Text classification on disaster tweets

Natural Language Processing - MC1 - HS 2023

Jan Zwicky, Gabriel Torres Gamez, Florin Barbisch

Inhaltsverzeichnis

1 Introduction

1.1 Objective

The objective of this report is to assess the performance of three distinct machine-learning models developed for the classification of Twitter messages into disaster-related or non-disaster categories. Through selecting a variety of models, we aim to explore the strengths and weaknesses inherent in different approaches to text classification. We focus on a deep learning-based method among the three to leverage recent advances in the field. A comprehensive error analysis, coupled with careful examination of individual predictions, is carried out to identify potential improvements and to understand the nuances of each model's performance in this particular task.

1.2 Background

Twitter, with its real-time posting capabilities, has emerged as a vital tool for communicating during emergencies. The ubiquity of smartphones allows individuals to report incidents as they happen, which has garnered the interest of agencies like disaster relief organizations and news outlets in monitoring Twitter feeds. However, the challenge remains in discerning whether tweets genuinely report on disasters amid the massive volume of data generated on the platform.

1.3 Scope of the Report

This report delves into the comparative analysis of three different machine-learning methods applied to the classification of disaster-related tweets. The chosen systems – TF-IDF with Histogram-based Gradient Boosting Classifier (HGBC) (and comparison to a Support Vector Machine (SVM) as a replacement for HGBC), BERTweet, and a Convolutional Neural Network (CNN) – are evaluated based on their theoretical underpinnings and practical application relevance to text classification challenges. The evaluation process involves the selection of a specific metric, justified by its suitability to the context of the classification task at hand. Through the lens of this metric, the performance of each model is scrutinized, revealing insights into particular strengths, weaknesses, and discrepancies in results. The overarching questions this report aims to address pertain to model-specific efficacies, universal shortcomings, overall best practices, differential outcomes, and the learning opportunities that can be gleaned from this exercise. The scope is thus circumscribed to the end of gaining a thorough understanding of how each model handles the classification task and what can be inferred from their respective performances in pinpointing disaster tweets accurately.

2 Machine Learning Methods

The field of Natural Language Processing (NLP) has experienced significant advancements in recent years, and one area of NLP is text classification, which involves assigning predefined categories to text. Twitter, a platform teeming with real-time data, has become a rich source for analyzing various kinds of text, including those related to disasters. Identifying disaster-related tweets can be pivotal for emergency response and relief efforts. To tackle this classification problem, we have chosen a range of machine learning methods, each promising in the domain of text analysis.

We first focus on a Convolutional Neural Network (CNN). CNNs are traditionally associated with image processing, but they have also proven effective in NLP tasks. The recommendation by Fernando Benites to utilize a CNN for text classification intrigued us because it suggests the model's applicability in capturing local patterns within text data. This model's strength lies in its ability to detect patterns over different parts of the data sequences, which could be beneficial for identifying key features within tweets.

Next, we consider BERTweet, a pre-trained language model specifically trained on English tweets. Its use in our study stems from the desire to leverage an advanced transformer model capable of capturing context and the nuances of language used on Twitter. Transformers have redefined the benchmarks for numerous NLP tasks, making BERTweet a state-of-the-art choice for our experiment.

Moreover, we have chosen to implement a model combining Term Frequency-Inverse Document Frequency (TF-IDF) with Support Vector Machines (SVM), again based on the advice from Fernando Benites. TF-IDF with SVM is a classic combination in text classification scenarios, mainly due to the SVM's robustness and effectiveness in handling high-dimensional text data.

Lastly, our selection includes TF-IDF combined with Histogram-based Gradient Boosting Classifier (HGBC). Our choice is rooted in the positive outcomes from previous machine learning projects that used HGBC. Given its success in other domains, we aim to gauge its performance when dealing with the classification of disaster-related content on Twitter. HGBC's capability for handling large datasets efficiently and tackling non-linear patterns makes it a suitable candidate for our task.

To sum up, the selection of these methods stems from a blend of their theoretical foundations, proven efficiency in prior applications, and expert recommendations. These diverse approaches will be assessed for their performance on the classification challenge, providing insights into the intricacies of disaster tweet identification and the strengths of different models in such a context.

