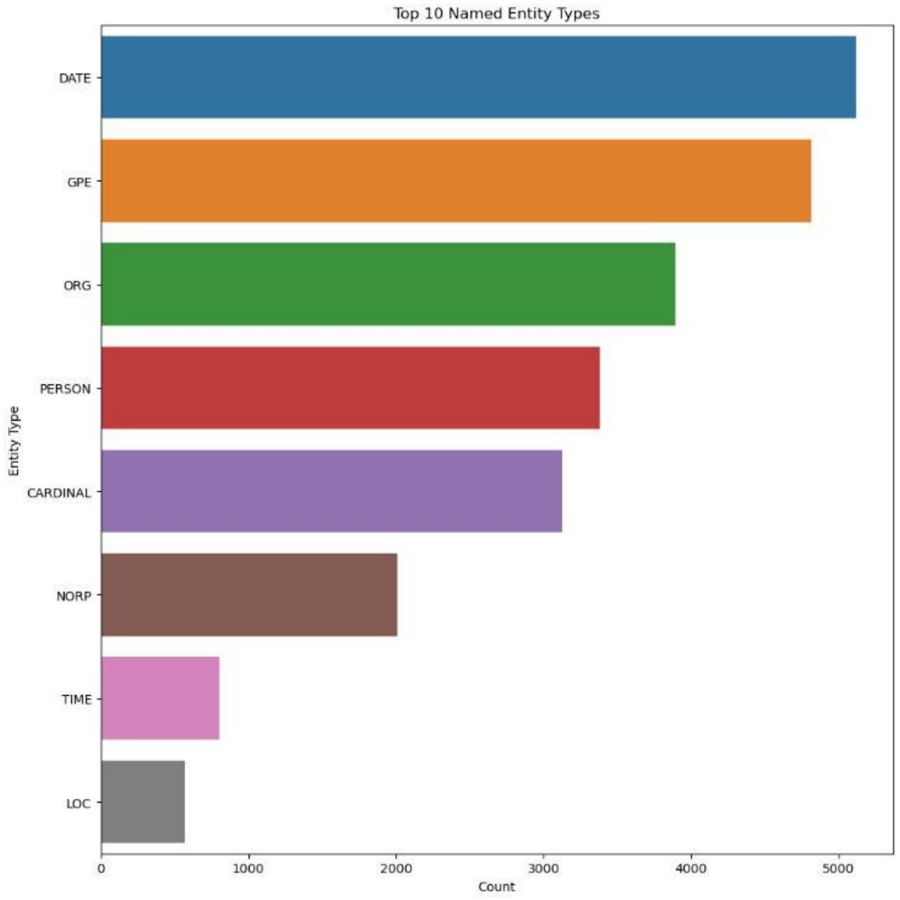
- Top 3 Entity Types: GPE (Locations), DATE, ORG.
- Sample Output: "west side haiti [GPE] ... tomorrow [DATE] ... radio nirvana [ORG]"

GPE: Direct routing signals for aid.

DATE: Prioritize urgent requests.

ORG: Identify key response agencies.

NER turns text into structured, actionable data for responders.



IMPLEMENTATION

Topic Modelling for Thematic Analysis

To uncover latent themes and discussion topics within the entire corpus of disaster messages to understand public concerns and response needs.

Approach & Results

- Applied two algorithms: **LDA** (classical) and **BERTopic** (modern, embedding-based).
- **LDA Topics:** Identified coherent themes like **"Basic Needs"** (water, food), **"Flooding,"** and **"Major Disasters"** (Haiti earthquake).
- **BERTopic Topics:** Produced more event-specific clusters (e.g., **"Hurricane Sandy," "Chile Earthquake"**) alongside thematic ones (e.g., **"Requests for Help"**).
- Generated interactive plots and bar charts showing the most important words for each topic.

Topic modeling effectively summarizes thousands of messages into actionable themes, offering valuable insights for improving disaster response planning and resource allocation.

IMPLEMENTATION

LDA Interpretation and Interactive Visualization

Brief interpretation

Topic 0 – Basic Needs: Water, food, power, hygiene, clothing: Essential supplies and infrastructure recovery.

