Analysis of Positional Encoding and Attention Mechanisms in Disaster-Related Tweet Classification

Javier Rodriguez & Edgar Leon

Abstract

This study explores how different methods of positional encoding and attention strategies affect the classification of disaster-related tweets using a Transformer-based model. We suggest that dynamic, trainable positional encodings and adjustable attention mechanisms will outperform static ones and only focus narrowly, especially in the dynamic realm of social media communication. We also investigate the potential benefits of incorporating pre-trained word embeddings to improve our model's understanding of disaster-specific language. We anticipate this research will contribute to the Natural Language Processing (NLP) field and enhance tools for managing disasters more effectively. We aim to improve the accuracy and speed with which vital information is shared during live disaster events.

Introduction

Social media has become a key source of information during disasters, which calls for refined real-time analysis methods. As a result, progress in natural language processing (NLP) has become crucial. Our research looks into several positional encoding strategies and attention mechanisms in a Transformer-based model, specifically examining tweets about natural disasters. Driven by the desire to better disaster management through social media insights, our study seeks to create a tool that deepens comprehension and acts as a helpful resource in managing disaster responses.

Background

In disaster management, quickly processing and understanding text from social media can save lives. Natural language processing, or NLP, has become an essential tool for real-time analysis that leads to actionable insights. At the heart of recent NLP advancements are Transformer models, which are highly effective in handling sequential data thanks to their innovative attention mechanisms [1]. These include positional encoding, which is critical for grasping language order, something older models often need to pay attention to.

Current research confirms that positional encoding is vital for recognizing the sequence of words, which is crucial for tasks like machine translation and syntactic parsing. Developing positional encoding from static methods to more complex, adaptable ones marks an essential development in NLP. This progression, along with different approaches to attention, has dramatically improved model performance.

Additionally, the field has made strides in data augmentation techniques designed for various linguistic tasks and datasets. These adjustments are vital for addressing social media's unique and

often abbreviated language, such as the informal and evolving language found on Twitter during disasters [2].

Our study aims to delve deeper into and refine these mechanisms. By integrating learnable positional encodings and adaptive attention mechanisms in a Transformer-based model, we plan to deepen the understanding and boost the effectiveness of NLP in disaster response. Our goal is to set a new standard in the field through detailed analysis and experiments, offering valuable knowledge and resources for academic study and real-world disaster management scenarios.

Methods

We developed a transformer-based classification model utilizing neural machine translation (NMT) architectures adapted for text classification using the last hidden state. Utilizing Keras Layers as our framework, we built the standard transformer architecture by integrating each transformer section (Positional Embedding, Base Attention, Feed Forward) through class instantiations (Figure 1).

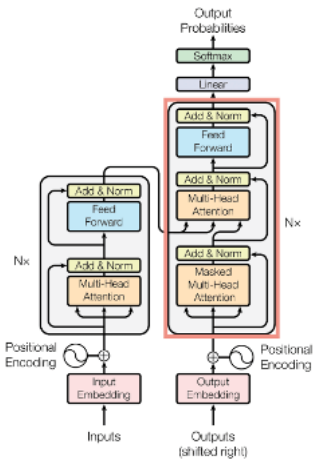

Figure 1: Transformer-Based Classification Model

After that, we explored updating the model architecture and explored features such as Learnable Positional Encodings, Local Attention, Adaptive Attention Span, Custom Schedule Regularization. We also revisited the combination of Learnable Positional Encodings with an Adaptive Attention Span. After evaluating these enhancements, we chose the best-fitting model and proceeded to fine-tune it using Optuna for hyperparameter optimization. Subsequently, we integrated pre-trained embedding vectors and rigorously assessed the performance improvements over the baseline model (Figure 2).