2.1 Method 1: TF-IDF with HGBC

The approach to classifying disaster-related tweets with the combination of TF-IDF Vectorization and Histogram-based Gradient Boosting Classifier (HGBC) is a compelling method due to its relative simplicity and efficiency. The methodology hinges on the transformation of textual data into a numerical form that can be processed by machine learning algorithms, followed by a classification system that predicts the nature of the tweets.

At the core of this system is the TF-IDF Vectorizer, which serves as a bridge between the raw textual data and the numerical features required for machine learning. By calculating the importance of each word within a tweet relative to the entire corpus, the Vectorizer assigns weights that reflect both the word's frequency in the tweet and its rarity across all tweets. Consequently, it can represent tweets as vectors in a multi-dimensional space, where the proximity of the vectors corresponds to the similarity in their content.

The HGBC then takes these vectors and learns to differentiate between disaster and non-disaster tweets. Leveraging the efficiency of histograms, the HGBC approximates the gradient boosting process, making it faster and more scalable than traditional gradient boosting methods. The classifier effectively identifies patterns within the vectorized tweets that are indicative of disaster-related content and utilizes these patterns to make predictions on new data.

In the training phase, it is crucial to manage the dimensionality of the TF-IDF vectors to avoid overfitting and to enhance model generalization. A common strategy is to reduce dimensionality, but our experiments have revealed that this can be counterproductive, leading to slower training times and suboptimal performance. As an alternative, we limit the scope of our Vectorizer to the 1000 most significant words. This technique not only simplifies the model but surprisingly maintains, if not improves, its predictive accuracy. By focusing on only the most relevant words, the model is less inclined to overfit on minor textual details.

The decision to exclude NGRAMs was made after determining that their inclusion did not materially benefit the model's performance but instead substantially increased computational requirements. Our observation suggests that single-word features sufficiently capture the necessary information for tweet classification in this context, thus the additional complexity of NGRAMs is unnecessary. This indicates that a number of buzzwords is enough to distinguish disaster tweets from non-disaster tweets, which is consistent with the nature of Twitter as a platform for short, concise messages.

In sum, our method of using TF-IDF Vectorization with HGBC demonstrates both effectiveness and computational prudence, making it an attractive option for classifying disaster tweets. The strategic choices in model construction, such as vector dimensionality and feature selection, play a pivotal role in shaping the model's performance and efficiency. We chose this model as our baseline because of its simplicity.

2.1.1 Results

The TF-IDF with HGBC model exhibited compelling results with high accuracy and F1 macro scores on the training dataset, achieving 0.888 and 0.883, respectively. However, the model demonstrated a susceptibility to

second-degree errors, where it struggled to classify some disaster tweets accurately. This complication arose particularly when judging tweets by their content alone, without the availability of external context such as embedded links or conversational threads.

Examples of such cases were where sarcastic or off-topic replies were incorrectly labeled as disasters due to the presence of keywords commonly associated with catastrophic events or overly dramatic language. Moreover, this set of misclassified tweets presented a challenge for the model, underscoring the intricacy of text classification that extends beyond mere keyword recognition.

When subjecting the model to validation data, the accuracy and F1 scores fell to 0.789 and 0.777, respectively. Nonetheless, these results remain praiseworthy for a validation dataset, hinting at the model's robust generalization capabilities. Second-degree errors were less frequent in this setting, though the absence of broader conversational context in certain replies still posed classification challenges.

First-degree errors were also observed, where the model misinterpreted the gravity of specific terms within a tweet. For example, „Crash“ in an educational context or „Storm“ in casual conversation were instances where the model faltered, selecting the disaster category when it was unwarranted.

The insight into feature relevance was gained through Permutation Importance, revealing that certain words, after stemming and lemmatization, were pivotal in decision-making. The direct feature relevance incorporating the IDF values confirmed the strong influence of disaster-related words on classification outcomes. However, the significance of non-disaster-specific words like „you,“ „in,“ and „i“ highlighted a potential disproportion in the dataset's size or composition.