Topic 1 – Drought & Agriculture: Rain, crop, drought, rainfall, province: Regional drought and agricultural impacts.

Topic 2 – Requests for Help: Help, need, water, tent, aid: Personal pleas for aid and local needs.

Topic 3 – Flooding: Rain, river, flood, road, wind: Floods and weather-related damage.

Topic 4 – Major Disasters: Earthquake, haiti, hurricane, sandy, tsunami: Earthquake and hurricane events

Topic 5 – Communication: Message, find, see, job, school: General communication and coordination.

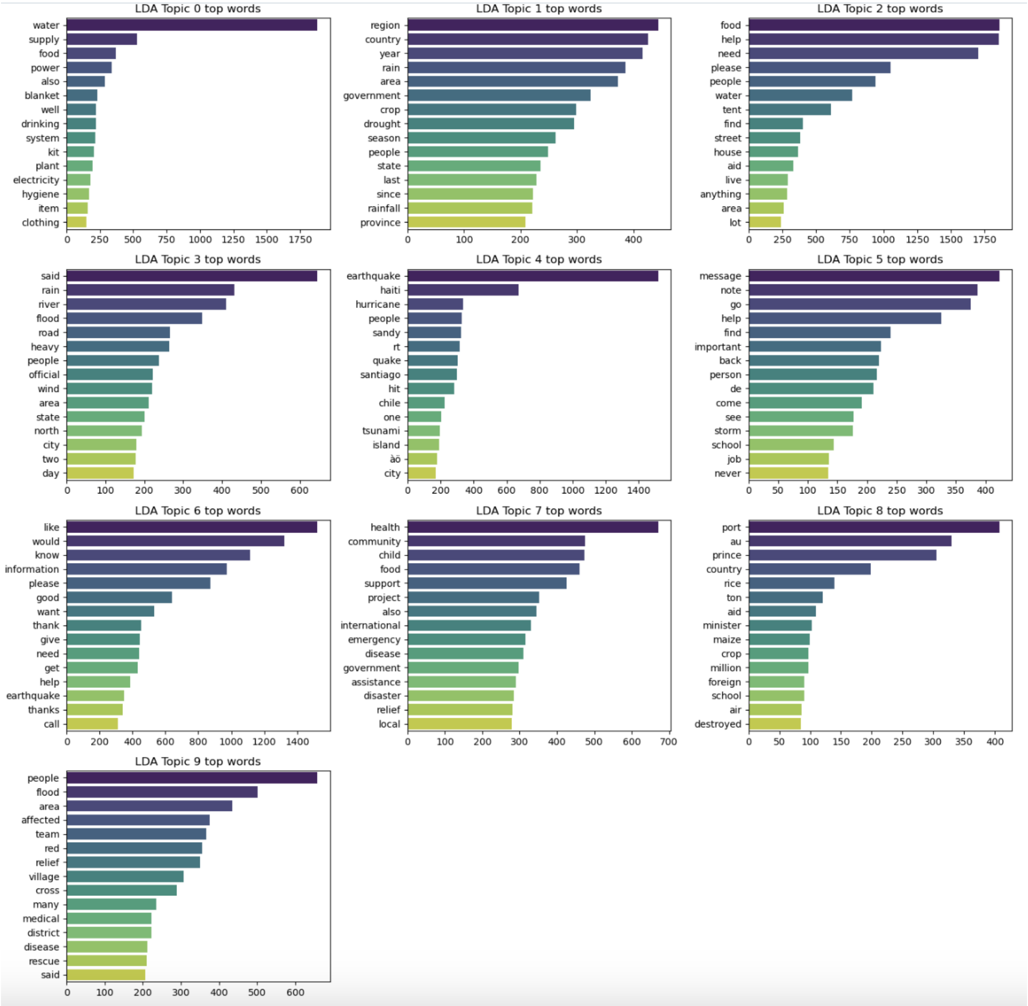
Topic 6 – Gratitude & Requests: Like, thank, want, help, call: Social interaction, thanks, and information seeking.

Topic 7 – Health & Aid: Health, community, project, disease, relief: Public health and humanitarian support.

