



NLP-Based Emergency Message Classification with Classification-Driven Triage

Zack Evans

Supervisor: Mark Huntly

Second Marker: Divya Chidambaram

Project submitted as part of the requirements for the award of
MSc Software Engineering and Artificial Intelligence

September 2026

DECLARATION OF ORIGINALITY

I, Zack Evans declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed : 
.....

Date : 2 Sept 2026
.....

FORM OF CONSENT

I Zack Evans hereby consent that my Project, submitted in candidature for the MSc Software Engineering and Artificial Intelligence, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed : 
.....

Date : 2 Sept 2026
.....

ABSTRACT

Natural disasters can disrupt conventional communication infrastructure while simultaneously increasing the need for rapid and reliable emergency information. This study investigates the use of Natural Language Processing (NLP) to classify disaster-related messages and support classification-driven priority assignment within a decentralised emergency communication architecture. A synthetic dataset containing 10,000 emergency messages across ten categories and 33 subcategories was developed, with additional language variation introduced through an open-question survey. Five NLP architectures were evaluated using a two-stage cascading classification approach: AWD-LSTM, BERT, DistilBERT, T5, and T5 (small). Performance was assessed using classification metrics alongside inference latency, throughput, and model storage requirements. Results showed that BERT achieved the highest subcategory accuracy of 99.93%, while DistilBERT achieved 99.80% accuracy with lower storage requirements and the best Orange Pi 5 Plus inference performance, reaching 13.52 messages per second with a mean latency of 73.94 ms per message. Based on the combined classification and deployment results, DistilBERT was identified as the most suitable candidate for integration within the proposed architecture. The study demonstrates the potential of lightweight transformer-based models for automated emergency message classification and classification-driven triage, while highlighting the need for further evaluation using independently collected real-world emergency messages.

TABLE OF CONTENTS

1. Introduction	6
1.1 Research Background	6
1.1.1 Disaster relief challenges	6
1.2 Research Problem Statement	7
1.3 Research Question	7
1.4 Research Aim	7
1.5 Research Objectives	7
1.6 Contribution of the study	8
2. Literature Review	9
2.1 Artificial Intelligence	9
2.2 Machine Learning	9
2.3 Deep Learning	10
2.3.1 Recurrent Neural Networks	10
2.3.2 Long Short-Term Memory Networks	11
2.4 Natural Language Processing	11
2.5 Transformers	12
2.6 Transformer-Based Language Models	12
2.6.1 BERT	12
2.6.2 DistilBERT	13
2.6.3 T5	13
2.7 Existing Emergency Triage and Prioritisation Approaches	13
2.8 Literature Review Summary	16
3. Research Methodology	17
3.1 Research Design	17
3.2 Open-Question Survey	19
3.3 Analysis Metrics	20
3.3.1 Classification Metrics	21
Multiclass Evaluation	22
3.3.2 Inference Latency and Throughput	22
4. Architecture Design	23
4.1 Decentralised Network	23
4.2 Emergency Message Classification Module	25
5. Experiments and Results Analysis	27
5.1 Software and Setup	27
5.1.1 Synthetic Data Generation	28
5.1.2 Dataset Structure	28
5.1.3 Hyperparameter Configuration	29
5.2 Results and Analysis	30
5.2.1 Model Performance	30
5.2.2 Category-Level Performance	31
5.2.3 Subcategory-Level Performance	32
5.2.4 Error Propagation in the Cascade	33
5.2.5 Model Complexity and Deployment Trade-offs	34

5.2.6 Inference Benchmarking	35
5.2.7 Class-Specific Error Analysis	35
5.2.8 Visual Summary of Experimental Results	36
Model performance comparison	36
Hierarchical error propagation	37
Accuracy storage trade-off	37
Class level performance heatmap	38
Dataset class distribution	39
CPU Inference Benchmark	40
5.3 Discussion	41
6. Conclusion, Limitations and Recommendations	43
6.1 Conclusion	43
6.2 Limitations	44
6.3 Recommendations and Future Work	45
7. Reflections	46
References	48
Appendices	53

1. Introduction

1.1 Research Background

Natural disasters cause damage and risk to all types of infrastructure [1]. One area that is particularly affected is the telecommunications sector [2], which is vital to emergency response and recovery operations [3]. During natural disasters, effective disaster relief becomes essential when it comes to saving lives. Robust infrastructure is therefore essential during disaster relief. To reinforce communication infrastructure, techniques have been developed to expand access for those in need [4]. Mesh networking and other technologies have been used in natural disasters to aid both people in need and relief teams [5][6]. Disaster relief operations are often constrained by limited resources, making effective prioritisation and resource allocation essential for managing available resources efficiently [7]. Proposed systems have incorporated technologies such as Artificial Intelligence (AI) in environments where reliable communication infrastructure is already available [8]. However, their effectiveness becomes less certain in disaster scenarios where existing infrastructure may be damaged, unavailable, or unable to support these systems.

1.1.1 Disaster relief challenges

Natural disasters create complex and uncertain environments, which present significant challenges when it comes to disaster relief [9]. The effectiveness of disaster relief response depends on the ability to rapidly assess damage, identify affected populations, establish communication, and coordinate the delivery of resources [9][10]. However, these tasks are frequently affected by limited information, damaged infrastructure, and constrained resources [9]. One of the biggest challenges is the loss of communication infrastructure [4]. Cellular networks, communication systems, and supporting power infrastructure may become unavailable, leaving affected regions without means of connectivity [4][5]. Without a reliable way to communicate, information sharing and coordination becomes significantly more difficult [9]. Establishing rapidly deployable and resilient communication infrastructure is therefore an important requirement for effective disaster response [4][5].

Another challenge is maintaining situational awareness. Relief teams may lack accurate information regarding affected populations, infrastructure damage, access routes, and available resources [9]. Delayed or unreliable information can reduce the quality and the speed at which decisions can be made [9]. Search and rescue operations also present significant challenges, particularly where affected individuals are located across large, dangerous, or inaccessible regions [11]. Traditional search methods may be limited by time, accessibility, and the safety of rescue personnel, encouraging the use of technologies capable of extending coverage and improving situational awareness [11]. These challenges have increased interest in technologies capable of supporting faster and more resilient disaster response [9][10]. AI can assist with tasks such as damage assessment, information classification, and decision support [8][10], while mesh networking can provide decentralised communication when conventional infrastructure is unavailable [5]. Together, these technologies have the potential to improve the availability of information, communication resilience, and coordination during disaster relief operations [4][5][10].

1.2 Research Problem Statement

Although current implementations of AI-based text classification have advanced considerably, many existing approaches depend on centralised systems [8][14][15]. This dependency can present a significant limitation in disaster environments, where communication infrastructure may be unavailable [2]-[4]. Previous research has therefore investigated decentralised and ad-hoc technologies for restoring connectivity in disaster affected areas [5][6][16].

At the same time, AI demonstrates its ability to process and classify disaster-related information to support emergency response and situational awareness [8][10][15]. However, the adoption of AI-based message classification with architectures designed to operate under infrastructure disruption remains limited [16]. Disaster scenarios can generate messages from both affected civilians and emergency personnel, requiring systems capable of distinguishing between different emergency categories, urgency levels, and operational priorities [8][9]. Therefore, this study focuses on developing an AI model that can categorise disaster-related messages and support priority assignment. The study will also design a low-cost decentralised communication architecture in which the classification model could potentially operate when conventional communication infrastructure is disrupted.

1.3 Research Question

Can an AI-based classification model effectively categorise disaster-related messages and support classification-driven priority assignment within a decentralised emergency communication architecture?

1.4 Research Aim

The aim of this study is to develop and evaluate candidate AI-based natural language processing systems capable of classifying disaster-related messages and supporting classification-driven priority assignment, and to assess their suitability for integration within a decentralised emergency communication architecture.

1.5 Research Objectives

To achieve the aim of this research, the following objectives were created:

1. To critically review the development and application of AI, machine learning and natural language processing, with particular focus on model architectures used for text classification and their suitability for disaster-related message classification.
2. To select, develop, and train candidate NLP classification models based on approaches identified in the literature review.
3. To evaluate and compare the performance of the selected candidate models using metrics including accuracy, precision, recall, F1-score, inference latency, and throughput, in order to identify the most suitable model for disaster-related message classification.

4. To assess the suitability of the selected classification model for integration within a decentralised emergency communication architecture intended for disaster scenarios where conventional infrastructure is disrupted.

1.6 Contribution of the study

This study contributes to the field of AI-assisted disaster relief management by investigating the use of automated message classification within environments that may be unreliable. The research develops, trains, and evaluates multiple candidate AI models for categorising messages from which predefined priority levels are derived from. This provides evidence on the suitability of different natural language processing based approaches for processing disaster-related messages.

A further contribution is the application of AI-based classification to disaster relief messaging, where incoming information may originate from both affected civilians and emergency personnel. By categorising messages according to emergency type and retrieving its priority from classification, the proposed approach aims to support faster and more consistent information processing and assist relief teams in identifying messages that require urgent attention.

Finally, the study contributes through the design of a decentralised emergency communication architecture capable of implementing the most suitable classification model. The proposed architecture explores how AI-based message processing could be integrated into a disaster relief system when conventional infrastructure is unavailable. This provides a conceptual framework for combining decentralised communication with AI-based classification in future disaster response systems.

The study therefore contributes by bringing together an AI-based emergency message classification and decentralised communication design, two areas that are often considered separately within disaster response research.

2. Literature Review

The existing literature and advancements in Artificial Intelligence (AI) for Natural Language Processing (NLP) and emergency message classification are the primary focus of this section. Existing approaches to text processing are examined through the development of Machine Learning (ML), Deep Learning (DL), and NLP-based methodologies. Particular attention is given to the progression from traditional rule-based and recurrent approaches towards transformer-based architectures, which have become widely adopted for NLP tasks. Additionally, this section examines the development and structure of transformer models and their advantages over Recurrent Neural Networks (RNNs) and Long Short-Term Memory Networks (LSTMs) when processing contextual and sequential information. Transformer-based models including BERT, DistilBERT, and T5 are subsequently explored. These approaches are examined to establish that they can become candidate models for the project. Finally, existing implementations of AI-based emergency triage and message classification are critically analysed to identify current capabilities, limitations, and opportunities for classifying and prioritising disaster messages within disaster relief.

2.1 Artificial Intelligence

AI is a field within computer science that focuses on creating systems capable of carrying out tasks that would normally require human intelligence. These tasks include reasoning, learning from data, perceiving and interpreting information, making decisions, and understanding human language [17]. The concept of AI comes from the idea of simulating human cognitive abilities. One of the earliest discussions surrounding machine intelligence was introduced through Turing's question of whether a machine could demonstrate behaviour that could be considered intelligent [18]. Early AI research was heavily influenced by symbolic AI, where knowledge representation and reasoning relied on explicit symbolic structures, logical rules, and formal inference mechanisms [19]. Within these systems, knowledge was generally encoded manually by given experts and predefined rules to produce an output [20]. However, symbolic approaches presented significant limitations because their performance depended heavily on such principles. These systems were often unable to effectively handle incomplete information or situations that extended beyond the existing knowledge base [21].

2.2 Machine Learning

The limitations of rule-based AI approaches contributed to the development of ML, a subset of AI in which computer systems improve their performance by self-learning from data rather than relying solely on predefined rules [22]. They improve their performance by analysing examples and identifying relationships within datasets resulting in systems that can make predictions or classifications when presented with previously seen or even unseen data [23]. ML approaches are generally divided into three principal learning subsets: supervised learning, unsupervised learning, and reinforcement learning [24]. Supervised learning trains a model using labelled examples in which input data is associated with known outputs. The model learns the relationship between these inputs and outputs so that classifications or predictions can be produced for previously unseen data. In contrast, unsupervised learning operates on unlabelled datasets and attempts to identify underlying structures, patterns, or

relationships within the available data. Reinforcement learning differs from both approaches by allowing a model to learn through interaction with an environment, where actions are evaluated through rewards or penalties [24]. For AI classification where predefined categories are available, supervised learning is a particularly relevant approach because the relationship between an input and its expected classification can be learned from labelled examples. Within text classification, this may involve assigning individual messages to predefined categories.

Conventional ML algorithms frequently require relevant characteristics, or features, to be identified and prepared before model training. For text based data, these features may represent properties such as word occurrence, frequency, or other statistical relationships within the text. While such approaches can perform effectively for classification problems, manually encoded representations may fail to capture more complex relationships within data [25].

2.3 Deep Learning

Deep Learning (DL) is a subset of ML which is based on multi-layered artificial neural networks [26][30]. Unlike many conventional ML approaches, DL is capable of automatically learning increasingly complex representations from input data using multiple processing layers [25]. This ability has contributed to significant improvements across areas including image recognition, speech processing, and natural language understanding [27][28][29]. Likewise, DL provides an important transition between traditional ML classification and the more advanced language processing architectures used within modern natural language systems. The ability to automatically extract representations has made DL particularly effective for processing vast amounts of complex data, including audio, images, and natural language [27]. Within language processing tasks, this is important because the meaning of text cannot always be determined from individual words. Instead, the interpretation of a word may depend on its position, surrounding words, and wider sentence context [31]. DL architectures allow for such models to exist by learning the direct relationships between words through large collections of text.

Since natural language consists of ordered sequences of words, natural language processing (NLP) systems commonly employ Recurrent Neural Networks (RNNs) [32] and/or Long Short-Term Memory Networks (LSTMs) [33]. These architectures attempted to preserve information obtained from previous elements of a sequence so that later words could be interpreted using preceding context [31]-[33].

2.3.1 Recurrent Neural Networks

RNNs are designed to process data in sequence, using information from earlier inputs to help understand inputs that come later on [31]. Unlike feedforward neural networks, RNNs maintain information from earlier inputs and use it when processing subsequent elements of a sequence [31]. Within NLP, this allows text to be processed sequentially, with information from previous words influencing later seen words [31][32]. This makes RNNs suitable for applications such as language modelling and sequential text processing [32]. However,

standard RNNs can struggle with long sequences because information from earlier inputs becomes increasingly difficult to preserve during training [34]. This can cause the vanishing gradient problem [34], making it difficult for the model to learn relationships between words that are far apart. To address this problem, LSTMs were developed to maintain information over much longer sequences using specialised memory cells and gating mechanisms that regulate the flow of information [33].

2.3.2 Long Short-Term Memory Networks

LSTMs are a type of RNN designed to better handle long-term relationships in sequential data [33]. They improve standard RNNs by using a memory system and gates that control what information is kept, updated, or forgotten [33]. This helps the model retain useful information over longer sequences and reduces some of the problems RNNs face with long-term dependencies [33][34]. Because of this, LSTMs have been applied to sequence-based tasks including speech recognition, handwriting recognition, and part-of-speech tagging [35]. Despite these improvements, LSTMs still process data in sequence, with each step depending on the one before it [35][36]. This limits how much processing can be done at the same time and makes long sequences require more computing power [36]. It can also make it harder to capture relationships between words that are far apart within a sequence [36].