Further analysis revealed that if a sentence included words with strong disaster relevance, the propensity for it to be categorized as a disaster tweet increased, potentially overshadowing the context in which these words were used. This detection raises concerns about the model's sensitivity to such terms and the need for more nuanced feature engineering to improve classification accuracy without relying overtly on specific keywords. Moreover, the disproportionate weighting on common words suggests potential overrepresentation of particular expressions in the dataset, which may need to be addressed through further data balancing or preprocessing measures.

2.1.2 Compare with SVM

The employment of Support Vector Machine (SVM) classifier paired with the TF-IDF Vectorizer presented an insightful comparison against the previously employed Histogram-based Gradient Boosting Classifier (HGBC). With metrics showcasing an accuracy of approximately 0.807 and an F1 macro score of nearly 0.794, the SVM model demonstrated a competitive, if not superior, performance relative to its HGBC counterpart.

The SVM algorithm operates on the principle of identifying the hyperplane that best separates different classes in a high-dimensional space. By maximizing the margin between support vectors of the different classes, the SVM classifier is renowned for its effectiveness in classification tasks, particularly in text classification scenarios where feature spaces are typically vast and sparse.

Notably, the SVM model maintained the transformation step, using the same TF-IDF Vectorization to generate the feature space. This continuity allowed for a direct and fair performance comparison between the HGBC and SVM models. While both models were susceptible to the same first and second-degree errors, likely due to inherent challenges in the dataset and the nature of text classification, the SVM model exhibited a slight edge in overall accuracy and F1 macro score.

The similarity in error patterns between both models suggests that these issues are less likely to be tied to the classification algorithms themselves and more so to the feature representation or the intrinsic properties of the training data. In both cases, classifier misjudgment seemed to endure, predominantly where tweets contained keywords typically linked to disasters but used in a different context or with a tone that the algorithm could not discern, such as sarcasm or metaphor.

It's intriguing to note that the SVM, while inherently different in methodology from the ensemble HGBC, could match and even exceed the latter's predictive capabilities. This congruence in error types yet divergence

in effectiveness implies that the SVM could be better attuned for handling the nuances of text classification, particularly in the domain of disaster tweet analysis.

In conclusion, the SVM's comparative success in this use case prompts consideration for its adoption, given its relatively straightforward implementation, computational efficiency, and notable accuracy.

2.2 Model 2: BERTweet

BERT, short for „Bidirectional Encoder Representations from Transformers", represents a paradigm shift in natural language processing (NLP), developed by Google AI Language. The transformative feature of BERT is its ability to understand language context bidirectionally, meaning it can consider the context of each word in a dataset from left to right and vice versa. BERT utilizes the Transformer, an attention mechanism structure that captures the relationship between all words in a sentence, as opposed to previous models that analyzed words sequentially.

Applications of BERT

BERT is designed to generate representations that can support a variety of NLP tasks, such as:

- Text classification
- Question answering
- Named entity recognition
- Text summarization
- Language translation
- Text similarity analysis
- It is important to note that BERT in its basic form is not suitable for text generation tasks, as it does not include a dedicated decoder component.

Specific Implementation: BERTweet

BERTweet is a BERT variant specifically developed for the analysis of English-language Twitter data. It is based on RoBERTa, a modification of BERT that surpasses BERT by optimizing some hyperparameters and using a different preprocessing method focused on dynamic masking. BERTweet understands tweets by recognizing language patterns from a massive dataset, making it an excellent candidate for classifying tweets in terms of catastrophic events.

The BERTweet model is structured as follows:

Embeddings: This area is responsible for converting words, positions, and token types into vectors. It uses word embeddings of size 64,001x768, position embeddings of size 130x768, and token-type embeddings of size 1x768. Layer normalization and dropout layers are used for model stability and to prevent overfitting.

Encoder: This consists of a list of 12 RobertaLayer modules. Each RobertaLayer has an Attention layer (RobertaAttention), which in turn is composed of Query, Key, and Value layers. This is followed by an intermediate step (RobertaIntermediate) with a linear layer that expands the dimensions from 768 to 3072, and a GELU activation function. Finally, the result is reduced back to 768 dimensions by another linear layer (RobertaOutput).

The model and various useful functions, such as the tokenizer, dataset-specific preprocessing methods, and the initial model architecture, were implemented and adapted using the transformers library by Hugging Face. Due to its pre-trained state on a large text corpus, BERTweet could be efficiently fine-tuned from general language understanding for the specific task of classifying tweets regarding their relevance to natural disasters.