Topic 8 – Haiti Reconstruction: Port, au, prince, rice, aid, school: Haiti-specific recovery and aid logistics.

Topic 9 – Flood Response: People, flood, red, cross, rescue, medical: Relief teams and rescue operations.

Overall: The topics are coherent and cover key disaster themes — needs, response, communication, and recovery across floods, droughts, and earthquakes, aligns well with marked category labels



IMPLEMENTATION

BERTopic Interpretation and Interactive Visualization

Brief summary

Topic 0 – Hurricane Sandy: sandy, hurricane, storm, power, nyc: Posts about Hurricane Sandy and related storm impacts.

Topic 1 – Communication & Information Requests: message, like, please, information, know: General messaging, info-seeking, and personal updates.

Topic 2 – Water & Health Issues: water, health, disease, food, child, medical: Public-health concerns and basic-needs discussion.

Topic 3 – Haiti Earthquake (General): haiti, earthquake, haitian, rt, passport: News and reactions around the Haiti earthquake.

Topic 4 – Requests for Help (Port-au-Prince): help, house, port, need, au: Direct pleas for assistance and local damage reports.

Topic 5 – Government & International Response: government, international, country, group, support: Official or NGO relief coordination.

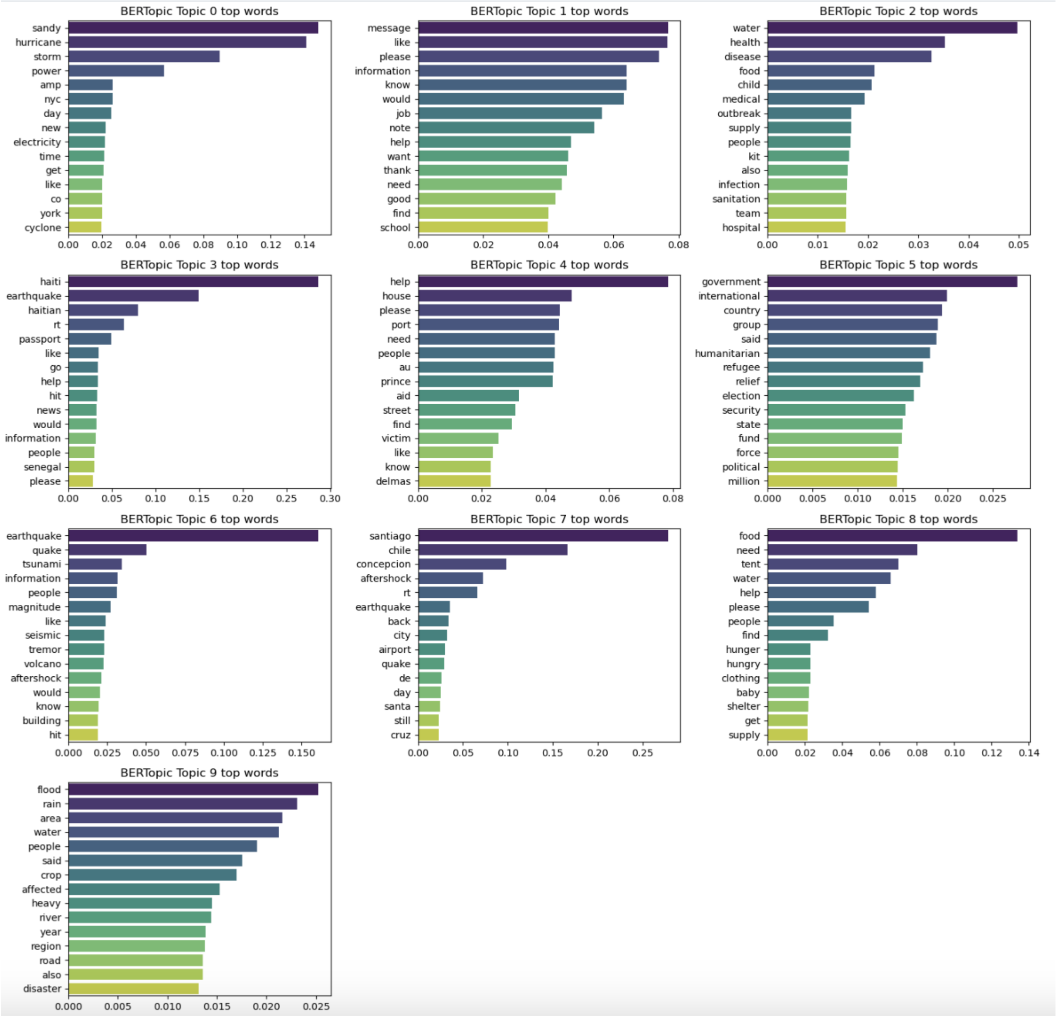
Topic 6 – Earthquake & Tsunami Information: earthquake, quake, tsunami, information, people: Event information and public alerts.