2.4 Natural Language Processing

NLP combines concepts from computer science and linguistics to transform human generated language into representations that models can process. Human language presents a significant challenge for such systems because the meaning of text is influenced by context, sentence structure, word order, and relationships between words, rather than by individual terms alone [37]. Early models primarily relied on manually designed linguistic rules and statistical methods. This limited their ability to understand more complex text and contextual meaning [38]. With the development of distributed word representation methods such as Word2Vec and Global Vectors for Word Representation (GloVe), a significant advancement in NLP was established by converting words into numerical vector representations based on their relationships within data [39][40]. With this, models could identify similarities between words and start to capture semantic meaning within text. However, traditional word-to-vector techniques assigned a single, fixed vector representation to each word regardless of its context. The result was that words with multiple meanings were assigned a single context instead of assigning different contexts depending on the surrounding words, limiting the model's ability to capture contextual semantics [41]. This requirement resulted in the development of transformer-based models, such as Bidirectional Encoder Representations for Transformers (BERT)[42].

2.5 Transformers

The transformer model was first introduced by Vaswani et al. [36] in 2017 through the paper "Attention Is All You Need", which presented a new approach entirely on attention mechanisms. This development influenced NLP research by allowing models to capture

relationships between distant words more effectively than previous recurrent approaches, leading to improved performance across many sequence-related tasks [43][44]. Before the adoption of transformers, RNNs and LSTMs were primarily used for handling sequential data including text [45]. While they improved the way models understand relationships between words, they tend to struggle with longer sequences of data. Transformers overcame these challenges through the development of self-attention, which enables the model to assign relevance to different words of the given sequence and capture their relationships [36][46]. Unlike recurrent architectures, transformers do not rely on sequential processing. Instead, self-attention allows tokens within a sequence to be processed in parallel during training, improving training efficiency and enabling the model to capture relationships between words over longer distances [36]. This improvement has led to a performance increase within NLP tasks and established transformers as a widely adopted architecture for pretrained language models [47]. The original transformer architecture consists of an encoder and a decoder. The encoder transforms the input sequence into contextual representations, while the decoder uses these representations to produce an output sequence [36]. Later transformer models adapted parts of this architecture for different purposes. BERT, for example, uses the encoder component to learn bidirectional contextual representations and is primarily designed for language understanding tasks such as text classification [42].

For disaster message classification, contextual interpretation is important because emergency type may not be identifiable from an individual keywords alone. The meaning of a message can instead depend on the relationship between multiple words and the wider context in which they occur. Transformer-based language models therefore provide a relevant approach for investigating the automated classification and prioritisation of messages within disaster management scenarios.

2.6 Transformer-Based Language Models

The development of transformer architectures led to a range of pre-trained language models designed for different NLP tasks and computational requirements. These models are typically pre-trained on large text datasets and then adapted to tasks such as text classification [42][48][49].

2.6.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) is a transformer model designed to understand the meaning of text [42]. It reads words using the context around them, which helps the model understand that the same word can have different meanings depending on the sentence [42][50]. This makes BERT useful for classification tasks because it can consider the meaning of the whole message rather than relying only on individual keywords [50]. BERT can also be adapted to specific tasks by training it with labelled examples [42]. For example, it can be trained to place messages into different categories. However, BERT is a relatively large model and can require significant computing power, which may make it less suitable for devices with limited hardware [51].

2.6.2 DistilBERT

DistilBERT is a smaller and faster version of BERT. It is created using a method called knowledge distillation, where a smaller model learns from the behaviour of a larger BERT model [48]. Sanh et al. found that DistilBERT was around 40% smaller and 60% faster than BERT, while still keeping about 97% of its language understanding performance [48]. Other research has also shown that smaller transformer models can learn from larger models while keeping much of their original performance [51]. This makes DistilBERT useful for emergency message classification because it can still understand the context of a message while requiring less computing power. This may be particularly useful in disaster situations where access to powerful hardware could be limited.

2.6.3 T5

The Text-to-Text Transfer Transformer (T5) is designed to handle NLP tasks by treating both the input and output as text [49]. Instead of only selecting a category through a classification layer, T5 can receive a message as text and produce the required label as text. T5 has been used for a range of NLP tasks, including text classification, question answering, and summarisation [49]. This makes it a flexible model, although it can require more computing power than smaller models such as DistilBERT.

2.7 Existing Emergency Triage and Prioritisation Approaches

Disaster relief management involves the coordination of limited resources and personnel to ensure that aid is delivered to affected groups and individuals [12]. Following a natural disaster, relief teams must make rapid decisions despite incomplete and changing information [9]. The success of relief operations depends on the ability to prioritise critical tasks, allocate resources effectively, and adapt as new information becomes available [7][9]. Ineffective allocation of these resources can reduce the effectiveness of relief operations and prevent aid from reaching areas where it is most urgently required [7][12]. Accurate assessment and decision making methods are therefore important for determining where resources should be deployed and which areas require immediate attention [9]. Emerging technologies have created new opportunities for improving disaster management and decision support. AI can process large volumes of disaster-related information, classify incoming data, and support emergency response activities [8][10]. AI-based systems can also assist with triage assessment by categorising and prioritising emergency cases according to severity or urgency [13]. Current research into automated emergency message processing uses a range of ML and DL approaches to identify useful information from disaster-related text. These systems are commonly designed to filter irrelevant information, categorise messages according to emergency type, extract additional contextual information, or estimate how important a message may be to emergency responders. Although the modelling approaches differ, much of the existing research uses social media data as the primary source of emergency information.

Dwarakanath et al. reviewed automated ML methods used for emergency response and coordination through social media and identified classification as a common method for extracting relevant information from large volumes of disaster-related posts [8]. These

approaches included algorithms such as Support Vector Machines, Naïve Bayes, and Logistic Regression, which were used to distinguish relevant information or assign messages to predefined categories. Although these methods demonstrated that automated classification could support emergency response activities, their effectiveness depended heavily on a single data source [8]. Kejriwal et al. similarly investigated existing sentiment analysis and text classification tools using social media datasets collected from six disaster events [52]. Their study demonstrated that existing off-the-shelf AI tools could extract useful insights from disaster-related social media data, although their outputs sometimes required careful interpretation. The authors noted that real-world crisis datasets collected from platforms such as Twitter can contain noise and inconsistencies, and they observed disagreements between sentiment analysis tools in some cases, highlighting the need for contextual and qualitative verification [52].

DL architectures have also been used to classify disaster-related messages. Ullah et al. compared LSTM, Gated Recurrent Unit (GRU), and BERT models for disaster tweet classification [53]. Rather than representing a progression between older and newer technologies, the study provides a direct comparison between different current modelling approaches applied to the same classification problem. The reported results showed an accuracy of 93.2% for LSTM, 88.4% for GRU, and 96.2% for BERT [53]. This suggests that all three architectures can successfully process disaster-related text, although the transformer-based BERT model produced the strongest performance within the evaluated dataset.

Recurrent networks have also been incorporated into hybrid disaster classification systems. A Deep Parallel Hybrid Fusion Model (DPHFM) was proposed by Krishna et al, which uses Convolutional Neural Networks (CNNs) and Bidirectional LSTMs as base learners [54]. Combining these representations allows the classifier to consider both local patterns within a message and relationships across the wider sequence. This demonstrates that recurrent architectures continue to be investigated alongside transformer models where sequential information remains useful for the classification task.

Tan et al. proposed an intelligent emergency notification service using multi-task BERT models [15]. Their system first determines whether a social media message contains emergency-related information before classifying the message into an appropriate emergency category. Named Entity Recognition is then used to identify location information contained within the message so that relevant notifications can be delivered to affected users [15]. This demonstrates that transformer-based models can perform several related tasks within a single emergency processing system. Contextual representations produced by BERT allow the model to interpret the meaning of a message beyond individual keywords. However, the purpose of the system is primarily emergency detection, categorisation, and notification rather than prioritising messages for relief resource allocation [15].

Larger datasets have also been developed to support the evaluation of current emergency message classifiers. Alam et al. introduced HumAID, a dataset containing approximately 77,000 human-labelled social media messages across 19 major disaster events [55]. The dataset contains multiple humanitarian information categories and was evaluated using several classical supervised and DL approaches. By incorporating multiple disaster types,

events, locations, and time periods, HumAID provides a broader basis for event-level, event-type and future cross-event evaluation than datasets focused on a single incident [55].

Hybrid implementations combining transformer and recurrent architectures are also present within current research. BART-Disaster-X uses representations produced by a BART model with an LSTM-based classification component to categorise disaster-related messages across multiple information classes [56]. This implementation demonstrates that recurrent and transformer architectures do not necessarily represent competing generations of technology. Instead, components from both architectures can be combined where their characteristics are useful to the required classification task.

While previous research has often focused on identifying what type of information is contained in disaster-related social media messages, this does not always show how urgent that information is. The TREC Incident Streams (TREC-IS) framework considers both aspects separately. It looks at whether a system can identify the type of emergency information in a message and also how important that message may be for emergency responders to see quickly [57]. This means that two messages in the same category can still have different levels of importance. Wang and Lillis applied multi-task transfer learning to the TREC-IS problem using transformer-based language models [58]. Their approach addressed information type classification and priority prediction as related tasks, allowing shared model parameters to be learned jointly across both tasks. The implementation demonstrates that contextual language models can be used not only to classify the type of information conveyed in a crisis-related message but also to estimate its importance for emergency response. Later TREC-IS work continued to investigate transformer-based approaches for information type and criticality prediction [59]. These systems demonstrate that automated prioritisation of crisis messages is technically feasible and that pre-trained language models can be adapted to identify and prioritise information that may require the attention of emergency services.

Despite the range of implementations available, several common limitations remain. The most significant is the reliance on social media datasets. Although these datasets provide valuable examples of real disaster communication, they may not fully represent messages sent directly by affected civilians or by relief personnel. However, current implementations demonstrate that a range of NLP architectures, including conventional ML models, LSTMs, GRUs, transformer models and hybrid architectures, can successfully process disaster-related text.

Rather than showing a simple progression from one architecture to another, the literature demonstrates that different approaches remain applicable depending on the required task, dataset, computational constraints, and desired output.

2.8 Literature Review Summary

The literature demonstrates that AI-based NLP methods are capable of extracting and classifying useful information from disaster-related text. Traditional machine learning methods, recurrent architectures such as LSTMs, and transformer-based models have all been successfully applied to crisis-message classification, although recent implementations

show particularly strong performance from contextual transformer models such as BERT. Existing research has also demonstrated that disaster messages can be classified according to both information type and operational importance, indicating that automated systems can potentially support emergency triage and prioritisation. However, several limitations remain within the existing literature. A large proportion of current research relies on social media data, particularly messages collected from platforms such as X (otherwise known as twitter). These datasets provide useful examples of real disaster communication but may not fully represent messages transmitted directly by affected civilians or relief personnel. Furthermore, many implementations focus primarily on classification performance without considering whether the selected model can operate efficiently on lower-resource hardware that may be required when conventional communication infrastructure is unavailable.

Based on these findings, this study evaluates AWD-LSTM, BERT, DistilBERT, T5, and T5 (small) approaches for the cascading emergency message classification. These architectures provide a comparison between recurrent, encoder-only transformer, and encoder-decoder transformer approaches with different computational requirements. The literature review therefore informs both the model selection used within the experimental methodology.

3. Research Methodology

This section introduces the research methodology used to address the research question and objectives. It describes the development of the NLP classification models, the proposed decentralised architecture, data collection and generation procedures, and the methods used to evaluate model performance.

3.1 Research Design

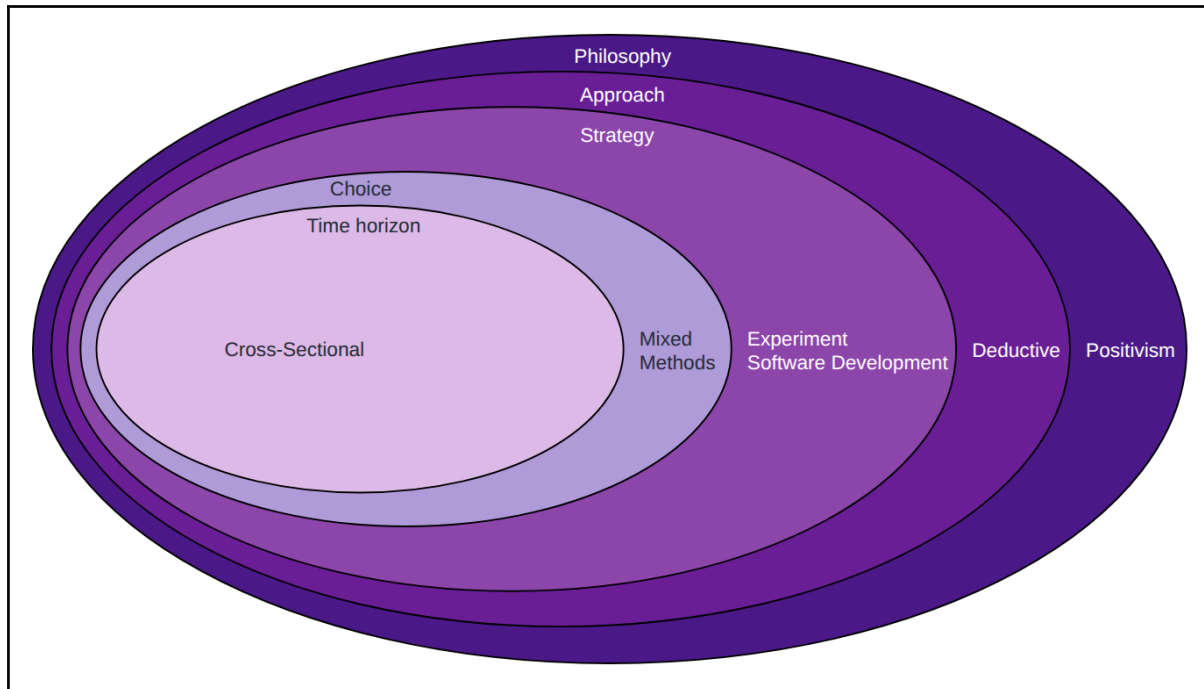


Figure 3.1: Research Design

This study adopts a positivist research philosophy and a deductive approach. The research strategies consist of experimentation and software development, while a mixed-methods choice and a cross-sectional time horizon are adopted [60].

The positivism philosophy is selected because the study relies on observable data, controlled experiments and evaluation metrics of the different AI models [60]. A deductive approach was adopted because established NLP theories and previously developed classification architectures informed the selection of candidate models, which were subsequently evaluated against predefined performance criteria. This makes it appropriate due to how the experiments are designed and the labelled datasets that train and test the models. This enables the candidate models to be compared and the most suitable model to be selected for the proposed architecture. The classification model learns the relationship between text features and predefined labels from individual message examples to categorise messages according to emergency type and derive priority level from classification [61].

The research strategy combines experimentation with software development. The experimental strategy is used to train, validate, test, and compare candidate classification

models under controlled conditions. The synthetically generated dataset was prepared and divided into suitable training, validation, and testing subsets. Where appropriate, stratified sampling was applied to preserve the distribution of emergency classes. The same evaluation procedure was applied to each candidate model to support a fair comparison [62], making it appropriate for assessing the given metrics in each given experiment to find out their given effectiveness and reliability [63]. The software development strategy is used to create an application to generate synthetic data.