2.2.1 Results

The training process of the BERTweet model reveals several key findings regarding its ability to classify disaster-related tweets. Notably, there was a rapid decline in the training loss during initial epochs, which levels off as

training progressed. This suggests the model quickly learned patterns within the training data. In contrast, the evaluation loss followed a different trajectory. After an initial decrease, the evaluation loss began to fluctuate, indicating difficulties in generalizing learnings to the test data, as evidenced by the absence of a sustained downward trend in this metric.

Stability in the evaluation was observed via both the accuracy and F1 score, which hovered around 0.75. These metrics did not show significant improvement over time, signifying a level of consistency in the model's performance regarding its classification abilities. However, given BERTweet's sophisticated architecture, expectations for this model were set slightly higher.

The potential overfitting issue was hinted at by the considerable divergence between the rapidly decreasing training loss and the fluctuating pattern of the evaluation loss. This discrepancy raised concerns about the model's generalization capabilities, as it appeared to perform better on training data than on unseen evaluation data.

Assessment results based on validation data reflected an accuracy rate close to 80% and an F1 score marginally above 77%, which, under ordinary circumstances, are marks of respectable performance for a text classification task. Yet for BERTweet, a model that is part of the advanced BERT family, such results were somewhat underwhelming, as higher performance was anticipated.

The confusion matrix displayed a relatively balanced classification ability for both disaster and non-disaster tweets, with a comparable rate of misclassifications observed between the two categories. It sheds light on the types of errors made, suggesting possible biases or deficiencies in the model's understanding of contextual cues associated with disaster-related content.

In incorrectly classified instances, we noted that formal or bureaucratic expressions characterized many of the false negatives. These encompassed specific jargon and abbreviations uncommon in day-to-day language. Difficulty interpreting terms and phrases exclusive to the dataset, particularly if they were not adequately represented during the training phase, might be attributable to the misclassifications.

Further, the model's capability to generalize across a varying range of disaster topics came to question, potentially affected by a limited representation of each category within the training set. Also, the presence of URLs and special characters in tweets possibly introduced additional noise, posing challenges to the model's predictive accuracy.

On the flip side, the false positives often encapsulated tweets with ironic tones or colloquial language, which the model might have taken at face value. Keywords typically associated with disasters could have biased the model, leading it to classify a tweet as disaster-related without thoroughly appreciating the wider context. Furthermore, tweets conveying ambiguous and terse sentiments presented inherent difficulties in classification accuracy.

To counterbalance these shortcomings and improve classification accuracy, future work should consider strategies that foster a deeper understanding of context. One might refine the model using a training set enriched with nuanced language structures - paying special attention to linguistic subtleties like irony and slang, perhaps by incorporating tweet metadata such as user profiles or linked content. This more comprehensive approach promises to endow BERTweet with a finer grasp of context, potentially reducing misclassifications and enhancing overall performance.

2.3 Method 3: CNN

Exploring the use of Convolutional Neural Networks (CNNs) for text classification in natural language processing (NLP) tasks, like disaster tweet classification, presents an opportunity to evaluate the effectiveness of this architecture outside its conventional application in image processing. Despite the rise of Transformer-based models like BERT, which have gained preeminence in NLP, recent literature implies that CNNs can still be an integral component in this domain. To this end, our analysis delves into the application of one-dimensional CNNs (1D-CNNs) for distinguishing disaster tweets, aiming to exceed the performance benchmarks set by previous methods.

Model Selection

Our decision to evaluate CNNs is grounded in recent research findings that continue to highlight the utility of CNNs in NLP. While there was an assumption that Transformer models had become the single go-to architecture for text data, our domain expert offered a different perspective, suggesting the enduring relevance of CNNs. A review of recent publications bolstered this view, suggesting a renewed interrogation into the application of CNNs for text classification tasks.

Model Explanation

The 1D-CNNs differ from their 2D counterparts by moving the filter across one dimension—corresponding to the sequential nature of text data—allowing the kernels to capture patterns in the succession of words. We used varying kernel sizes to examine the effect of capturing different range dependencies between words, alongside multiple convolution layers to deepen the model's feature extraction capabilities.