The data for this project came from the Kaggle competition NLP Disaster Tweets. Natural Language Processing with Disaster Tweets is a Natural Language Processing (NLP) competition hosted by Kaggle [4]. The goal was to train a model that could correctly predict whether or not a Twitter tweet was about a natural disaster or was using similar language figuratively (Figure 3). The data consisted of keywords, locations, text, and targets, with an equal distribution of positive and negative labels.

We improved the dataset by utilizing the tweetnlp package to classify each tweet. The classifications

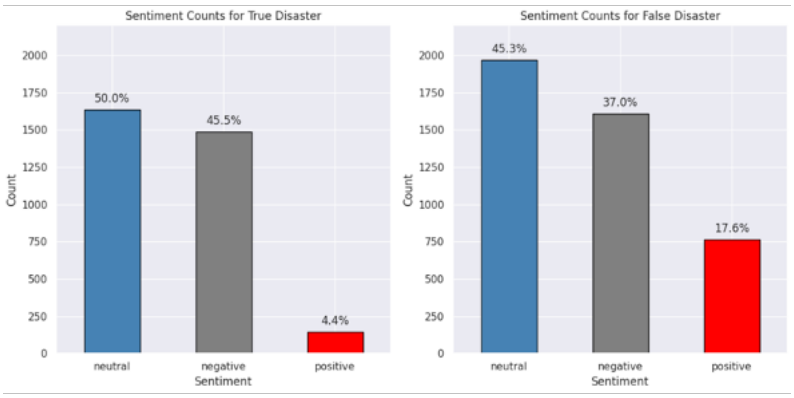

Figure 2: Sentiment Counts for True and False Disasters

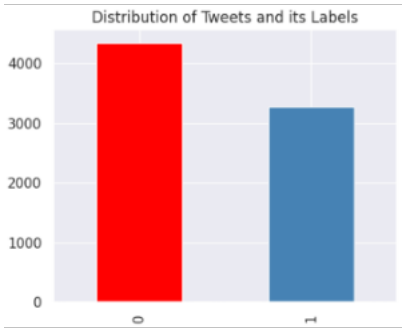

Figure 3: Distribution of Tweets and Labels

included the sentiment category, whether it was ironic or not, the topic category, and the tweet's emotional tone [5][6]. We embedded these labels directly into the text to enrich the data.

For our preprocessing, we enhanced the text by inserting the keyword and location tags, removing all punctuation, converting all text to lowercase, and removing special characters. We used a Keras standard Tokenizer and set the maximum sentence length to 30 Tokens.

Results

Initial Evaluation on Raw Data The preliminary assessment of our Transformer-based model, using unprocessed data, yielded a validation loss (val_loss) of 0.5612 and a validation accuracy (val_accuracy) of 0.7806 over ten epochs. Notably, overfitting occurred beyond the seventh epoch. We set the configuration for this initial run with two layers (num_layers = 2), a model dimensionality of 100 (d_model = 100), a feed-forward network dimensionality of 512 (dff = 512), eight attention heads (num_heads = 8), and a dropout rate of 0.3v (Figure 4). The analysis revealed many false positives (FP) and false negatives (FN), with counts of 520 and 450, respectively. Attention analysis on misclassified examples, such as "Blew up those mentions" (a false positive), indicated a disconnect where crucial words did not receive adequate attention, suggesting a potential deficiency in the model's understanding of context.

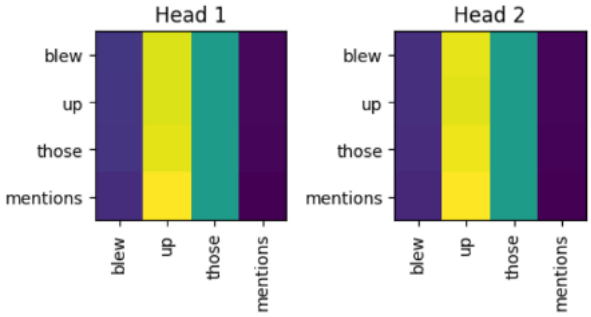

Figure 4: Configuration of the Initial Run

Embedding Enhancements To address this, we augmented the model with 100-dimensional embeddings from the FastText library, pre-trained on a corpus of Twitter-specific language ("fasttext_english_twitter_100d.vec"). Subsequent training on this enhanced data improved the validation loss to 0.4808 and accuracy to 0.8042—an uplift of 2% (Figure 5).