Following the evaluation of the classification models, a decentralised communication architecture was designed to demonstrate how the best performing model could be deployed. The architecture will consider how messages are received, classified, prioritised, and forwarded when conventional communication infrastructure is unavailable. This development and evaluation process is consistent with design science research in which a technological artefact is created and evaluated as a response to an identified problem [64].

This study adopts a mixed methods choice by combining quantitative and qualitative methods commonly used in the AI field [65][66]. Quantitative experimentation is well established in crisis message classification, where human-annotated datasets have been used to train, test, and compare ML and DL models [67][68]. Previous research has also classified crisis messages according to humanitarian information type and priority level [69]. The study extends these quantitative approaches by incorporating open-ended survey responses and qualitative error analysis to explain recurring classification failures. The quantitative component forms the main part of the research and measures the ability of each model to classify messages. Accuracy, precision, recall, and F1-score were calculated from the model predictions. Precision is important for limiting false alerts, whereas recall is important for reducing the number of genuine emergency messages that remain undetected. However, accuracy, precision, recall and F1-score may be influenced by class prevalence and bias and may provide an incomplete assessment of model performance [70]. Where appropriate, confusion matrices, specificity, ROC curves and area under the ROC curve will also be examined to provide a more comprehensive evaluation. Model inference time and where feasible, resource consumption will also be measured to assess suitability as a real-time system [71]. The qualitative component consists of an open-ended survey and visual error analysis. During the data collection phase, a survey was conducted to collect different messages and phrases that could occur within an emergency network that would not otherwise have occurred when creating the first iteration of the synthetic data generator. The error analysis was conducted on incorrectly classified messages to identify possible problems, such as: Ambiguous wording, Informal language, Spelling errors, Limited context, or Multiple emergencies within a single message.

The results of this analysis will help explain why particular errors occurred and indicate where the classification model, dataset, or message processing procedure may require improvement [72].

For data gathering, the study implements a cross-sectional time horizon because the synthetic data and open-ended survey were collected and evaluated within a given timeframe rather than multiple points in time [73]. Once the data has been gathered, experiments were conducted to give all candidate models consistent experiment conditions with the snapshot of the data gathered at the given time. Once a suitable candidate is

selected, the study will assess immediate model performance and perceived system suitability rather than changes in performance over an extended period.

3.2 Open-Question Survey

An open-question survey was conducted to collect examples of how individuals may naturally communicate during an emergency. The purpose of the survey was not to measure participant opinions or attitudes, but to gather a wider range of vocabulary, sentence structures, and writing styles that could be incorporated into the synthetic data generation process. This was intended to reduce the reliance on researcher-created phrases and improve the variation of the messages contained within the dataset. The initial synthetic dataset was generated using predefined emergency categories, subcategories, and priority levels.

Previous research has shown that the assumptions, collection methods, and subjective decisions involved in dataset construction can introduce representational bias and other limitations that may affect the behaviour of ML models [74]. Therefore, relying entirely on researcher-defined emergency phrases could result in repetitive linguistic patterns.

Open questions were selected because they allow participants to respond using their own words rather than selecting from predefined responses. This was particularly important for this study because the variation within the written responses was itself the information being collected. Participants were presented with emergency-related scenarios and asked to provide a message that they may send if they were experiencing or reporting the given situation. The responses could therefore include differences in terminology, sentence length, grammar, spelling, and the amount of contextual information provided. For example, multiple participants described the same medical emergency using substantially different language. One participant explicitly stated that an individual is "unconscious", while another described the situation as "will not wake up". Although both messages can represent a similar emergency, the vocabulary and sentence structure are different. Collecting this variation provides additional examples of how emergency information may be communicated and can therefore help prevent the synthetic dataset from becoming dependent on a limited number of predefined keywords.

A total of 10 participants took part in the survey. Participants were recruited through convenience sampling and consisted of friends, family members, and work colleagues. This sampling approach was considered suitable because the purpose of the survey was exploratory and focused on collecting a range of natural language examples rather than producing statistically generalisable findings about a wider population.

The survey was distributed using physical paper forms. All participants received the same form and were informed that they were not required to complete every question. This allowed participants to leave individual scenarios unanswered where they did not feel able to provide a suitable response. Before completing the survey, participants were informed about the purpose of the study and were required to provide consent by signing the form. No demographic information was collected, as the study was concerned with the linguistic content of the responses rather than characteristics of individual participants.

The survey scenarios were derived from the predefined emergency categories used by the synthetic data generator (Appendix 2). Each category was further divided into specific subcategories. These categories were selected because they directly matched the classification structure already used within the dataset. Each emergency subcategory was associated with a predefined priority level, meaning participants were not asked to determine the emergency classification or priority themselves. Instead, they were asked only to provide the message that they believed could realistically be sent in that situation. This helped maintain consistency between the qualitative survey responses and the labelled structure required for supervised model training.

At the end of the survey, the responses were manually reviewed to identify useful phrases, terminology and different ways of describing the predefined emergency scenarios. Around 100 responses were incorporated into the synthetic data generation process. The survey responses were not treated as a standalone training dataset. Instead, they were used to expand the range of human-derived language available to the generator while retaining the predefined category, subcategory and priority structure.

3.3 Analysis Metrics

Performance metrics for the trained AI models were evaluated using a combination of classification and operational metrics. Accuracy, precision, recall, F1-score, and Receiver Operating Characteristic (ROC) curves were used to assess the predictive performance of each model [70]. Inference latency and throughput were also measured to assess whether the models were suitable for use within a real-time emergency communication system [71].

It was beneficial to use multiple evaluation metrics because a single value such as accuracy may not provide a complete representation of a given model's performance, particularly where the number of examples within each emergency category is uneven [70]. Therefore, the evaluation considered both how accurately each model classified emergency messages and how efficiently the models could process messages during a mock real-world scenario.

The main classification metrics are calculated using four values:

- True Positive (TP): The model correctly predicts that a message belongs to a particular class
- True Negative (TN): The model correctly predicts that a message does not belong to a particular class
- False Positive (FP): The model incorrectly predicts that a message belongs to a particular class
- False Negative (FN): The model fails to identify a message that belongs to a particular class

These values can be represented using a confusion matrix and used to calculate the different classification metrics [70].

3.3.1 Classification Metrics

Accuracy will measure the proportion of messages that were correctly classified [70].

$$Accuracy = \frac{\text{Number of correctly classified messages}}{\text{Total number of classified messages}}$$

Precision measures how often the model is correct when it predicts that a message belongs to a particular class [70]. Precision is important within disaster management because FP classifications may result in emergency resources being allocated incorrectly.

$$Precision = \frac{TP}{TP + FP}$$

Recall measures the proportion of messages belonging to a particular class that the model correctly identifies [70]. Recall is particularly important for emergency message classification because failing to detect a genuine emergency could result in an urgent message being incorrectly categorised or assigned a lower priority.

$$Recall = \frac{TP}{TP + FN}$$

The F1-score combines precision and recall into one measure using the harmonic mean [70]. This makes the F1-score useful for evaluating the proposed emergency classification system because both FP and FN may have negative consequences. A model with high recall but low precision may identify most genuine emergencies while also producing a large number of incorrect classifications. In comparison, a model with high precision but low recall may produce reliable predictions when it identifies a class but fails to detect a significant number of genuine emergency messages. The F1-score therefore provides a balance between these two measurements [70].

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Receiver Operating Characteristic (ROC) curves were used to examine the ability of the models to distinguish between different classes across the range of different classifications [70]. The ROC curve compares the True Positive Rate (TPR) against the False Positive Rate (FPR). The Area Under the ROC Curve (AUC) summarises the ROC curve into a single value. An AUC value closer to 1 indicates stronger class discrimination, while a value close to 0.5 represents performance approximately equivalent to random classification [70].

Because the proposed system performs multiclass classification, ROC analysis can be applied using a one-versus-rest approach. Each emergency class is treated as the positive class while the remaining classes are treated as negative, allowing an ROC curve and AUC value to be calculated for each emergency category.

Multiclass Evaluation

The proposed model performs multiclass classification because emergency messages can belong to multiple possible categories rather than only a positive or negative class. Precision, recall and F1-score can therefore be calculated separately for each class before being combined into overall measurements [70]. This is done by using macro averaging which calculates a metric independently for every class before taking the average.

$$\text{Macro Metric} = \frac{1}{C} \sum_{i=1}^C M_i$$

Where C is the total number of classes, and M_i is the specific metric score calculated independently for class i.

3.3.2 Inference Latency and Throughput

In addition to classification performance, inference latency was measured to assess how quickly each trained model could produce a prediction when presented with a new emergency message. Inference benchmarking commonly considers latency because model suitability depends not only on predictive quality but also on how quickly predictions can be produced under deployment conditions [71].

$$\text{Inference latency} = \frac{\text{Total Inference Time}}{\text{Number of Messages}}$$

This metric is important for the proposed emergency communication system because messages may need to be classified and prioritised in real time. A model that produces highly accurate predictions but requires a long period of time to process each message may be unsuitable for deployment where rapid decision making is required.

Throughput measures the number of messages that a model can process within a given period of time and is another commonly used measure of inference performance [71].

$$\text{Throughput} = \frac{\text{Number of Messages}}{\text{Total Inference Time}}$$

Throughput is relevant within disaster scenarios because the communication system may receive a large number of messages within a short timeframe. A model that is capable of processing individual messages accurately but has low throughput could become a bottleneck when message volume increases.

4. Architecture Design

This section presents the proposed architecture for integrating the classification model within a decentralised disaster communication network. The architecture consists of two principal components: the decentralised communication network and the AI classifier module. The decentralised network provides a method through which civilian and relief team messages could be transmitted when conventional communication infrastructure is unavailable. The AI classifier module processes these messages and assigns an emergency category, subcategory, and associated priority level before the resulting information is forwarded to relief personnel.

The architecture is intended to demonstrate how the classification models evaluated within this study could be deployed within a wider disaster response environment.

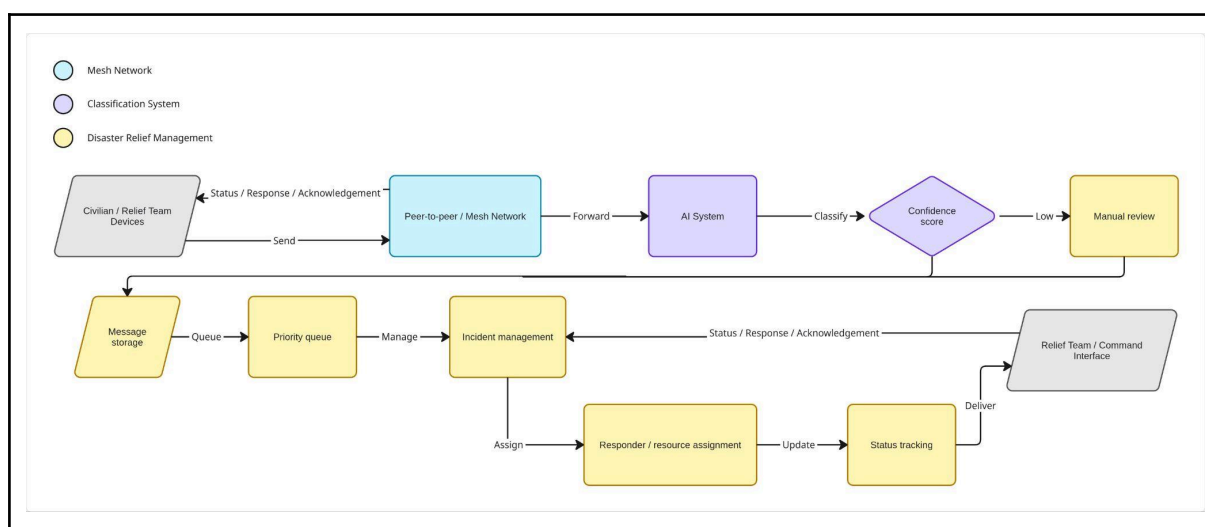


Figure 4.1: High-level architecture

4.1 Decentralised Network

The proposed communication architecture uses a decentralised mesh network to allow messages to be transmitted between civilian devices, relief personnel, and the AI classification system. This approach was selected because disasters may damage telecommunication systems [2]-[5]. Previous disaster communication research has therefore investigated ad-hoc, device-to-device, and mesh networking as methods of temporarily restoring connectivity following infrastructure disruption [5][6][16]. Within a mesh network, participating devices can communicate directly with nearby nodes and relay information on behalf of other devices. A message therefore does not necessarily require a direct connection between its sender and its final destination. Instead, it can travel through intermediate nodes until it reaches a device capable of processing or forwarding it. WiMesh demonstrates the potential use of this approach within disaster environments, where wireless devices can be used to establish communication despite limited infrastructure [5].

Similarly, device-to-device disaster networks have been investigated as a method of allowing communication to continue when conventional network infrastructure is unavailable [6].

The principal advantage of this architecture is resilience. A centralised network can become unavailable if a critical communication node fails, whereas a mesh architecture may provide alternative routes when individual nodes become unavailable. This characteristic is particularly relevant within disaster environments because devices may move, lose power, or become physically inaccessible. However, the network should not be considered immune to failure. Increasing the number of wireless nodes can increase communication latency and reduce available throughput, while rapidly changing network topology may require routes to be continuously updated [5]. The proposed system therefore uses mesh networking as a method of increasing communication availability rather than guaranteeing uninterrupted connectivity.

As shown in Figure 4.2, the mesh architecture may also be extended through elevated communication nodes. Ground-level wireless communication can be restricted by buildings, terrain, debris, and the transmission range of individual devices. Unmanned Aerial Vehicles (UAVs) can potentially operate as temporary aerial relays or base stations, providing flexible coverage where terrestrial infrastructure has been damaged. Research into UAV-assisted disaster communication has identified rapid deployment and flexible positioning as important advantages of aerial communication systems. UAVs can also act as intermediate relays between ground nodes that would otherwise be outside direct communication range [11]. A limitation of conventional UAVs is their dependence on batteries, which restricts the period for which they can remain airborne. Suspended or tethered aerial platforms provide another possible approach. Tethered balloons can remain elevated for longer periods and provide a comparatively large communication coverage area while maintaining a physical connection to a ground station. Research into high-altitude tethered balloon base stations has specifically investigated their deployment for emergency communication networks following infrastructure disruption [75]. Consequently, the proposed architecture does not depend on a single type of aerial platform. UAVs may be appropriate where rapid deployment and mobility are required, whereas tethered platforms may provide longer duration coverage over a more fixed area.