Topic 7 – Chile Earthquake Aftershocks: santiago, chile, concepcion, aftershock: Reports from Chile following the earthquake.

Topic 8 – Basic Aid Needs: food, need, tent, water, help: Requests for basic humanitarian supplies.

Topic 9 – Floods & Rainfall Damage: flood, rain, area, water, crop: Flooding, rain impact, and agricultural losses.

Overall summary: BERTopic cleanly separates specific disaster events (Hurricane Sandy, Haiti, Chile, floods) from response themes* (health, aid requests, government coordination). Topics are coherent and event-focused, with clear clusters for both **crisis types** and **relief activities**.



IMPLEMENTATION

Sentiment and Emotion Analysis

To analyze the emotional tone of disaster messages to understand public sentiment and how it varies across different disaster categories.

Approach & Results

Applied two sentiment analyzers: **VADER** (rule-based, sensitive to context) and **TextBlob** (lexicon-based).

The tools reveal different aspects of the data. **VADER** highlights the negative gravity of reports, while **TextBlob** reflects their factual, request-driven nature. This shows the complex emotional landscape of crisis communication.

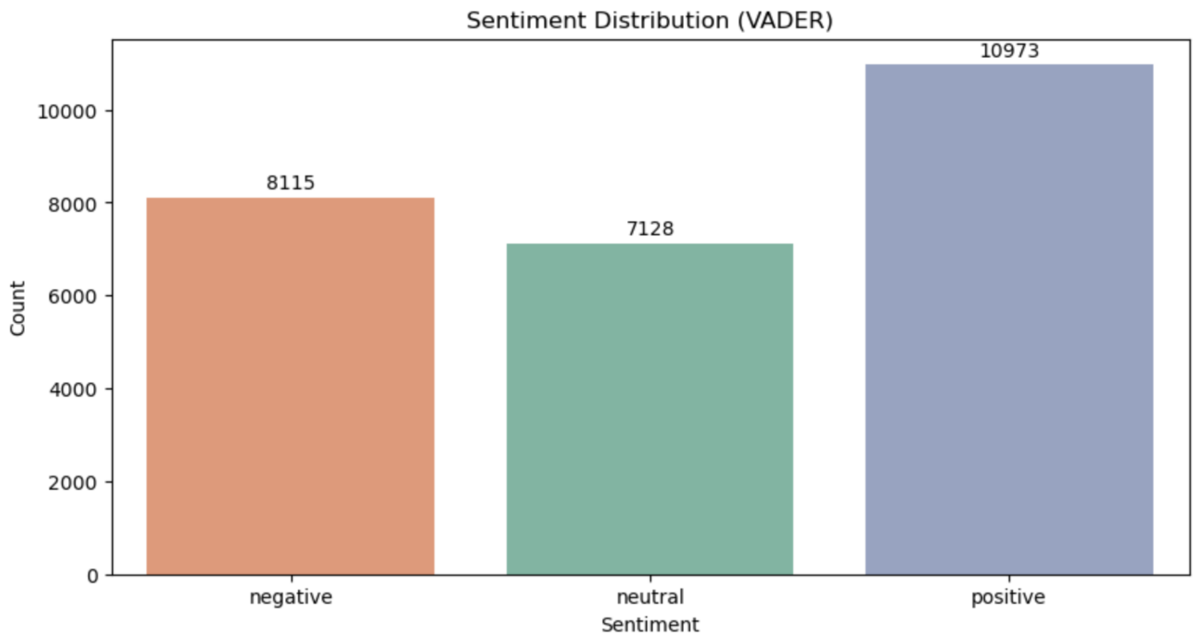
Sentiment by Category

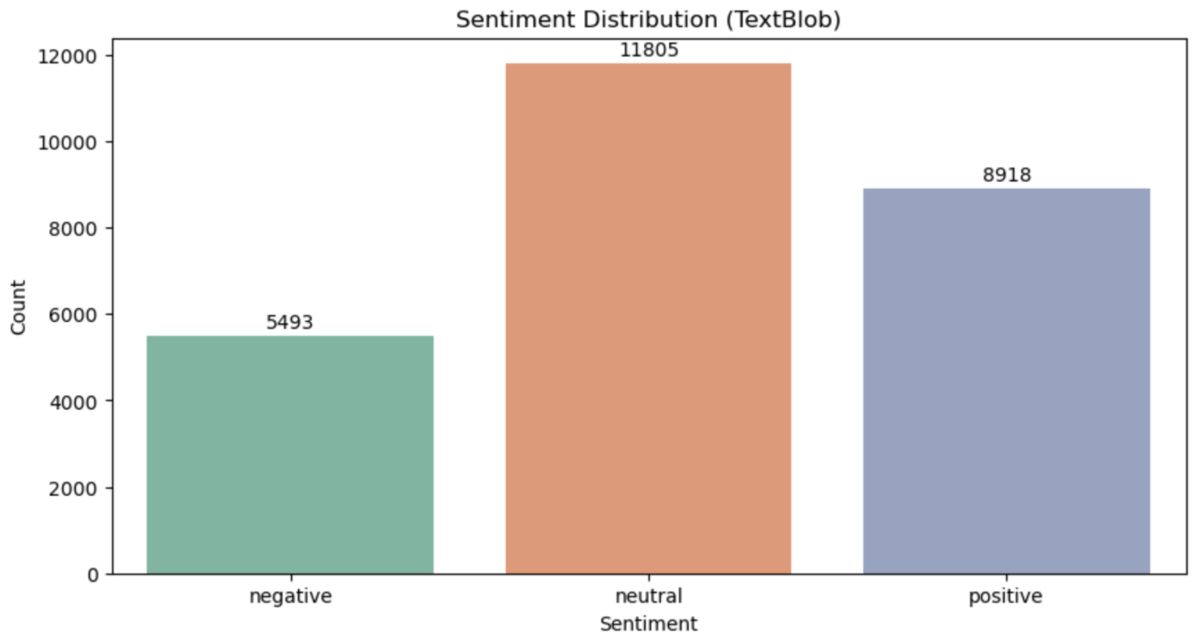
- **Most Negative Categories:** `death`, `military`, `buildings`, `fire` (associated with destruction and fatalities).
- **Most Positive Categories:** `offer`, `money`, `clothing`, `request` (associated with aid and generosity).

Sentiment analysis provides a crucial layer of understanding, showing that disaster communication is not uniformly negative but is strategically focused on needs and coordination.