Pre-Processing

Essential pre-processing steps involved vectorizing the tweet sentences, which transformed each unique word into a distinct dimensional vector set at orthogonal angles to one another. This high-dimensionality was then reduced using embedding techniques, where words with semantic similarities were placed closer together in a lower-dimensional vector space.

Non-Pretrained Embeddings

In this study, we expressed words in a 30-dimension space using non-pretrained embeddings, allowing us to capture the semantic features relevant to our disaster classification task without the influence of an external corpus.

Word2Vec Embeddings

Pretrained Word2Vec embeddings were also utilized, leveraging the model's understanding of word dependencies in a larger dataset to inform our classification. We chose the CBOW-trained Word2Vec model on Google News due to its demonstrated performance in the literature.

BERTweet Embeddings

Additionally, BERTweet embeddings, pretrained specifically on tweets, were tested. The model's subword tokenization allowed for a granular representation of tweet text, offering unique insights in understanding the intricacies of social media language.

CNN Model Architecture

The first round of experiments used a 1D-CNN with a kernel size of 5 and 64 filters. Comparisons were made between non-pretrained embeddings, Word2Vec, and BERTweet embeddings, using binary cross-entropy as the loss function and a sigmoid activation in the output layer for binary classification.

2.3.1 Results

Surprisingly, the CNN models that were applied to the task of classifying disaster tweets witnessed a nuanced performance advantage when incorporating Word2Vec embeddings over models that relied on non-pretrained embeddings. This was unexpected, especially considering that the models with BERTweet embeddings, which were pretrained on domain-specific data, were anticipated to have the upper hand. However, the domain-specific data might not be domain specific enough compared to our use-case, since BERTweet was trained on english tweets in general but not on disaster related tweets. The minimal outperformance of Word2Vec underscores the sometimes unpredictable nature of model efficacy, given that prior assumptions favored the domain-specific Bert embeddings.

A consistent pattern of misclassification across different models and embedding techniques surfaced, signaling the presence of fundamental issues inherent in the text data itself. Rather than reflecting problems with the models or embedding methods, these recurrent errors speak to the inherent challenges of text classification, suggesting that issues might lie with ambiguous language use, context, or nuances such as irony and sarcasm in the dataset.

The inspection of individual layer contributions within the CNNs, especially the filters, disclosed a lack of consistency in their impact upon the predictions. Some filters illustrated noteworthy influence while others did not, contradicting the expectation that particular filters would consistently dominate in predictive importance. The implications of these findings merit further investigation, with a focus on understanding which features are most salient for successful disaster tweet classification.

The exploration of models with increased architectural complexity, through the introduction of max pooling layers, dropout techniques, and additional fully connected layers, failed to produce the hypothesized improvement in classification results. Surprisingly, these more intricate models sometimes succumbed to overfitting faster than their simpler counterparts. These observations reinforce the notion that augmenting complexity is not synonymous with enhanced performance and that a well-optimized, streamlined architecture might often be preferable.

In summation, the experiment utilizing CNNs for the classification of disaster tweets highlighted that the pursuit of complexity in neural network design does not assuredly lead to superior performance. Notwithstanding the diverse approaches and embedding strategies tested, the models faced consistent challenges in text classification. The Word2Vec-equipped 1D-CNN model emerged as a notable approach, albeit with a modest margin of performance advantage, reiterating the value of model optimization over complexity. Future work in this area is encouraged to build upon these findings, potentially integrating broader contextual information to refine the NLP models' capacity for classifying tweets accurately.

3 System Description

3.1 Dataset

The dataset used for the evaluation of the machine-learning methods comes from a Kaggle competition titled „Natural Language Processing with Disaster Tweets“. It was designed to provide a gateway for data scientists entering the field of Natural Language Processing (NLP).

The dataset comprises 10,000 tweets that have been manually classified into two categories: tweets that are about real disasters and tweets that are not. The competition is tailored for models that can distinguish between tweets that describe actual emergency situations from those that do not, even if disaster-related terminology is present.