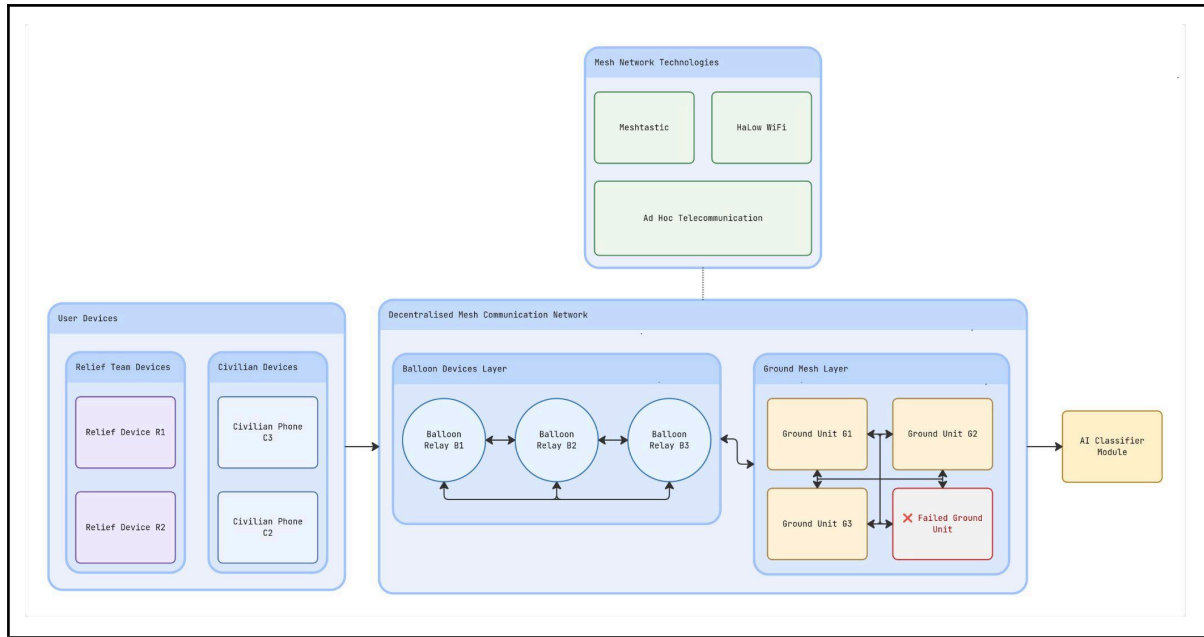


Figure 4.2: Decentralised communication network

4.2 Emergency Message Classification Module

The emergency message classification module is responsible for converting incoming emergency messages into structured information that can subsequently be forwarded to relief personnel. As shown in Figure 4.3, an incoming message is first prepared for processing before being passed to the trained NLP classification model. The first stage consists of message preprocessing and tokenisation. Preprocessing prepares the received text for the input format expected by the selected model, while tokenisation converts the text message into numerical tokens that can be processed by the NLP architecture. The specific tokenisation procedure depends on the deployed model. Transformer models such as BERT and DistilBERT use their corresponding pretrained tokenisers, while the AWD-LSTM model uses the vocabulary and numericalisation procedure established during training.

The trained classifier then performs cascading classification using the two-stage structure evaluated in Section 5. The first stage predicts one of the ten broad emergency categories. Following this prediction, the second stage determines the appropriate subcategory while restricting the available outputs to subcategories belonging to the selected parent category. This prevents structurally invalid category-subcategory combinations from being returned and reduces the number of possible labels considered during the second-stage.

Following classification, a priority level is obtained from the resulting subcategory. Priority is therefore not independently predicted by the AI model. Instead, each predefined emergency subcategory is associated with a priority value in the classification metadata. For example, emergencies identified as requiring immediate intervention can be mapped to a higher priority than less urgent operational messages. This deterministic relationship ensures that identical classifications receive consistent priority values.

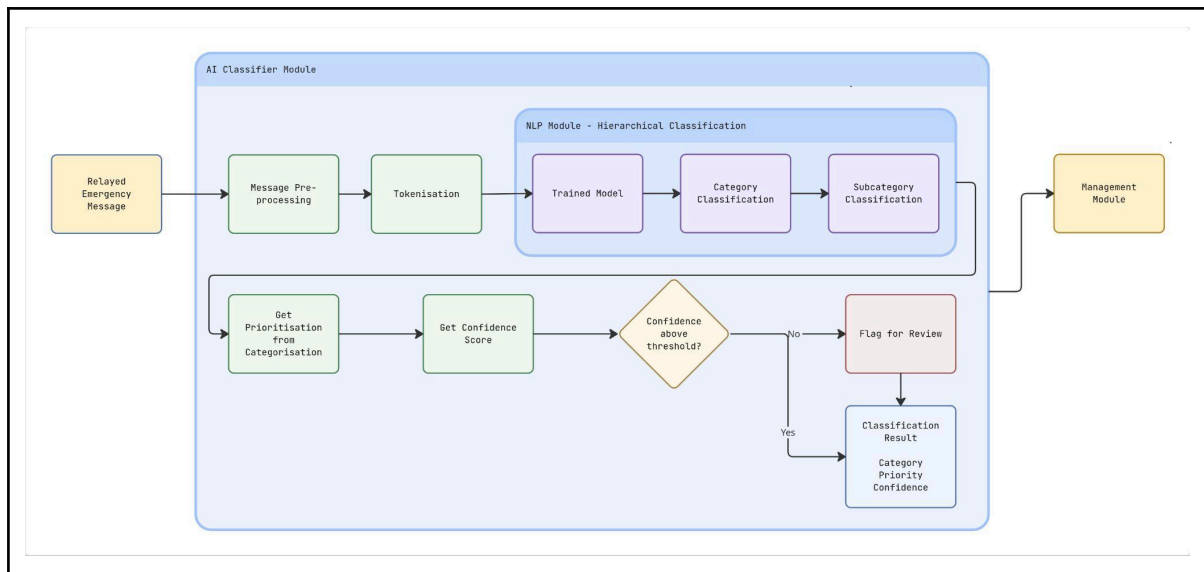


Figure 4.3: AI Classifier Module

5. Experiments and Results Analysis

This section goes through the experimental results. Five pretrained language models were evaluated: AWD-LSTM, BERT, DistilBERT, T5, and T5 (small). Each model was set up to work in two cascading stages. The first stage predicts one of ten broad emergency categories, while the second-stage predicts the corresponding subcategory from the category selected by the first-stage. This design ensures that the final category-subcategory pair is hierarchically consistent. The analysis compares the models using the metrics discussed in Section 3. Macro averaged measures are emphasised because they assign equal importance to every class and therefore provide a more informative assessment than accuracy alone when class frequencies differ. In addition to test performance metrics, this section examines error propagation between the two stages, model storage requirements, and inference testing on a desktop PC and an Orange Pi 5 Plus. Visual representations are placed at the end of this section.

5.1 Software and Setup

The experiments were executed using Python and PyTorch on a Camber Cloud GPU instance. This allowed for hardware acceleration via CUDA-enabled hardware, reducing training times. As a result, all model training and evaluation were performed on the GPU. Table 5.1 presents the principal software versions used within the project environment.

Python	3.12.12	This was set up as a Python environment
Python Imports		
PyTorch	>=2.13.0	Training and inference for neural network models
Transformers	>=5.16.1	Loading and fine-tuning pretrained transformer models
Datasets	>=2.10.0	Preparing datasets for model training and evaluation
Scikit-learn	>=1.3.0	Stratified data splitting and classification metrics
fastai	>=2.7.0	Training and evaluating AWD-LSTM classifiers
NumPy	>=1.24.0	Numerical operations and array processing
pandas	>=2.0.0	Loading and manipulating tabular datasets
Matplotlib	>=3.7.0	Visualising classification results

Table 5.1: Experimental software environment

All five experiments used the same disaster_messages.csv dataset containing 10,000 synthetic emergency messages, with a fixed random seed of 42. The dataset was split into 70% training, 15% validation, and 15% testing data. Stratified sampling was used to maintain the distribution of emergency categories across each split. The test set was kept separate from training and hyperparameter tuning.

5.1.1 Synthetic Data Generation

A synthetic dataset was generated to provide labelled emergency messages for training, validating, and testing the candidate models. The process was implemented using a Python script that combined predefined disaster categories, subcategories, priority levels, and sentence components to produce varied emergency messages. The resulting dataset was stored as a CSV file containing the fields type, category, subcategory, priority, and message. The generator loaded the disaster category information and corresponding sentence structures from separate JSON configuration files. Each category contained several subcategories, with each subcategory associated with a predefined priority value. For each generated record, a disaster category was first randomly selected, followed by a subcategory. A phrase associated with the selected subcategory was then randomly chosen and combined with an optional opening and closing sentence component. To increase linguistic variation, approximately 40% of eligible messages could also contain a second description selected from the same subcategory. The category and subcategory therefore determined the label assigned to each generated message, while the priority value was obtained from the selected subcategory.

Duplicate records were prevented during generation by maintaining a collection of previously generated combinations. Generation continued until the requested number of unique records had been obtained, after which the records were randomly shuffled before being written to the final CSV file. The generator also supported the use of a fixed random seed. This allowed the random generation process to be reproduced when the same seed, sentence structures, and category definitions were used. For the experiments in this study, the final synthetic dataset contained 10,000 emergency messages using the seed 42 and then was subsequently divided into the training, validation and testing subsets described in Section 5.1.

5.1.2 Dataset Structure

An example of the dataset structure is shown in Table 5.2.

Type	Category	Subcategory	Priority	Message
disaster	flooding	flash_flood	1	SOS. flood water rising fast. Floodwater rose suddenly and is moving toward us. pls respond.
disaster	infrastructure_damage	other	4	sum1 help. Communication equipment across the area has failed. public facility destroyed. send help quick.
disaster	fire	wildfire	1	emergency situation. forest fire heading towards us. wildfire spreading toward us. Please acknowledge this message and dispatch assistance.
disaster	fire	structural_fire	2	need emerg svc. smoke coming from the roof. pls respond.

Table 5.2: Example records from the synthetic emergency dataset

The examples demonstrate that messages within the same general category can contain different subcategories and priority levels. They also show the variation introduced into the language of the generated messages, including abbreviated terms, short fragmented

sentences, and more descriptive statements. This variation was included to provide the models with emergency messages that were not restricted to a single sentence structure or writing style. The final dataset contained 10,000 unique synthetic emergency messages. Before model training, the dataset was divided into training, validation and testing subsets using a 70:15:15 split. Stratified sampling was used to maintain the distribution of emergency categories across each subset, reducing the likelihood that less frequently occurring categories would be underrepresented during validation or testing.

5.1.3 Hyperparameter Configuration

To maintain consistency between the transformer-based experiments, a shared training-argument function was used to define the common fine-tuning configuration. Unless otherwise stated, all transformer models were trained for a maximum of three epochs using a weight decay of 0.01, a fixed random seed of 42, evaluation and checkpoint saving at the end of each epoch, and automatic restoration of the checkpoint with the lowest validation loss. A maximum of two checkpoints were retained during training. Model-specific parameters, including learning rate, per-device batch size and gradient accumulation, were varied where required and are reported in Tables 5.3 and 5.4.

Model	Maximum length	Train / evaluation batch	Gradient accumulation	Initial learning rate
AWD-LSTM	72	64/64	1	1×10^{-2}
BERT <i>bert-base-uncased</i>	128	16/32	1	2×10^{-5}
DistilBERT <i>distilbert-base-uncased</i>	128	16/32	1	2×10^{-5}
T5 (small) <i>t5-small</i>	128	16/32	4	1×10^{-4}
T5 <i>t5-base</i>	128	16/32	4	1×10^{-4}

Table 5.3: Training and Hyperparameter configuration Part 1

Model	Regularisation and schedule	Training duration and selection
AWD-LSTM	<i>drop_mult=0.5</i> ; fastai one-cycle fine-tuning	1 frozen epoch followed by 3 unfrozen epochs; final learner exported
BERT	weight decay 0.01; linear learning-rate decay; no warm-up	Maximum 3 epochs; evaluate and save each epoch; lowest validation-loss checkpoint
DistilBERT		
T5 (small)		
T5		

Table 5.4: Training and Hyperparameter configuration Part 2

Preliminary validation experiments indicated that directly applying the BERT and DistilBERT optimisation configuration to T5 (small) produced substantially weaker performance. Two model-specific parameters were therefore adjusted: learning rate and gradient accumulation were both increased. The per-device training batch size remained 16, resulting in an effective batch size of 64 for T5. The remaining shared training configuration was retained, including a maximum of three epochs, weight decay of 0.01, epoch-based evaluation and checkpointing, selection of the checkpoint with the lowest validation loss, and a fixed random seed of 42.

With the revised settings, T5 (small)'s category accuracy increased to 98.33% and category macro F1-score to 98.34%; end-to-end path accuracy increased to 93.73% and macro F1-score to 93.35%. The change was therefore retained on observational grounds rather than merely because T5 conventionally uses a different learning rate. The same revised configuration was then transferred to T5.

The objective was to establish a reasonable task-specific configuration. BERT and DistilBERT performed strongly under the initial configuration, whereas T5 (small) required adjustment because direct transfer of the encoder-classifier settings produced substantially weaker validation performance.

5.2 Results and Analysis

All results in this section are based on the test set of 1,500 messages. The same stratified 70:15:15 split for training, validation, and testing, along with the same random seed was used for every model. This ensured that all models were compared under the same conditions. The dataset contains 10,000 unique emergency messages across ten categories and 33 subcategories. The main categories are fairly balanced (see figures at the end of this section). Each candidate model was tested in a simulated environment to test for inference latency and resource cost.

5.2.1 Model Performance

Tables 5.5 and 5.6 provide an overall comparison of the five cascading classifiers. The strongest result in each column is shown in bold.

Model	Accuracy	Macro F1	ROC AUC
AWD-LSTM	98.33%	98.36%	99.99%
BERT	100.00%	100.00%	100.00%
DistilBERT	100.00%	100.00%	100.00%
T5 (small)	98.33%	98.34%	99.89%
T5	100.00%	100.00%	100.00%

Table 5.5: Overall category performance

Model	Accuracy	Macro F1	ROC AUC
AWD-LSTM	97.80%	97.84%	99.10%
BERT	99.93%	99.90%	100.00%
DistilBERT	99.80%	99.74%	100.00%
T5 (small)	93.73%	93.35%	99.02%
T5	99.87%	99.84%	100.00%

Table 5.6: Overall subcategory performance

BERT achieved the best end-to-end result, correctly classifying all 1,500 broad categories and 1,499 of the 1,500 category-subcategory paths. Its subcategory accuracy of 99.93% and macro F1-score of 99.90% indicate that its performance was consistent across the 33 subcategories. T5 ranked second by subcategory accuracy, with two errors, followed by DistilBERT with three errors. The difference among these three models was therefore no more than two test messages. DistilBERT is particularly competitive because it matched BERT and T5 at category-level and lost only 0.13% of subcategory accuracy relative to BERT. AWD-LSTM also performed strongly, reaching 98.33% category accuracy and 97.80% subcategory accuracy. T5 (small) obtained the weakest end-to-end result. Although its category accuracy equalled that of AWD-LSTM, its subcategory accuracy fell to 93.73%, showing that most of its errors arose during the subsequent subcategory stage.

The ROC AUC values were high for every model. However, these results should be interpreted alongside macro F1-score and accuracy. A high one-versus-rest ROC AUC shows that a model generally ranks the correct class above alternatives over a range of thresholds; it does not mean that the class selected by the model is always correct. This distinction is visible for T5 (small), which achieved a subcategory ROC AUC of 99.02% but a lower subcategory accuracy of 93.73%.