IMPLEMENTATION

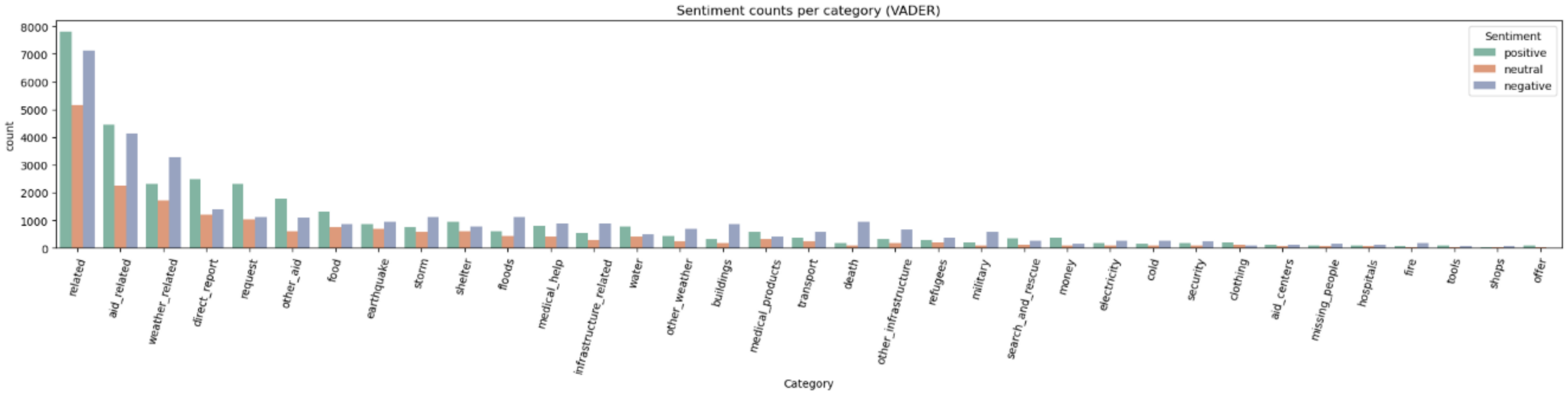
Sentiment and Emotion Analysis

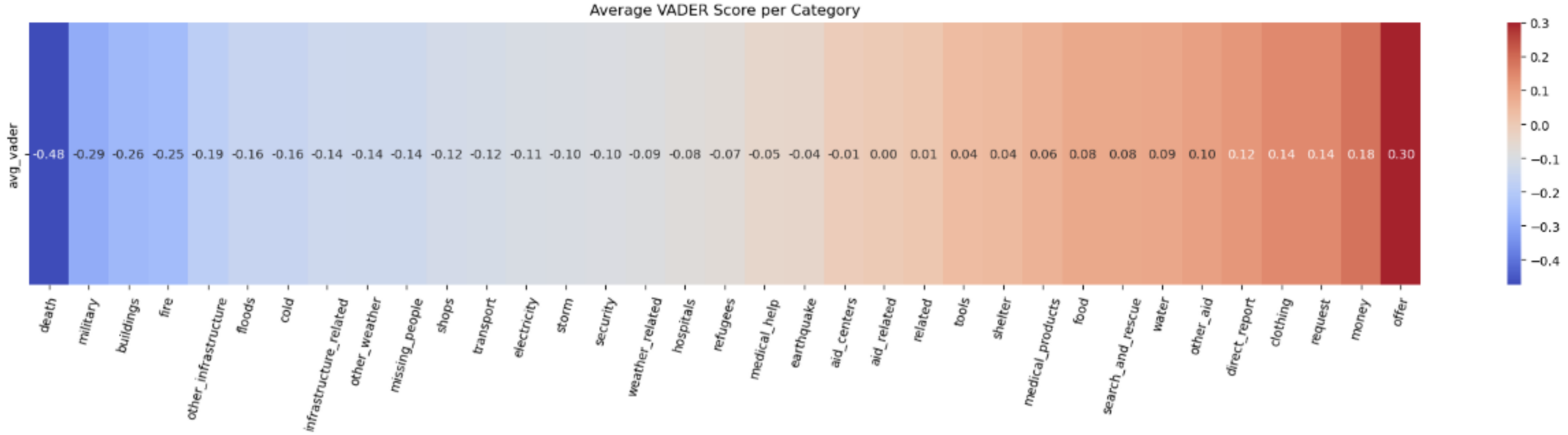




IMPLEMENTATION

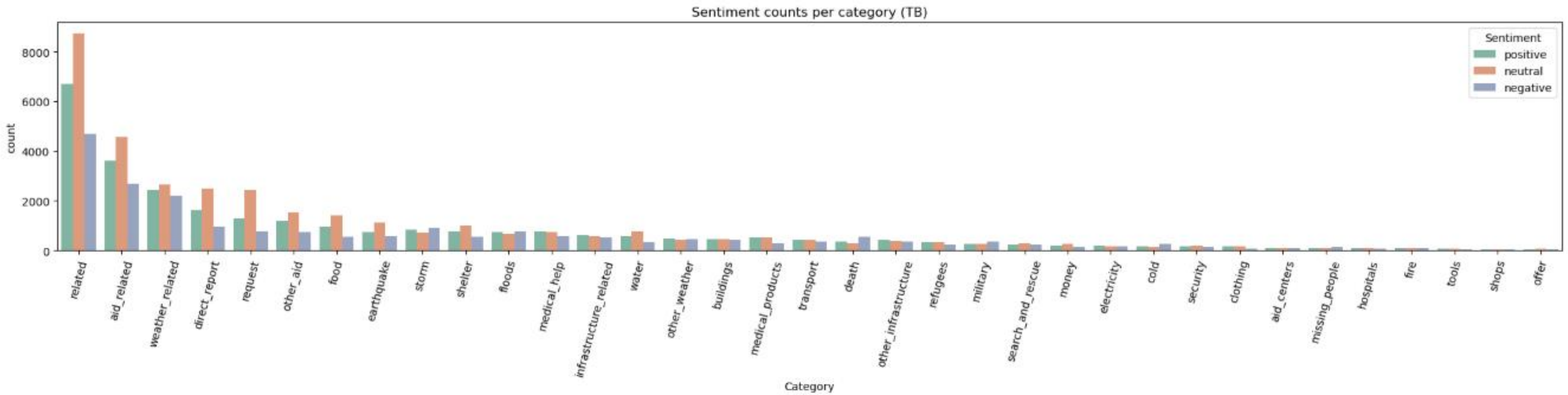
Sentiment and Emotion Analysis

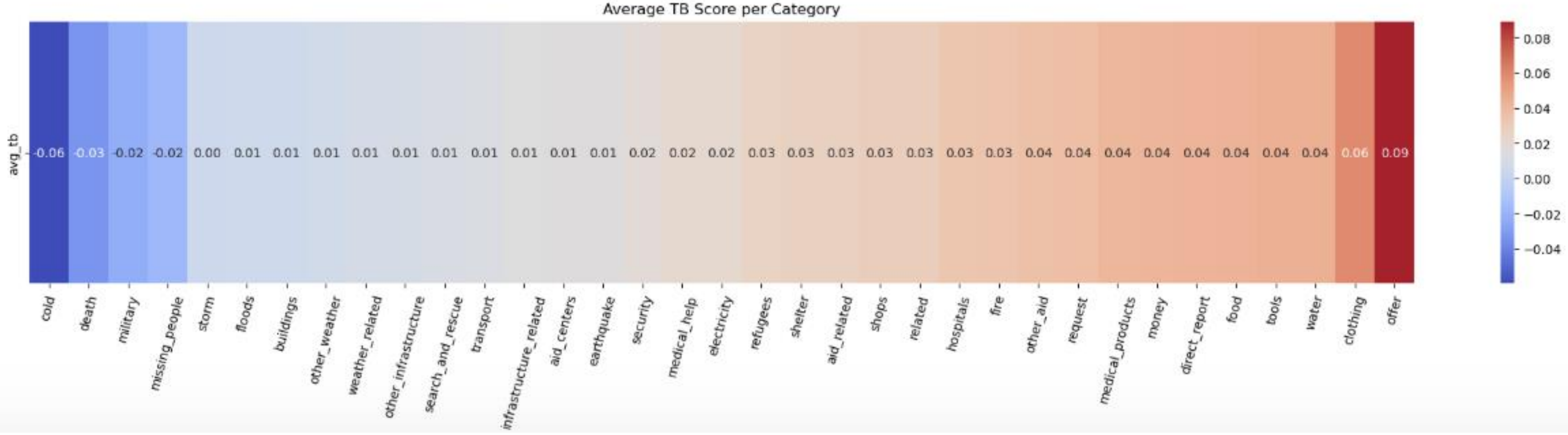




IMPLEMENTATION

Sentiment and Emotion Analysis





Future work

1. Potential future work can refine dense text representations of Word2Vec sentence embeddings by TF-IDF weighting and comparison of pretrained embedding models such as FastText, GloVe, or MiniLM for richer contextual semantics.

2. Better text cleaning methods

3. Further evaluation of n-gram approach in BOW and TF-IDF models.

4. Exploring models that combine sparse lexical and dense contextual features

UNIVERSITY
OF OULU