Each tweet sample in both the training and the test set is accompanied by metadata containing the following information:

1. The text of the tweet: This is the primary data to be used for the text classification task.
2. A keyword from that tweet: The keyword is a term from the tweet which may be related to disaster terms (some samples may have this blank).
3. The location the tweet was sent from: Like the keyword, this can be blank for some samples but when available provides context in relation to the geographic area of the tweet.

The files included in the dataset are as follows:

- `train.csv`: his file constitutes the training set containing examples to train the models.
- `test.csv`: This set is to test the models' performance in predicting whether new, unseen tweets are about real disasters or not.
- `sample_submission.csv`: This file provides a template for submitting predictions in the correct format to the Kaggle platform.

Regarding the structure of 'train.csv', it includes several columns:

- `id`: It serves as a unique identifier for each tweet.
- `text`: Contains the actual tweet text.
- `location`: States the geographic location where the tweet was sent from.

- `keyword`: This is a notable keyword extracted from the tweet's text.
- `target`: A binary indicator, only present in 'train.csv', which shows whether a tweet pertains to a real disaster (1) or not (0).

This dataset is particularly well-suited for this task due to its focus on NLP and its moderate size, allowing for experimentation with different models without requiring extensive computational resources. The inclusion of keywords and location data also provides the opportunity for feature engineering and the exploration of how additional metadata can support or enhance the performance of text classification models.

3.2 Preprocessing

The preprocessing stage of the dataset is critical to ensure that the machine-learning models function effectively, without interference from irrelevant or redundant data. In this phase, we have decided not to utilize the 'keyword' and 'location' columns for classification purposes. While these columns might offer additional context that could enhance predictions in a hybrid model, the focus remains solely on the tweet text for this analysis.

Firstly, to prevent any leakage of training data into the test set, it was essential to scour the dataset for any duplicates. Upon inspection, 110 duplicate instances were discovered. To maintain the integrity of the data, only those duplicates that were consistent in their 'target' values were retained, effectively removing any potential contamination that could bias the models.

Subsequently, a distinct training and a test set were established to segregate the data effectively. This division aims to avoid any inadvertent gain of information about the test set during the training phase.

Further examination confirmed that no 'NA' values were present in the dataset, and due to the earlier removal of duplicates, there were no duplicate values left either. However, an imbalance was noted in the class distribution—there were unequal numbers of disaster and non-disaster tweets. This skewness necessitates careful consideration when choosing an evaluation metric to ensure it accurately reflects the models' performance.

Analysis of the length of the tweets revealed that most contained between 10 to 20 words. Few tweets were longer than 30 words or shorter than 5 words, which may present challenges for the models as shorter tweets could lack sufficient information for accurate classification.

In preparation for TF-IDF analysis, the tweets underwent further cleaning. Special HTML characters were converted to their corresponding correct characters, eliminating the risk of misinterpreting artifacts like „&amp“ as valid words. All non-alpha characters such as hashtags, mentions, punctuation, conjunctions, etc., were stripped away, focusing analysis solely on the relevance of words themselves. This was only done for the TF-IDF analysis. Though in retrospect, it would have been beneficial to apply this to the models as well.

Our TF-IDF model training involved refining the stopwords list. The built-in list was too narrow, failing to filter out many common stopwords, so a more extensive list from the internet was utilized. Additionally, accents were removed, and all words were converted to lowercase to prevent TF-IDF from misidentifying identical words due to case differences or diacritics.

Notably, words that carry more negative connotations were found to be more relevant in tweets categorized as disasters, a pattern not observed in non-disaster tweets. This distinction might provide valuable insights into the characteristics that differentiate disaster tweets from their non-disaster counterparts.

4 Evaluation

4.1 Metric Selection

To determine the efficacy of our models in classifying disaster tweets, we employ several metrics catering to different aspects of performance.

The F1 Macro-Score is chosen as a primary metric to account for the imbalance in our classes. It ensures that both classes, disaster and non-disaster tweets, contribute equally to the final score, thus preventing the

overshadowing of the minority class by the majority.

Accuracy is also tracked to gauge the overall correctness of predictions. It reflects the proportion of total tweets classified correctly but should be interpreted in conjunction with other metrics due to the imbalanced dataset.