5.2.2 Category-Level Performance

The first stage separates messages into the ten broad emergency categories: medical, structure damage, utility, missing person, supply request, flooding, trapped, fire, infrastructure damage, and evacuation. Table 5.7 shows the complete category-level macro metrics.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Correct / 1,500
AWD-LSTM	98.33%	98.39%	98.35%	98.36%	1,475
BERT	100.00%	100.00%	100.00%	100.00%	1,500
DistilBERT	100.00%	100.00%	100.00%	100.00%	1,500
T5 (small)	98.33%	98.38%	98.34%	98.34%	1,475

T5	100.00%	100.00%	100.00%	100.00%	1,500
----	---------	---------	---------	---------	-------

Table 5.7: Category-Level Performance

BERT, DistilBERT, and T5 produced perfect category predictions. This shows that the language patterns distinguishing the ten broad emergency types were captured reliably within the current dataset by both encoder-only classification models and the larger encoder-decoder model. The perfect category result is also important when interpreting subcategory performance. Since no errors were made in the first-stage, the few remaining mistakes came only from distinguishing between subcategories. AWD-LSTM and T5 (small) each made 25 category errors. Their total scores were nearly identical, but their per-class behaviour was different. For AWD-LSTM, the lowest category recall was observed for *flooding* at 95%, while the lowest precision was 95% for *utility*. T5 (small)'s lowest recall was 93% for *medical*, and its lowest precision was also 95% for *utility*. Both models learned the task well but still confused some messages when different incident types used similar words. For example, a message about water could refer to an outage, flooding, or a supply request, while damage to utilities could also be confused with general infrastructure damage.

The small gap between accuracy and macro F1-score across all models confirms that the category-level results were not produced by favouring high-frequency classes. In practical terms, the first-stage provides a strong routing mechanism for the second-stage classifier, particularly for the three models that made no routing errors.

5.2.3 Subcategory-Level Performance

The second stage is more challenging because it predicts 33 specific outcomes. The predicted category from the first-stage is used to restrict the second-stage predictions to only valid subcategories. Table 5.8 shows the final end-to-end results after applying this hierarchical constraint.

Model	Accuracy	Macro Precision	Macro Recall	Macro F1	Correct / 1,500
AWD-LSTM	97.80%	97.94%	97.85%	97.84%	1,467
BERT	99.93%	99.91%	99.90%	99.90%	1,499
DistilBERT	99.80%	99.75%	99.73%	99.74%	1,497
T5 (small)	93.73%	94.26%	93.09%	93.35%	1,406
T5	99.87%	99.84%	99.83%	99.84%	1,498

Table 5.8: End-to-end subcategory results

The three highest performing transformer models again became clear leaders. BERT made one error, T5 made two, and DistilBERT made three. Given the 1,500-example test set, these correspond to performance differences of only one to two samples. The results do not

identify a clearly superior candidate that has any large advantage from their performance alone without repeating training or changing datasets. Nevertheless, the recorded experiment shows that all three architectures fit the current task extremely well.

AWD-LSTM made 33 end-to-end subcategory errors. Its lowest class-level F1-scores were 94% for *fire/structural_fire*, *structure_damage/moderate* and *utility/power_outage*. The first two errors are consistent with overlap among severity or incident descriptions, while power outage statements can share general service failure language with other utility messages. Despite these errors, every AWD-LSTM subcategory achieved an F1-score of at least 94%, demonstrating robust performance across the label hierarchy. T5 (small) made 94 subcategory errors and exhibited wider class-to-class variation. Its weakest F1-score was 83% for *medical/unconscious*, followed by 86% for *utility/water_outage* and 87% for *medical/illness*. Other difficult labels included *medical/heavy_bleeding*, *supply_request/water*, and *utility/power_outage*, each with an F1-score of 88%. The pattern shows difficulty telling apart closely related labels, rather than a general failure across the task. By contrast, it achieved perfect precision and recall for *trapped/in_vehicle*, *trapped/under_debris* and *utility/gas_leak*, whose messages appear to contain more distinctive cues.

The comparison between T5 and T5 (small) shows the strongest effect of model scale in the experiment. Increasing from the smaller model to the base T5 improved subcategory accuracy by 6.14% and macro F1-score by 6.49%, while also eliminating category-level errors. The corresponding difference between BERT and DistilBERT was only 0.13% in subcategory accuracy, suggesting that compression through DistilBERT preserved nearly all of the useful discriminative capacity for this dataset.

5.2.4 Error Propagation in the Cascade

In a cascade, an incorrect category prediction prevents the final category-subcategory path from being correct, because the second-stage candidates are restricted to children of the predicted category. Table 5.9 separates these unavoidable routing failures from the additional errors made when selecting among subcategories.

Model	Category routing errors	Total end-to-end errors
AWD-LSTM	25	33
BERT	0	1
DistilBERT	0	3
T5 (small)	25	94
T5	0	2

Table 5.9: Error propagation from category to subcategory

For AWD-LSTM, 25 of the 33 final errors were already determined by incorrect first-stage routing, leaving only eight additional subcategory errors. Its second stage was therefore

effective whenever the hierarchy was entered through the correct branch. T5 (small) presents a different pattern. The same 25 routing failures occurred, but 69 further errors were introduced during subcategory selection. The principal limitation of T5 (small) is consequently not the cascading architecture itself, but the model's ability to resolve finer classification labels at its current capacity and training configuration. The zero category errors recorded by BERT, DistilBERT, and T5 isolated their second-stage performance from cascade propagation on this test. Their one, three and two final errors respectively show that constraining predictions to valid children can support highly accurate fine-grained decisions once the broad category has been established. However, a production system should retain a category confidence threshold or an escalation path, because a confident second-stage prediction cannot recover from an incorrect branch selected in the first-stage.

5.2.5 Model Complexity and Deployment Trade-offs

Predictive accuracy is essential in emergency-message classification, but deployment also depends on memory, storage, latency, and operational constraints. Table 5.10 reports the observed combined disk footprint of each saved two-stage model. These values include both category and subcategory artefacts, together with the configuration and tokenizer files stored in the corresponding model directories.

Model	Category artefact	Subcategory artefact	Combined footprint
AWD-LSTM	123.2 MiB	123.3 MiB	246.5 MiB
BERT	418.4 MiB	418.4 MiB	836.8 MiB
DistilBERT	256.1 MiB	256.2 MiB	512.3 MiB
T5 (small)	233.2 MiB	233.2 MiB	466.3 MiB
T5	852.7 MiB	852.7 MiB	1,705.3 MiB

Table 5.10: Saved two-stage model footprint

AWD-LSTM had the smallest combined footprint and remained accurate, making it a plausible option when storage is the dominant constraint. DistilBERT provides the strongest overall trade-off between classification performance, storage requirements and inference performance. Its cascade is approximately 39% smaller than BERT and 70% smaller than T5, while its subcategory accuracy is only 0.13% below BERT and 0.07% below T5. T5 offers no measured accuracy advantage over BERT in this experiment despite requiring approximately twice the saved storage. T5 (small) is not favoured by the present accuracy-storage comparison. Its saved cascade is larger than AWD-LSTM, but its subcategory accuracy is 4.07% lower. Further hyperparameter optimisation or additional fine-tuning may improve its position, but the recorded configuration does not provide a deployment advantage over the alternatives.

5.2.6 Inference Benchmarking

The saved two-stage models were benchmarked on two CPU platforms: a desktop PC with an AMD Ryzen 7 5700X3D using 16 threads and an Orange Pi 5 Plus single board computer using eight threads. The benchmark times combine the tokenisation, category inference, conditioned subcategory inference and post-processing. Model loading is excluded. For every model and batch size, one warm-up iteration was done before five repeats. Batch sizes of 4, 16, and 64 resulted in 20, 80, and 320 messages being processed for each given candidate. Table 5.11 reports mean per-message latency and throughput at the largest tested batch size and the setting most relevant to high-volume processing. The slowdown column is the Orange Pi mean per-message latency at batch size 64, the largest tested batch-message latency divided by the PC value for the same batch size.

Model	PC Latency	Orange Pi Latency	Orange Pi Slowdown	PC Throughput	Orange Pi Throughput
AWD-LSTM	9.99 ms	105.63 ms	10.58x	100.14 msg/s	9.47 msg/s
BERT	22.55ms	148.54 ms	6.59x	44.35 msg/s	6.73 msg/s
DistilBERT	10.68 ms	73.94 ms	6.92x	93.59 msg/s	13.52 msg/s
T5 (small)	97.24 ms	451.40 ms	4.64x	10.28 msg/s	2.22 msg/s
T5	319.07 ms	1,611.45 ms	5.05x	3.13 msg/s	0.62 msg/s

Table 5.11: End-to-end CPU inference performance

At batch size 64, DistilBERT achieved the lowest Orange Pi latency, 73.94 ms per message, and the highest Orange Pi throughput, 13.52 msg/s. On the PC it reached 93.59 msg/s at 10.68 ms per message, narrowly behind AWD-LSTM. These results strengthen its position as the most balanced deployment candidate when predictive quality, storage and high-volume inference performance are considered together. AWD-LSTM achieved the strongest PC batch 64 result, processing 100.14 msg/s at 9.99 ms per message. This was 7.0% more throughput than DistilBERT. Its Orange Pi throughput was 9.47 msg/s, 30.0% below DistilBERT. The T5 models were the least efficient at batch size 64. T5 processed only 3.13 msg/s on the PC and 0.62 msg/s on the Orange Pi, where its mean per-message latency was 1.61 seconds and whole batch completion required approximately 103.13 seconds. T5 (small) improved on T5 but remained slower than both encoder-only transformers. Figure 5.6 also shows that larger batches were not universally beneficial on the PC: BERT, T5 (small) and T5 reached their highest PC throughput before batch size 64. By contrast, every model achieved its highest Orange Pi throughput at batch size 64.

5.2.7 Class-Specific Error Analysis

The near-perfect results from BERT, DistilBERT and T5 leave too few test errors for a meaningful class-by-class comparison. AWD-LSTM and T5 (small) provide more evidence about the distinctions that remain difficult. Their errors were concentrated in labels with overlapping language or differences expressed through severity, context, or implied need.

Medical subcategories illustrate this challenge. A short message requesting an ambulance may imply injury, illness, bleeding or unconsciousness without explicitly naming the condition. T5 (small) achieved only 73% recall for *medical/unconscious* and 87% recall for *medical/illness*, whereas more explicit labels such as *utility/gas_leak* were classified perfectly. A similar pattern occurred in structure damage: T5 (small) achieved 100% recall but only 80% precision for *structure_damage/moderate*, indicating that it over-selected the middle severity class when descriptions did not clearly distinguish light, moderate, and severe damage. Several utility and supply labels also share key nouns. A reference to water may describe *utility/water_outage* or *supply_request/water*. T5 (small) recorded an F1-score of 86% for *utility/water_outage* and 88% for *supply_request/water*. Its 98% recall but 80% precision for *utility/power_outage* indicates that power outage predictions absorbed some related utility examples. AWD-LSTM showed the same tendency on a smaller scale, with 89% precision and 100% recall for *utility/power_outage*.

The hierarchical restriction prevents these models from selecting an invalid child after a category has been chosen, but it cannot resolve ambiguity within that child set. Improvements should therefore target subcategory discrimination. Useful directions include adding difficult paraphrases, training with contrastive examples from subcategories, calibrating category and subcategory confidence jointly and introducing an *uncertain* outcome for underspecified messages.

5.2.8 Visual Summary of Experimental Results

The following figures summarise the principal quantitative findings reported in Sections 5.2.1 - 5.2.6.

Model performance comparison

Figure 5.1 compares category-level and end-to-end subcategory performance. The expanded vertical scale makes the otherwise small differences among the leading models visible.

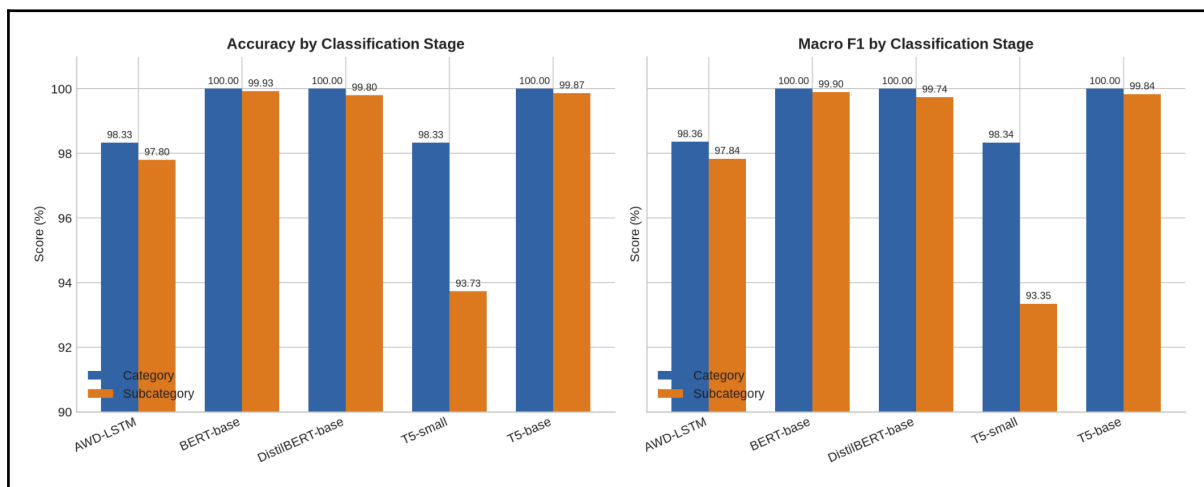


Figure 5.1: Category and End-to-End Subcategory performance

Hierarchical error propagation

Figure 5.2 separates category routing errors from errors introduced while selecting a child subcategory. It shows that T5 (small)'s main weakness lies in the second-stage.

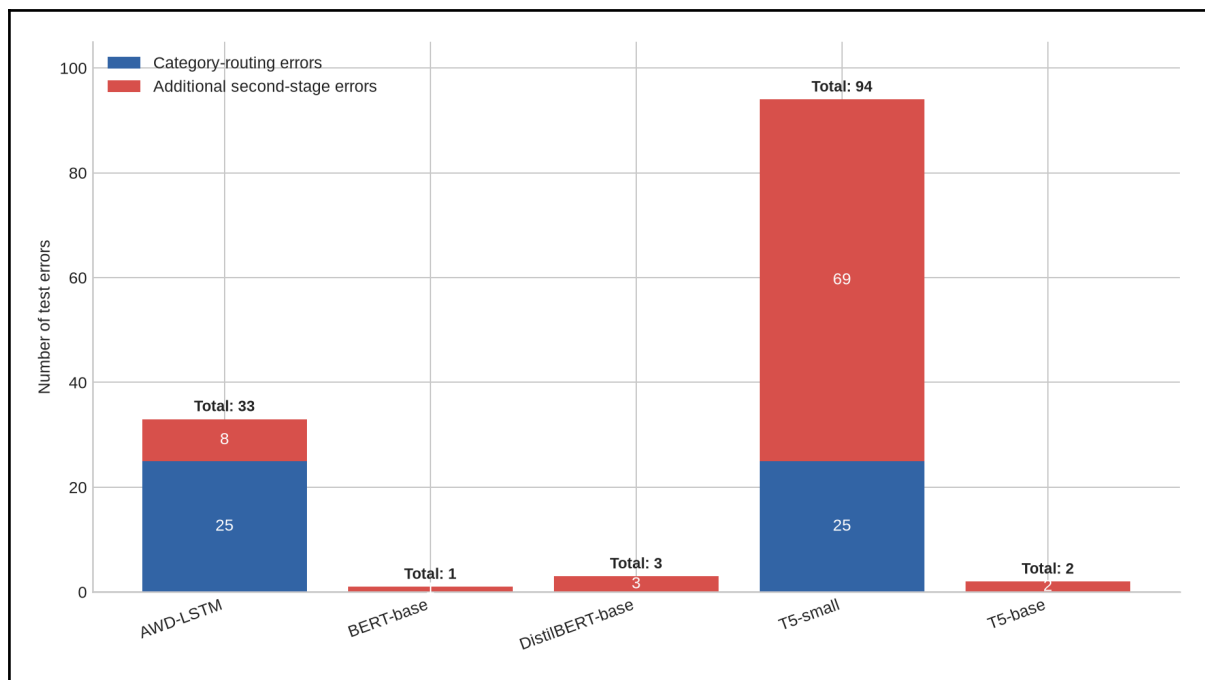


Figure 5.2: Cascading Error Propagation

Accuracy storage trade-off

Figure 5.3 plots end-to-end subcategory accuracy against the combined saved footprint of both stages. DistilBERT occupies the strongest region among the transformer models by remaining close to the maximum accuracy at a substantially smaller footprint than BERT and T5.