Baseline Model Performance with Pre-Trained Twitter Embeddings Further refinements led to integrating the model with FastText embeddings on cleaner, preprocessed input data. The resulting validation metrics showed a loss of 0.4518 and an accuracy of 0.8346. Hyperparameter tuning refined the model structure to two layers (num_layers = 2), with a model size of 100 (d_mode = 100), two attention heads (num_heads = 2), a smaller feed-forward dimensionality of 256 (dff = 256), and an increased dropout rate of 0.6. While the false negatives (FN) at 308 were higher than false positives (FP) at 119, a detailed review indicated that many resulted from initial data mislabeling. Notably, attention visualization demonstrated that the model focused more on disaster-relevant terms (Figure 6). Our next step was to update the model, experimenting with different upgrades. We updated several models with Learnable Positional Encodings, Local Attention, Adaptive Attention, and Custom Regularization.

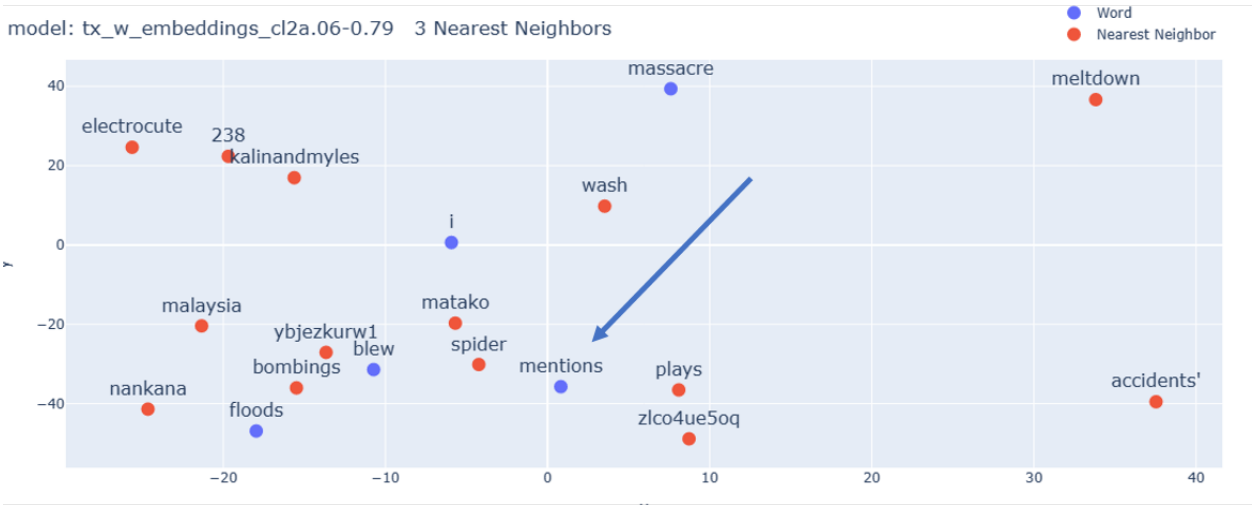

Figure 5: Embeddings

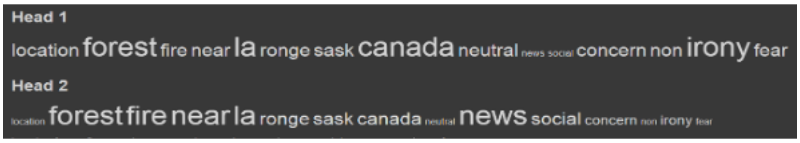

Figure 6: Head1

Comparative Evaluation of Models In our investigation of how various positional encoding strategies and attention mechanisms affect disaster-related tweet classification using a Transformer-based approach, we developed advanced versions of the Transformer framework with enhancements optimized for TensorFlow machine learning tasks:

Baseline Model: This initial model includes trainable positional encodings and a dynamic Adaptive Attention Layer, along with conventional elements such as multi-head attention, normalization, and residual connections. It is designed for classification tasks and requires precise adjustments, especially for its advanced features.