The Confusion Matrix is imperative in understanding the types and quantities of errors made by the classifiers. It reveals the frequency of first-degree errors (false positives) and second-degree errors (false negatives), which is critical in a domain where the repercussions of misclassification can vary greatly.

Together, these metrics form a comprehensive framework for evaluating the performance of different models on the task of disaster tweet classification.

4.2 Results Comparison

4.2.1 Quantitative Results

TF-IDF HGBC	**no disaster**	**disaster**
no disaster	761	88
disaster	228	420

Tabelle 1: Confusion Matrix for TF-IDF HGBC

TF-IDF SVM	**no disaster**	**disaster**
no disaster	789	60
disaster	229	419

Tabelle 2: Confusion Matrix for TF-IDF SVM

BERTweet	**no disaster**	**disaster**
no disaster	695	154
disaster	148	500

Tabelle 3: Confusion Matrix for BERTweet

Across the four models evaluated, the TF-IDF SVM outperformed others in terms of accuracy, boasting an impressive 80.69%. It also yielded the highest precision; however, all models showed comparable recall rates, with BERTweet leading slightly at 77.16%. When it came to the F1 Score, which balances precision and recall, TF-IDF SVM maintained its superiority with 74.36%, followed closely by BERTweet.

In terms of errors, the TF-IDF SVM model demonstrated a notable advantage in minimizing first-degree errors, with only 60 misclassifications of non-disaster tweets as disaster tweets. Interestingly, BERTweet and CNN grappled with a higher number of first-degree errors, possibly indicating a tendency toward higher recall at the cost of precision.

BERTweet, while excelling in recall, manifested a balanced error profile with relatively equal first and second-degree errors. This suggests an equilibrium between the types of misclassifications, which may reduce bias towards either class.

On the other hand, CNN experienced the highest number of second-degree errors, which implies that it was more conservative in predicting the disaster class and consequently mislabelled a notable number of disaster tweets as non-disaster.

Overall, while each model has its strengths, the TF-IDF SVM emerged as the most robust, with the highest accuracy and F1 Score, coupled with the lowest first-degree error count, revealing a strong capacity for balanced classification in this dataset.

CNN	no disaster	disaster
no disaster	688	161
disaster	170	478

Tabelle 4: Confusion Matrix for CNN

Model	Accuracy	Precision	Recall	F1 Score
TF-IDF HGBC	78.89%	82.68%	64.81%	72.66%
TF-IDF SVM	80.69%	87.47%	64.66%	74.36%
BERTweet	79.83%	76.45%	77.16%	76.80%
CNN	77.89%	74.80%	73.77%	74.28%

Tabelle 5: Performance Metrics for Different Models

4.2.2 Qualitative Results

In the qualitative analysis of our models' misclassifications, several patterns emerge that provide insights into where and why each algorithm stumbles.

For instance, BERTweet misclassified tweets containing the phrases „natural disaster“ and „war zone“ despite these not referring to actual disasters. This suggests that BERTweet may heavily weigh specific keywords without enough context consideration. Phrases common in discussions around disasters, like „sinking away from you,“ were also mistaken as disaster-related, which implies BERTweet could be overfitting to disaster-specific language.

Conversely, CNN erred in classifying a tweet mentioning „da bomb“ as a disaster. This could suggest that CNN has a tendency to misinterpret urban slang or colloquial expressions, possibly due to a lack of representative training data capturing these linguistic nuances.

Across all models, the tweet about Bin Laden's family plane crash was correctly identified as a disaster, confirming that models have learned to associate such keywords with the disaster class effectively. However, the failure of BERTweet and other models to recognize a tweet jokingly self-deprecating as a „KNOBHEAD“ as a non-disaster indicates that models may not differentiate sarcasm or self-deprecatory humor well.

Another profound misclassification involves a tweet expressing gratitude for safety after a car wreck. All models, except CNN, failed to classify it as a disaster, possibly because the broader context conveys a sense of relief rather than the occurrence of a disaster. This could reflect a gap in the models' grasp of deeper semantic meanings and the emotional undertones of the text.

These qualitative observations underline the need for models to not only recognize disaster-specific keywords but also to contextualize language more effectively, accounting for sarcasm, idiomatic expressions, and sentiment to reduce misclassification rates. These factors play into the nuanced understanding of human language and present fruitful avenues for model improvement.