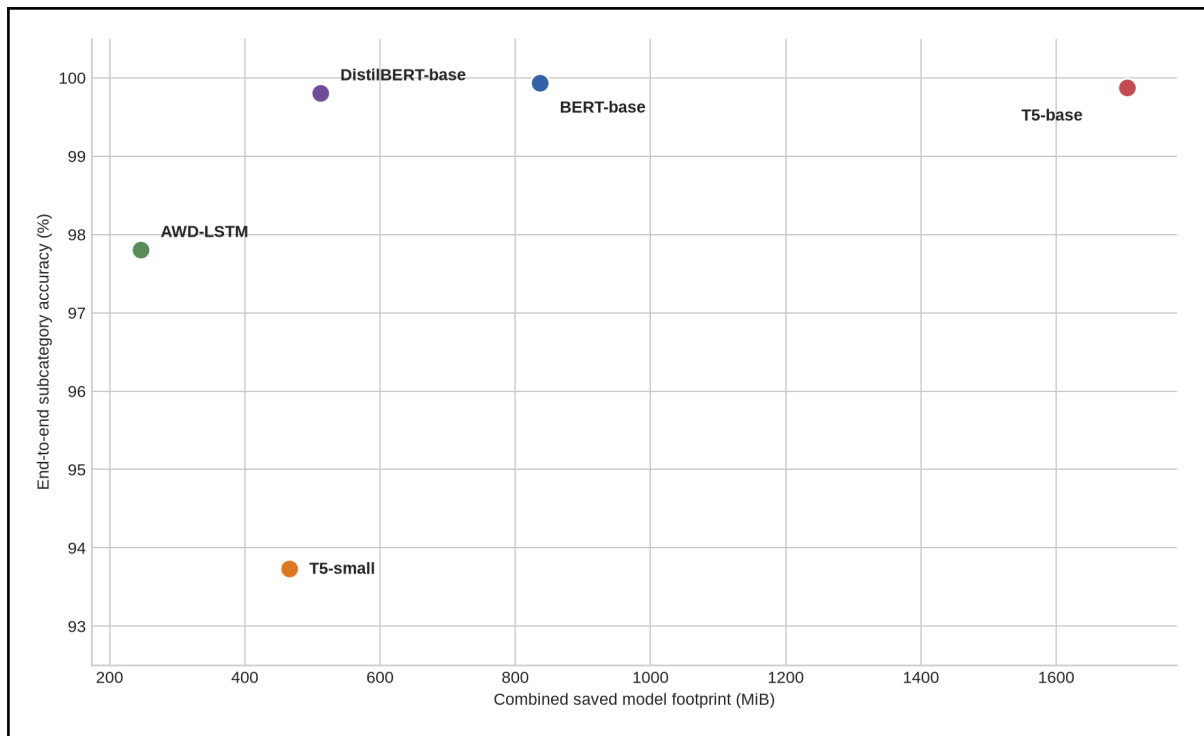


Figure 5.3: Accuracy Storage Trade-off

Class level performance heatmap

Figure 5.4 compares the per-subcategory F1-scores of AWD-LSTM and T5 (small), the two models with sufficient errors for an informative class-level visualisation. Lower-scoring cells identify the subcategory labels requiring further improvement.

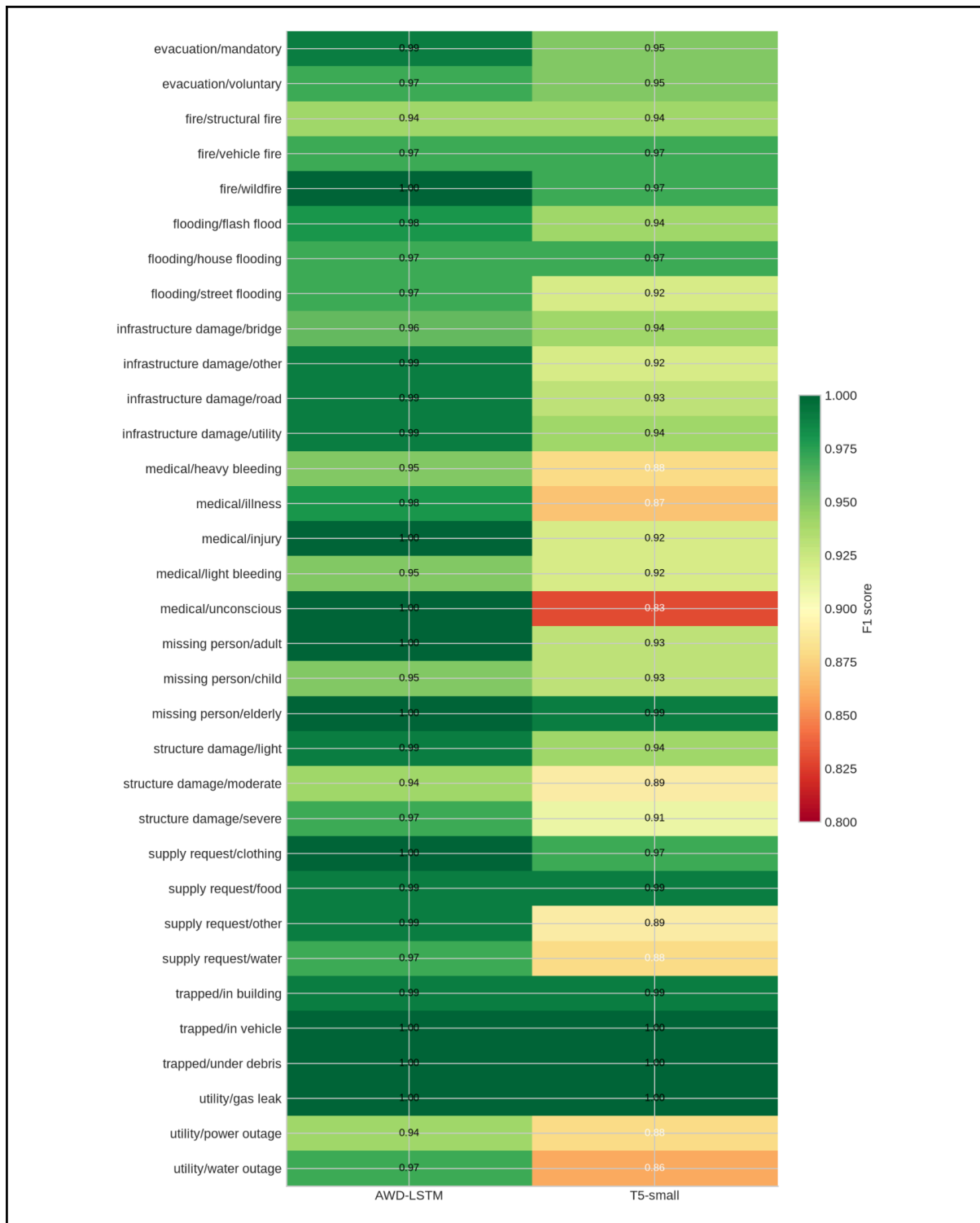


Figure 5.4: Subcategory F1-scores for AWD-LSTM and T5 (small)

Dataset class distribution

Figure 5.5 shows that broad categories are comparatively balanced, while the 33 subcategories have greater frequency variation. This supports the use of macro-averaged metrics in addition to accuracy.

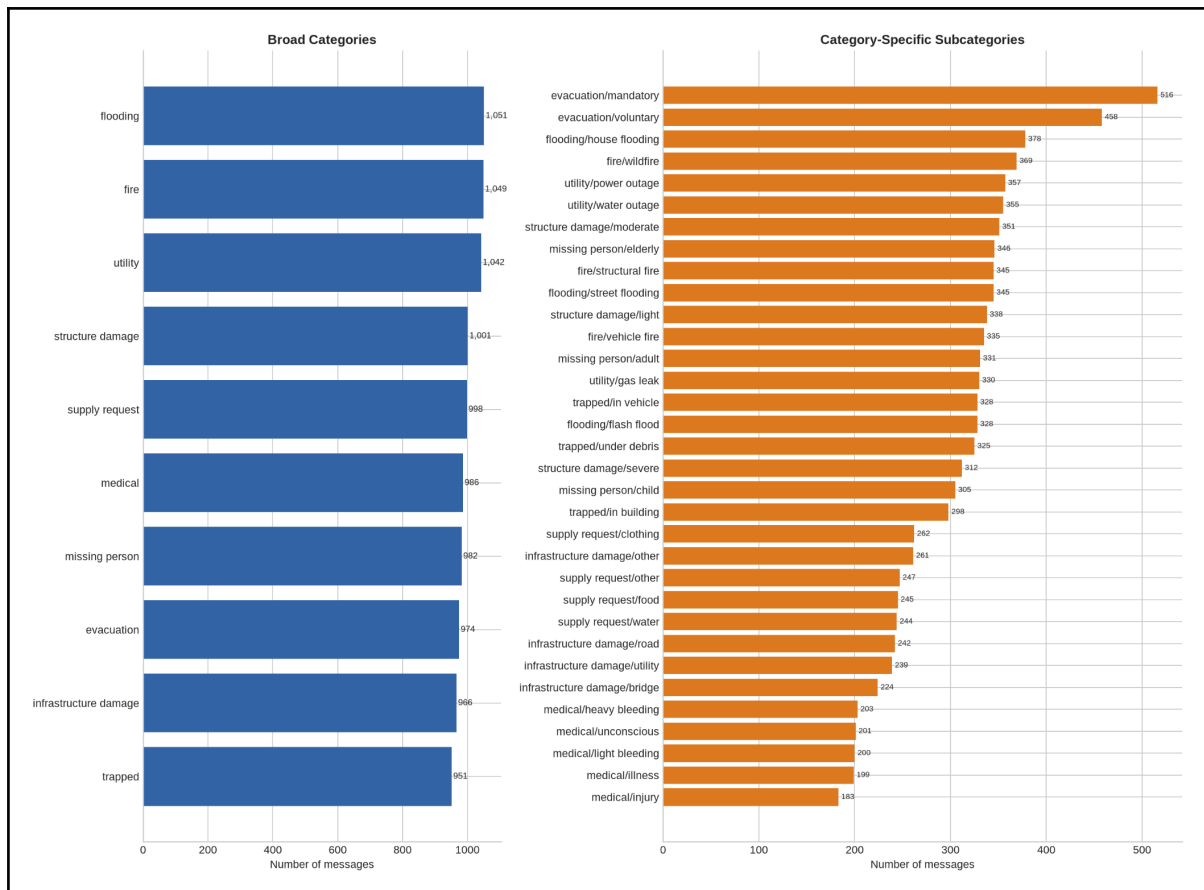


Figure 5.5: Category Distribution

CPU Inference Benchmark

Figure 5.6 shows how end-to-end throughput and mean per-message latency change with batch size on each CPU platform. The logarithmic vertical scales preserve the large differences between models.

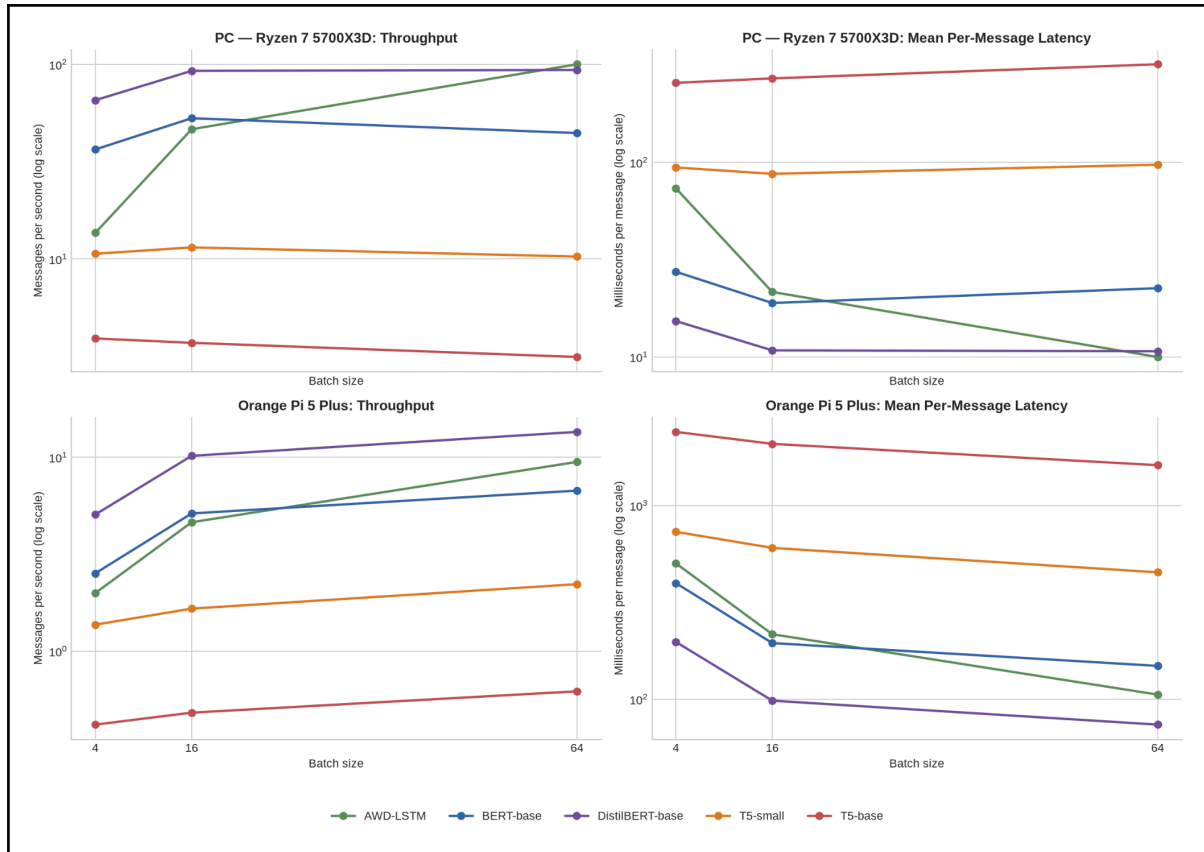


Figure 5.6: End-to-end CPU inference performance

5.3 Discussion

The experimental results demonstrate that the cascading design is highly effective on the current dataset. BERT, DistilBERT, and T5 classified all broad categories correctly and either equalled or exceeded 99.80% end-to-end subcategory accuracy. These results support the central design choice of breaking down the problem into broad categorisation followed by constrained fine-grained subclassification. Once the category is known, limiting the second-stage decision to a valid child class removes invalid alternatives and guarantees a valid category-subcategory output. BERT produced the highest recorded result, but the practical difference between the three leading transformers was extremely small. BERT, T5 and DistilBERT differed by only two correctly classified messages at subcategory-level. On a single 1,500-message test, this is insufficient to claim a stable ordering under changes in random seed or data composition. Repeated runs or cross-validation would be needed to estimate variance, while an independently collected dataset would be needed to assess generalisation beyond the linguistic patterns represented in the current data. DistilBERT provides the most attractive overall deployment trade-off in the recorded experiments. It retained perfect category accuracy and 99.80% subcategory accuracy while using substantially less storage than BERT and T5. At batch size 64 it achieved the highest Orange Pi throughput at 13.52 msg/s and the lowest Orange Pi latency at 73.94 ms per message, while on the PC it was a close second to AWD-LSTM at 93.59 msg/s. BERT remains the strongest choice if the observed maximum accuracy is the sole criterion, but its

two-sample advantage over DistilBERT should be weighed against greater storage and slower batch 64 inference on both systems. T5, which was the largest model pair, did not surpass BERT, and was the slowest model on both platforms. This finding illustrates that increased model size does not automatically improve a task once the dataset is already well separated by a smaller architecture.

AWD-LSTM remains a credible lightweight baseline. Its end-to-end subcategory macro F1-score of 97.84% is strong, and only eight errors occurred beyond those caused by first-stage routing. It combined the smallest saved footprint with the highest batch 64 PC throughput, reaching 100.14 msg/s at 9.99 ms per message. Its Orange Pi result was less competitive than DistilBERT, however, and its subcategory accuracy was 2% lower. AWD-LSTM is therefore attractive for storage-constrained and PC throughput processing, whereas DistilBERT provides a stronger cross-platform balance.

T5 (small) was the only model whose second-stage introduced a large performance reduction. It matched AWD-LSTM at category-level but made 69 additional second-stage errors beyond its 25 category errors. The improvement from T5 (small) to T5 indicates that the generative formulation requires adequate model capacity for this label set, or that the smaller model is more sensitive to the chosen learning rate, decoding score and training duration. Further experiments could test longer training, label description prompts, class-balanced loss weighting and contrastive training between subcategory labels. Based on the current results, however, T5 (small) is not competitive with the other candidates.

Overall, the experiment supports a hierarchy-aware transformer classifier, with DistilBERT providing the most balanced current candidate and BERT providing the highest observed accuracy. Before either model is deployed, evaluation should be extended to independent real-world data, confidence calibration, robustness testing, peak memory and energy measurement and sustained workload testing on the intended hardware.

6. Conclusion, Limitations and Recommendations

6.1 Conclusion

This study investigated whether an AI-based classification model could effectively categorise disaster-related messages and support classification-driven priority assignment within a decentralised emergency communication architecture. To address the research question, five candidate models were developed and evaluated using a two-stage cascading classification approach. The first stage classified messages into one of ten broad emergency categories, while the second stage classified the message into one of 33 corresponding subcategories. A predefined priority level was then derived from the predicted subcategory. The experimental results demonstrated that the proposed classification approach performed strongly on the dataset used within this study. BERT achieved the highest recorded end-to-end classification performance, correctly classifying 1,499 of the 1,500 test messages and achieving a subcategory accuracy of 99.93% and a macro F1-score of 99.90%. T5 and DistilBERT produced similarly strong results, achieving subcategory accuracies of 99.87% and 99.80% respectively. The small difference between these models indicates that classification accuracy alone provides limited evidence for selecting a deployment model when their predictive performance is already very similar. The deployment evaluation demonstrated that model size and computational requirements were also important when selecting a suitable model. DistilBERT provided the strongest overall balance between classification performance, storage requirements, and inference performance. The two-stage DistilBERT model required approximately 512.3 MiB of storage, which was considerably lower than BERT and T5. On the Orange Pi 5 Plus, DistilBERT also achieved the highest measured throughput of 13.52 messages per second and the lowest mean per-message latency of 73.94 ms at batch size 64. These results suggest that DistilBERT is the most suitable candidate from the models evaluated where both predictive performance and deployment constraints are considered.

The cascading classification structure was able to produce valid category-subcategory combinations by restricting the second-stage prediction to subcategories belonging to the predicted parent category. The error analysis also demonstrated an important limitation of this approach. Where an incorrect category was selected during the first-stage, the correct subcategory could no longer be produced because the second-stage options were restricted to the incorrect branch. This was particularly visible for AWD-LSTM and T5 (small), where first-stage routing errors contributed directly to the final classification errors. A confidence or escalation mechanism would therefore be important before such a system could be considered for operational deployment.

The study also proposed a decentralised communication architecture through which the selected classification model could potentially be incorporated into a disaster communication environment. The proposed design combines mesh networking with AI-based message processing so that emergency information could be transmitted, classified, and prioritised when conventional communication infrastructure is damaged. The architecture was developed as a conceptual design rather than implemented as a complete operational communication network. Therefore, the results of this study demonstrate the feasibility of the classification component and its potential suitability for integration, rather than the performance of the complete decentralised system.

Overall, the research question can be answered positively within the scope of the current experiments. The candidate models were able to categorise the disaster-related messages represented within the dataset with high accuracy.

6.2 Limitations

Several limitations should be considered when interpreting the results of this study. The most significant limitation concerns the synthetic dataset. Although duplicate messages were prevented during data generation, the training, validation, and test split was performed after the complete dataset had been generated. Messages appearing in different subsets may therefore contain sentence components derived from the same underlying phrase templates. The test results should consequently be interpreted as measuring generalisation to unseen combinations of the generator's language rather than generalisation to completely independent real-world emergency messages. The survey was introduced to increase the linguistic variation available to the synthetic data generator and reduce reliance on entirely researcher-defined phrases. However, the survey involved only ten participants recruited using convenience sampling. Approximately 100 responses were incorporated into the generation process. While these responses increased the variation of the synthetic messages, they cannot be considered representative of the wider range of language that may be produced during real disaster situations. Messages sent during actual emergencies may contain greater variation in vocabulary, spelling, grammar, emotional state, language ability, and available contextual information. The dataset also assumes that each message can be represented using a single category and subcategory. In a real disaster scenario, an individual message may describe several incidents simultaneously. For example, a message could report structural damage, injuries, and people trapped within the same building. The current cascading classifier is designed to return a single category-subcategory path and would therefore be unable to represent multiple simultaneous classifications without modification.

The experiments were also conducted using a single 70:15:15 dataset split and a fixed random seed of 42. Although this ensured consistent experimental conditions across the candidate models, it does not show how much model performance may change with different data partitions or random initialisation. The extremely small difference between BERT, DistilBERT, and T5 should therefore not be interpreted as establishing a definitive ranking between these models. Repeated experiments using multiple random seeds or cross-validation would provide a stronger estimate of performance variation.

The inference benchmarks also contain several limitations. CPU inference was measured on one desktop PC and one Orange Pi 5 Plus, with five repetitions for each configuration. These experiments provide an indication of deployment performance but do not represent every possible operating condition. Peak memory consumption, energy consumption, temperature, thermal throttling, sustained workloads, and simultaneous requests were not measured. These factors may be particularly important for deployment on low-power or battery-operated hardware within disaster environments.

Finally, the decentralised communication architecture was not implemented and evaluated as a complete operational network. The proposed architecture should be considered a design demonstrating how the classification system could be incorporated into a decentralised emergency communication environment rather than evidence that the complete network would operate effectively during a real disaster.

6.3 Recommendations and Future Work

Future research should initially focus on evaluating the selected model using a different set of synthetic data to separate the underlying phrase templates between the training and test datasets so that test messages are constructed from language components that were not present during training. The current cascading classifier should also be compared against a single flat classifier that directly predicts all 33 subcategories. This would determine whether the two-stage architecture provides a measurable classification advantage or whether its primary benefit is the prevention of structurally invalid category-subcategory combinations. Repeated training using multiple random seeds would also provide a stronger basis for comparing the leading models and determining whether the small differences between BERT, DistilBERT, and T5 remain consistent. Future versions of the classifier could also support multi-label classification so that messages describing several emergencies can be represented simultaneously. This would more closely reflect complex disaster communications where one message may include information about injuries, infrastructure damage, trapped individuals, and supply requirements at the same time.

Further deployment testing should evaluate memory usage, power consumption, thermal performance, and sustained inference workloads on devices such as the Orange Pi 5 Plus. Model optimisation approaches including quantisation could also be investigated to reduce storage and computational requirements while maintaining acceptable classification performance.

Finally, future development should implement and evaluate the proposed decentralised communication architecture. This could include testing mesh routing between multiple physical devices, measuring message delivery time across different numbers of network hops, simulating node failures, and investigating the use of UAV or tethered relay nodes. Integrating the DistilBERT classifier into such a prototype would allow the complete system to be evaluated rather than considering communication and classification as separate components.

7. Reflections

This project provided an opportunity to investigate both AI-based NLP and decentralised communication within the same disaster-response problem. At the beginning of the project, the main focus was the development of a system capable of automatically classifying and prioritising emergency messages. As the research developed, it became clear that classification accuracy alone would not be sufficient for evaluating whether an AI model would be suitable for the intended application. This resulted in the project expanding to consider model size, inference latency, throughput, and the hardware on which the models could potentially operate.

One of the main challenges encountered during the project was the lack of an existing dataset that directly represented the classification structure required by the proposed system. Existing disaster-related datasets identified within the literature frequently focused on social media messages and used different information categories. A synthetic data generator was therefore developed to create labelled messages using the categories, subcategories, and priority structure required by the project. This allowed model development to take place, but also introduced limitations that became clearer as the experiments progressed. The initial synthetic generation process relied heavily on researcher-defined language. This created the possibility that the resulting dataset would represent only a narrow range of ways in which an emergency might be described. The open-question survey was introduced to address this problem by collecting additional phrases and sentence structures from participants. This improved the linguistic variation available to the generator and also demonstrated the importance of considering dataset construction as part of the development process rather than treating the dataset as a separate component.

The T5 experiments were particularly useful in demonstrating why identical optimisation parameters cannot always be transferred directly between different model architectures. The T5 family performed worse using the initial BERT settings, requiring adjustments to the learning rate and gradient accumulation before producing a more suitable configuration. This reinforced the importance of understanding the characteristics of each architecture rather than assuming that a single training configuration will perform equally well across different models.

The development of the decentralised communication architecture also broadened the scope of the project beyond model training. This reinforced the wider objective of the project: AI classification should be considered as one component within a complete emergency communication system rather than as an isolated model.

There were also areas where the project could have been approached differently. With additional time, the dataset design would have separated sentence templates before creating the training and test subsets. This would have produced a stronger evaluation of language generalisation. A flat 33-class classifier could also have been implemented alongside the cascading design to directly compare the two approaches. Repeated training using multiple random seeds would have provided greater confidence when comparing models whose final performance differed by only a small number of messages.

Overall, the project developed my understanding of the complete AI development process. It also demonstrated the importance of critically interpreting high performance metrics. The near-perfect results achieved by several models are encouraging within the current dataset, but they do not automatically demonstrate equivalent performance on real-world emergency communications. Recognising this distinction became one of the most important outcomes of the project and provides a clear direction for future development of the proposed system.

References

- [1] United Nations Office for Disaster Risk Reduction, "Analytical estimate of damages caused by the June 24, 2026 earthquakes in Venezuela," Accessed: Jul. 2026. [Online]. Available: <https://www.undrr.org/publication/documents-and-publications/analytical-estimate-damages-caused-june-24-2026-earthquakes>
- [2] Z. EL Khaled and H. Mcheick, "Case studies of communications systems during harsh environments: A review of approaches, weaknesses, and limitations to improve quality of service," *International Journal of Distributed Sensor Networks*, vol. 15, no. 2, p. 155014771982996, Feb. 2019, doi: 10.1177/1550147719829960.
- [3] D. G.C., A. Ladas, Y. A. Sambo, H. Pervaiz, C. Politis, and M. A. Imran, "An Overview of Post-Disaster Emergency Communication Systems in the Future Networks," *IEEE Wireless Communications*, vol. 26, no. 6, pp. 132–139, Dec. 2019, doi: 10.1109/mwc.2019.1800467.
- [4] B. Karaman et al., "Solutions for Sustainable and Resilient Communication Infrastructure in Disaster Relief and Management Scenarios," *IEEE Communications Surveys & Tutorials*, pp. 1–1, Jan. 2025, doi: 10.1109/comst.2025.3610793.
- [5] U. Ashraf, A. Khwaja, J. Qadir, S. Avallone, and C. Yuen, "WiMesh: leveraging mesh networking for disaster communication in resource-constrained settings," *Wireless Networks*, vol. 27, no. 4, pp. 2785–2812, Apr. 2021, doi: 10.1007/s11276-021-02621-2.
- [6] J. Z. Moghaddam, M. Usman, and F. Granelli, "A Device-to-Device Communication-Based Disaster Response Network," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 2, pp. 288–298, June 2018, doi: 10.1109/tccn.2018.2801339.
- [7] L. Yu, H. Yang, L. Miao, and C. Zhang, "Rollout algorithms for resource allocation in humanitarian logistics," *IIEE Transactions*, vol. 51, no. 8, pp. 887–909, Mar. 2018, doi: 10.1080/24725854.2017.1417655.
- [8] L. Dwarakanath, A. Kamsin, R. A. Rasheed, A. Anandhan, and L. Shuib, "Automated Machine Learning Approaches for Emergency Response and Coordination via Social Media in the Aftermath of a Disaster: A Review," *IEEE Access*, vol. 9, pp. 68917–68931, 2021, doi: 10.1109/access.2021.3074819.
- [9] T. Kedia et al., "Technologies enabling situational awareness during disaster response: A systematic review," *Disaster Medicine and Public Health Preparedness*, vol. 16, no. 1, pp. 1–19, 2022, doi: 10.1017/dmp.2020.196.
- [10] A. Raj, A. Shetgaonkar, L. Arora, D. Pradhan, S. S. Girija, and S. Kapoor, "AI and Generative AI Transforming Disaster Management: A Survey of Damage Assessment and Response Techniques," *2025 IEEE 49th Annual Computers, Software, and Applications Conference (COMPSAC)*, pp. 1834–1840, July 2025, doi: 10.1109/compsac65507.2025.00251.
- [11] M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned Aerial Vehicles for Search and Rescue: A Survey," *Remote Sensing*, vol. 15, no. 13, pp. 3266–3266, June 2023, doi: 10.3390/rs15133266.
- [12] G. Kovács and K. M. Spens, "Humanitarian logistics in disaster relief operations," *International Journal of Physical Distribution & Logistics Management*, vol. 37, no. 2, pp. 99–114, Mar. 2007, doi: 10.1108/09600030710734820.
- [13] P. J. J. C. P. and G. Mahalakshmi, "Medico Assistant: An AI-Driven Real-Time Multi-Modal Triage System," *2025 2nd International Conference on Artificial Intelligence and*