4.3 Results Discussion

The comparative evaluation of the machine learning methods showcased a spectrum of performances influenced by various factors inherent to each model's architecture and the nature of the dataset. Each model brought unique strengths and weaknesses to bear upon the classification task.

The SVM, when paired with TF-IDF, yielded a slight improvement over the HGBC model, possibly due to its ability to find a better hyperplane in the feature space provided by the TF-IDF vectors. The recurrent misclassification patterns across the TF-IDF models pointed towards difficulties in capturing the subtle linguistic cues, such as sarcasm or context-dependent meaning.

BERTweet, despite its advanced architecture, did not fulfill the high expectations set for it, with evaluation metrics suggesting an overfitting tendency and challenges in generalizing its learning to unseen data. The TF-

Model	First Degree Error	Second Degree Error
TF-IDF HGBC	88	228
TF-IDF SVM	60	229
BERTweet	154	148
CNN	161	170

Tabelle 6: Error Analysis for Different Models

IDF with HGBC model, while strong in accuracy and F1 score, exhibited a vulnerability to misclassifying tweets with non-disaster-related sarcastic or dramatic language as disasters. This model also demonstrated an overemphasis on certain keywords for classification decisions.

On the other hand, the CNN models with Word2Vec embeddings revealed an unexpected edge over those with supposedly superior BERTweet embeddings. This suggested that, in some cases, the ability of a model to generalize might not be significantly enhanced with domain-specific pretraining. However, architectural complexities in the CNN did not guarantee better performance, with simpler designs sometimes proving more effective.

From the results, it is evident that none of the models completely mastered the task at hand, with each facing similar issues regarding the interpretation of context, the variability in language use, and the identification of subtle linguistic nuances. This significant insight highlights the need for a balanced approach that combines the strengths of different models and possibly leverages additional contextual information to improve the classification of disaster-related content on Twitter. The results collectively underscore the importance of feature representation and the potential benefit of model ensembles or hybrid approaches to mitigate individual model weaknesses.

4.4 Conclusion

In conclusion, this report has provided a comparative analysis of varying machine-learning methods for classifying disaster-related tweets. The results elucidated the distinct capabilities and shortcomings of each model. The TF-IDF SVM model emerged as the most effective approach, achieving superior accuracy and F1 scores. This report underlines the complexity of text classification tasks, especially when dealing with the nuances of human language in social media.

tweet	label	BERTweet	HGBC	SVM	CNN
she's a natural disaster she's the last of the American girls ??	no disaster	disaster	no disaster	no disaster	no disaster
In your eyes I see the hope I once knew. I'm sinking. I'm sinking away from you. Don't turn around you'll see... You can make it.	no disaster	disaster	no disaster	no disaster	no disaster
13 reasons why we love women in the military - lulgzimbestpicts http://t.co/XKMLQ99SjY http://t.co/a3RGQuCUgo	no disaster	disaster	no disaster	no disaster	no disaster
E-Hutch is da bomb ?? http://t.co/aqmpxzo3V1	no disaster	no disaster	no disaster	no disaster	disaster
They turned Jasmines house into a war zone. ?? #LittleWomenLA	no disaster	disaster	no disaster	no disaster	no disaster
Neil_Eastwood77: I AM A KNOB-HEAD!! Bin Laden family plane crashed after 'avoiding microlight and landing t... http://t.co/dUVUzhMVUT	disaster	no disaster	disaster	disaster	disaster
Rt hirochii0: There is no country that making fun of Hiroshima 's tragedy but Korea. http://t.co/And1Btizao #Indonesia #Malaysia #Jamaica	disaster	no disaster	disaster	disaster	disaster
@eeenice221 true because of the truck that caught fire?	disaster	disaster	no disaster	no disaster	disaster
@KurtSchlichter He's already done it by negotiating with the #1 state of terrorism in the World. What was his hurry in trying to get a deal	disaster	no disaster	no disaster	no disaster	disaster
My baby girls car wreak this afternoon thank God no serious injuries and she was wearing her seatbelt!!!... http://t.co/NJQV45ndS2	disaster	no disaster	no disaster	no disaster	no disaster

Tabelle 7: Classification of Tweets by Different Models