- Knowledge Discovery in Concurrent Engineering (ICECONF), pp. 1–7, Oct. 2025, doi: 10.1109/iceconf65644.2025.11379660.
- [14] K. T. A. P. Shijish, K. A. M. S., and R. Rajeev, "ResQlink: An Emergency Response WebApp," 2025 5th International Conference on Intelligent Technologies (CONIT), pp. 1–5, June 2025, doi: 10.1109/conit65521.2025.11167160.
- [15] W. Tan, Y. Zhang, and K. Liu, "Intelligent Emergency Notification Mobile Service via Multi-task BERT Models," 2023 11th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 51–58, July 2023, doi: 10.1109/mobilecloud58788.2023.00014.
- [16] M. Raza et al., "Establishing effective communications in disaster affected areas and artificial intelligence based detection using social media platform," *Future Generation Computer Systems*, vol. 112, pp. 1057–1069, Nov. 2020, doi: 10.1016/j.future.2020.06.040.
- [17] IEEE, "Artificial Intelligence," IEEE Technology Navigator, 2022. <https://technav.ieee.org/topic/artificial-intelligence/> (accessed Aug, 2026).
- [18] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed. Hoboken, NJ, USA: Pearson, 2021.
- [19] W. Wang, Y. Yang, and F. Wu, "Towards Data-And Knowledge-Driven AI: A Survey on Neuro-Symbolic Computing," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 2, pp. 878–899, 2025. doi: 10.1109/TPAMI.2024.3483273.
- [20] D. A. Waterman, *A Guide to Expert Systems*. Addison Wesley Publishing Company, 1986.
- [21] D. Partridge, "The scope and limitations of first generation expert systems," *Future Generation Computer Systems*, vol. 3, no. 1, pp. 1–10, Feb. 1987, doi: 10.1016/0167-739x(87)90038-0.
- [22] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, July 2015, doi: 10.1126/science.aaa8415.
- [23] T. Mitchell, *Machine Learning*. London: McGraw-Hill Pub. Co. (Ise Editions), 1997.
- [24] K. P. Murphy, *Probabilistic machine learning : an introduction*. Cambridge, Massachusetts: The Mit Press, 2022.
- [25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: The MIT Press, 2016. Accessed: Aug. 2026. [Online]. Available: <https://www.deeplearningbook.org/>
- [26] X.-W. Chen and X. Lin, "Big Data Deep Learning: Challenges and Perspectives," *IEEE Access*, vol. 2, pp. 514–525, 2014, doi: 10.1109/access.2014.2325029.
- [27] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, no. 61, pp. 85–117, Jan. 2015, doi: 10.1016/j.neunet.2014.09.003.
- [28] L. Deng and D. Yu, "Deep Learning: Methods and Applications," *Foundations and Trends® in Signal Processing*, vol. 7, no. 3–4, pp. 197–387, 2014, doi: 10.1561/20000000039.
- [29] D. Yu and J. Li, "Recent progresses in deep learning based acoustic models," *IEEE/CAA Journal of Automatica Sinica*, vol. 4, no. 3, pp. 396–409, 2017, doi: 10.1109/jas.2017.7510508.
- [30] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.
- [31] D. Jurafsky and J. H. Martin, "Speech and Language Processing," *Stanford.edu*, 2025. <https://web.stanford.edu/~jurafsky/slp3/> (accessed Aug, 2026).

- [32] T. Mikolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," *Interspeech 2010*, Sept. 2010, doi: 10.21437/interspeech.2010-343.
- [33] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997, doi: 10.1162/neco.1997.9.8.1735.
- [34] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, Mar. 1994, doi: 10.1109/72.279181.
- [35] A. Graves, *Supervised Sequence Labelling with Recurrent Neural Networks*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. doi: 10.1007/978-3-642-24797-2.
- [36] A. Vaswani et al., "Attention Is All You Need," June 2017. Accessed: Aug. 2026. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [37] C. D. Manning and Hinrich Schütze, *Foundations of Statistical Natural Language Processing*. Cambridge, Mass.: MIT Press, 2000.
- [38] E. Cambria and B. White, "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]," *IEEE Computational Intelligence Magazine*, vol. 9, no. 2, pp. 48–57, May 2014, doi: 10.1109/mci.2014.2307227.
- [39] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Advances in Neural Information Processing Systems 26*, 2013, Accessed: Aug. 2026. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2013/hash/9aa42b31882ec039965f3c4923ce901b-Abstract.html
- [40] J. Pennington, R. Socher, and C. Manning, "Glove: Global Vectors for Word Representation," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.
- [41] M. Peters et al., "Deep Contextualized Word Representations," *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, vol. 1, 2018, doi: 10.18653/v1/n18-1202.
- [42] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *Proceedings of the 2019 Conference of the North*, vol. 1, pp. 4171–4186, 2019, doi: 10.18653/v1/n19-1423.
- [43] D. Merks and S. L. Frank, "Human Sentence Processing: Recurrence or Attention?," *Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics*, pp. 12–22, 2021, doi: 10.18653/v1/2021.cmcl-1.2.
- [44] Q. Fournier, Gaétan Marceau Caron, and D. Aloise, "A Practical Survey on Faster and Lighter Transformers," *ACM Computing Surveys*, vol. 55, no. 14s, pp. 1–40, July 2023, doi: 10.1145/3586074.
- [45] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," Dec. 2014. doi: 10.48550/arXiv.1409.3215.
- [46] A. Galassi, M. Lippi, and P. Torroni, "Attention in Natural Language Processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 1–18, 2020, doi: 10.1109/tnnls.2020.3019893.
- [47] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, vol. 3, Oct. 2022, doi: 10.1016/j.aiopen.2022.10.001.
- [48] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter." Oct. 02, 2019. Accessed: Aug. 2026. [Online]. Available: <https://arxiv.org/abs/1910.01108v4>

- [49] C. Raffel et al., "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020, Accessed: Aug. 2026. [Online]. Available: <https://www.jmlr.org/papers/volume21/20-074/20-074.pdf>
- [50] I. Tenney, D. Das, and E. Pavlick, "BERT Rediscovered the Classical NLP Pipeline," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, doi: 10.18653/v1/p19-1452.
- [51] X. Jiao et al., "TinyBERT: Distilling BERT for Natural Language Understanding," *Empirical Methods in Natural Language Processing*, Nov. 2020, doi: 10.18653/v1/2020.findings-emnlp.372.
- [52] M. Kejriwal, G. Fang, and Y. Zhou, "A Feasibility Study of Open-Source Sentiment Analysis and Text Classification Systems on Disaster-Specific Social Media Data," *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, Dec. 2021, doi: 10.1109/ssci50451.2021.9660089.
- [53] I. Ullah, A. Jamil, I. U. Hassan, and B.-S. Kim, "Unveiling the Power of Deep Learning: A Comparative Study of LSTM, BERT, and GRU for Disaster Tweet Classification," *IEIE Transactions on Smart Processing & Computing*, vol. 12, no. 6, pp. 526–534, Dec. 2023, doi: 10.5573/ieiespc.2023.12.6.526.
- [54] D. S. Krishna, G. Srinivas, and P.V.G.D. Prasad Reddy, "A deep parallel hybrid fusion model for disaster tweet classification on Twitter data," *Decision analytics journal*, vol. 11, pp. 100453–100453, June 2024, doi: 10.1016/j.dajour.2024.100453.
- [55] F. Alam, U. Qazi, M. Imran, and F. Ofli, "HumAID: Human-Annotated Disaster Incidents Data from Twitter with Deep Learning Benchmarks," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15, pp. 933–942, May 2021, doi: 10.1609/icwsm.v15i1.18116.
- [56] S. Yicheng, S. A. Louw, Y. M. Thant, A. Arya, R. T. Konduru, and R. Avtar, "BART-Disaster-X: an effective model to extract information from disaster-related tweets on X," *Remote Sensing Applications: Society and Environment*, vol. 43, p. 102174, Aug. 2026, doi: 10.1016/j.rsase.2026.102174.
- [57] R. McCreddie, C. Buntain, and I. Soboroff, "TREC Incident Streams: Finding Actionable Information on Social Media." Accessed: Aug. 2026. [Online]. Available: https://www.dcs.gla.ac.uk/~richardm/TREC_IS/2018/Papers/TREC_IS_2018_ISCRAM.pdf
- [58] C. Wang and D. Lillis, "Multi-task Transfer Learning for Finding Actionable Information from Crisis-related Messages on Social Media," 2020, doi: 10.6028/nist.sp.1266.incident-ucd-cs.
- [59] E. Boros, N. Nguyen, G. Lejeune, M. Coustaty, and A. Doucet, "Transformer-based Methods with #Entities for Detecting Emergency Events on Social Media." Accessed: Aug. 2026. [Online]. Available: https://trec.nist.gov/pubs/trec30/papers/L3i_Rochelle-IS.pdf
- [60] J. W. Creswell, *Research design : Qualitative, Quantitative and Mixed Methods Approaches.*, 3rd ed. Sage Publications, Inc, 2009. Accessed: Aug. 2026. [Online]. Available: https://www.ucg.ac.me/skladiste/blog_609332/objava_105202/fajlovi/Creswell.pdf
- [61] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [62] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," *arXiv (Cornell University)*, Nov. 2018, doi: 10.48550/arxiv.1811.12808.
- [63] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, Oct. 1998, doi: 10.1162/089976698300017197.
- [64] A. Hevner, S. March, J. Park, and S. Ram, "Design Science in Information Systems Research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004, doi: 10.2307/25148625.

- [65] A. J. Nashwan and H. Abukhadijah, "Harnessing Artificial Intelligence for Qualitative and Mixed Methods in Nursing Research," *Cureus*, Nov. 2023, doi: 10.7759/cureus.48570.
- [66] J. K. Fieldhouse et al., "The intersection of artificial intelligence with qualitative or mixed methods for communicable disease research: a scoping review," *Public Health*, vol. 248, p. 105961, Nov. 2025, doi: 10.1016/j.puhe.2025.105961.
- [67] M. Imran, P. Mitra, and C. Castillo, "Twitter as a Lifeline: Human-annotated Twitter Corpora for NLP of Crisis-related Messages," *ACL Anthology*, pp. 1638–1643, May 2016, Accessed: Aug. 2026. [Online]. Available: <https://aclanthology.org/L16-1259/>
- [68] F. Alam, H. Sajjad, M. Imran, and F. Ofli, "CrisisBench: Benchmarking Crisis-related Social Media Datasets for Humanitarian Information Processing," *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 15, pp. 923–932, May 2021, doi: 10.1609/icwsm.v15i1.18115.
- [69] C. Wang and D. Lillis, "Classification for Crisis-Related Tweets Leveraging Word Embeddings and Data Augmentation," 2019, doi: 10.6028/nist.sp.1250.incident-cs-ucd.
- [70] D. M. W. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
- [71] V. J. Reddi et al., "MLPerf Inference Benchmark," May 2020. doi: 10.1109/ISCA45697.2020.00045.
- [72] T. Wu, M. T. Ribeiro, J. Heer, and D. Weld, "Errudite: Scalable, Reproducible, and Testable Error Analysis," *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 747–763, 2019, doi: 10.18653/v1/p19-1073.
- [73] M. S. Setia, "Methodology Series Module 3: Cross-sectional Studies," *Indian Journal of Dermatology*, vol. 61, no. 3, pp. 261–264, 2016, doi: 10.4103/0019-5154.182410.
- [74] H. Suresh and J. Guttag, "A Framework for Understanding Sources of Harm throughout the Machine Learning Life Cycle," *Equity and Access in Algorithms, Mechanisms, and Optimization*, Oct. 2021, doi: 10.1145/3465416.3483305.
- [75] M. Guan, Z. Feng, S. Jiang, and W. Zhou, "Tethered Balloon Cluster Deployments and Optimization for Emergency Communication Networks," *Entropy*, vol. 26, no. 12, p. 1071, Dec. 2024, doi: 10.3390/e26121071.

Appendices

Appendix 1: Source code and Notebooks:

<https://github.com/z-evans/emergency-classifier>

Appendix 2: Open question survey

Open-Question Emergency Message Survey

Researcher	Zack Evans
Programme	MSc Software Engineering and Artificial Intelligence
Purpose	Collection of natural-language emergency message examples for synthetic dataset generation

Participant Information

This survey collects examples of how people may naturally communicate during an emergency or disaster scenario. The responses will be used to introduce a wider range of vocabulary, sentence structures, and writing styles into a synthetic dataset used to train and evaluate natural language processing classification models. The survey is not intended to measure your opinions, knowledge, or ability to identify emergency priorities.

You will be shown a series of short emergency scenarios. For each scenario, write the message that you might realistically send if you were experiencing or reporting that situation. Please use your own wording. Short messages, abbreviations, informal language, spelling mistakes, or incomplete sentences are acceptable if that is how you would naturally communicate. You do not need to answer every question and may leave any scenario blank.

No demographic information is requested. Your responses will be reviewed for useful phrases and ways of describing emergencies; they are not intended to form a standalone training dataset. Do not include real names, addresses, telephone numbers, or other personally identifying information in your answers.

Participant Consent

- ☐ I have read the participant information above and understand the purpose of the survey.
- ☐ I understand that participation is voluntary and that I may leave any question unanswered.
- ☐ I consent to my written responses being used to support the research dataset generation process described above.

Participant signature:

Signed Date:

Medical emergencies

1. A person has been injured and needs help.

2. A person has become unwell and may require medical assistance.

3. A person is bleeding heavily and the bleeding appears serious.

4. A person has a smaller or less severe bleeding injury.

5. A person is not responding and cannot be woken.

Fire emergencies

6. A fire is spreading through grass, woodland, or another outdoor area.

7. A building or other structure is on fire.

8. A vehicle is on fire.

Flooding

9. Water has risen very quickly and is creating an immediate flood danger.

10. Flood water is entering or has entered a home.

11. A road is flooded and travel is affected.

Utility failures

12. Power has been lost in the area.

13. The normal water supply has stopped or is unavailable.

14. There may be a gas leak nearby.

Missing persons

15. A child is missing and cannot be located.

16. An adult is missing and cannot be located.

17. An elderly person is missing and cannot be located.

Supply requests

18. You or a group of people need food supplies.

19. You or a group of people need drinking water.

20. You or a group of people need clothing.

21. You need another essential supply that is not food, water, or clothing.

Evacuation

22. People have been ordered to leave an area and evacuation is mandatory.

23. People are being advised to leave, or are choosing to evacuate, even though evacuation is not mandatory.

Structural damage

24. A building or structure has suffered light damage but remains mostly intact.

25. A building or structure has suffered noticeable damage that may affect its safety or use.

26. A building or structure has suffered severe damage and may be unsafe or at risk of collapse.

Infrastructure damage

27. A road has been damaged and access or travel is affected.

28. A bridge has been damaged and may be unsafe or unusable.

29. Utility infrastructure, such as power, water, gas, or communications equipment, has been physically damaged.

30. Another type of public or community infrastructure has been damaged.

Trapped persons

31. A person is trapped inside a building and cannot get out.

32. A person is trapped inside a vehicle and cannot get out.

33. A person is trapped under debris or rubble.