Learnable Positional Encodings: This model builds on the baseline by integrating trainable positional encodings within its Positional Embedding Layer. This innovation enables the model to better adjust to the particularities of different datasets, potentially improving performance where the positional context is critical.

Local Attention: Unlike the baseline, this iteration employs local attention to concentrate on smaller data segments. Unlike typical baseline models that apply global attention to the entire input sequence, the iteration could improve efficiency and outcomes for tasks with significant local context.

Adaptive Attention Span: This variation features an AdaptiveAttentionLayer that adjusts the attention span according to the input size alongside a configurable LocalAttention mechanism to manage better the dataset's different sequence lengths and focus areas.

Implementing Custom Regularization: This model diverges from the baseline by embedding L1 and L2 regularization within the embeddings and attention mechanisms. Aiming to prevent

overfitting, it achieves comprehensive regularization throughout embeddings, attention projections, and the feed-forward network, bolstering the model's generalization ability.

Learnable Positional Encodings + Adaptive Attention Span: By merging trainable positional encodings with an AdaptiveAttentionLayer, this model dynamically fine-tunes attention relative to the sequence length. It maintains a standard Transformer FeedForward network, incorporating dropout for regularization and applying conventional practices like residual connections and layer normalization to ensure stability.

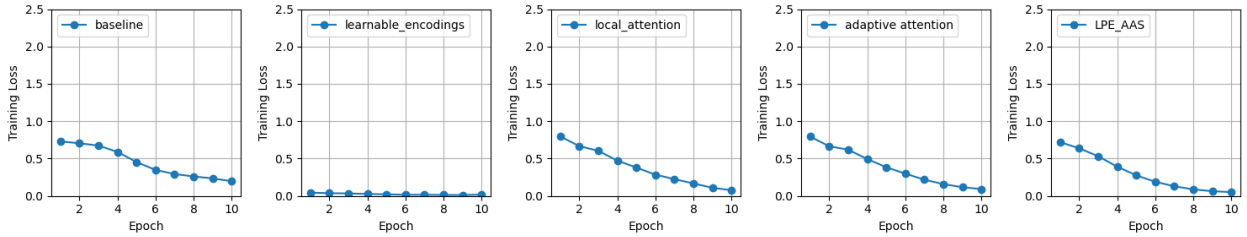

Figure 7: Training Loss

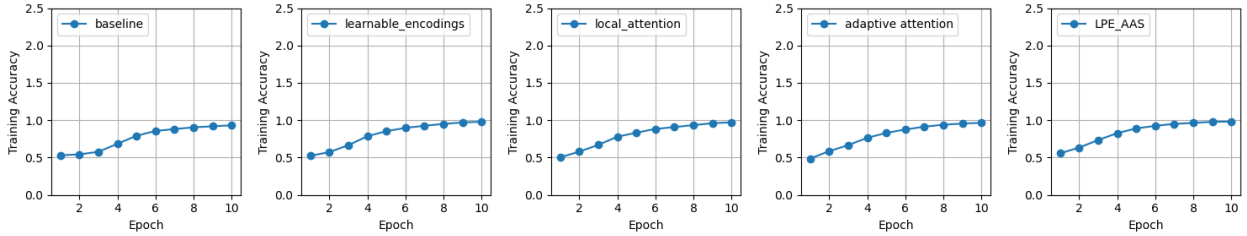

Figure 8: Training Accuracy

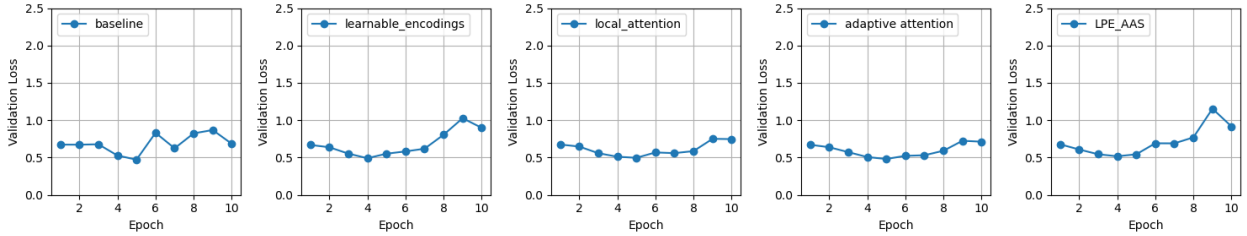

Figure 9: Validation Loss

In the comparative analysis of five different models trained for a classification task:

Model 1: Baseline shows a solid increase in training accuracy but suffers from a significant increase in validation loss later in training, which suggests overfitting (Figure 7). The validation accuracy peaks and then fluctuates, indicating that the model might need to generalize better to unseen data. ***Model 2: Learnable Positional Encodings*** starts with low accuracy and loss but improves steadily. However, despite high training accuracy, validation loss increases over time, which again suggests overfitting. Validation accuracy remains fairly stable after peaking (Figure 8). ***Model 3: Local Attention*** has good initial progress, improving training and validation accuracy. Yet, as with the other models, the increasing validation loss points to overfitting. It does, however, reach the highest validation accuracy at one point (Figure 9). ***Model 4: Adaptive Attention***

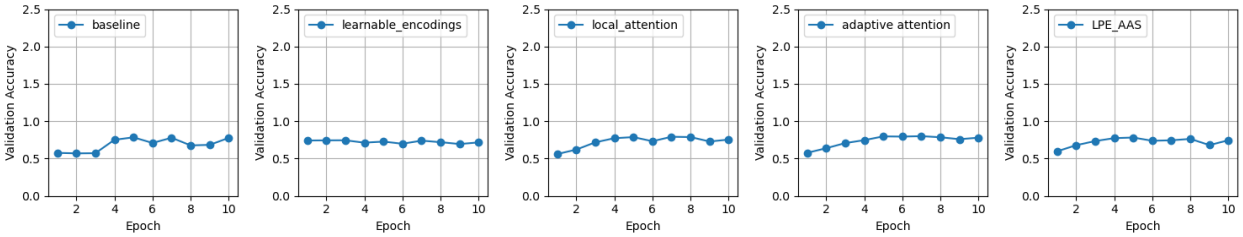

Figure 10: Validation Accuracy

Span begins with lower performance metrics but shows consistent improvement. Although it ultimately displays signs of overfitting (with increasing validation loss), it achieves the highest validation accuracy by the end of training (Figure 10). ***Model 5: Implementing Custom Regularization*** has peculiarly high initial losses, which decrease consistently. This unusual pattern might indicate an error in loss computation or a scale issue. Despite improvements, the loss values doubt the reliability of this model's performance metrics. ***Model 6: Learnable Positional Encodings + Adaptove Attention Span*** starts with mid-range accuracy and loss but shows consistent improvement in accuracy. Like the other models, the increasing validation loss later in training is a concern. Nonetheless, it maintains a relatively high validation accuracy in later epochs without drastic fluctuations.

Upon evaluation, Model 4 emerges as the most competent choice. Although it did not boast the topmost training accuracy, this characteristic is beneficial because overly high training accuracy raises the risk of overfitting. Instead, Model 4 demonstrates the best validation accuracy as the training progresses, indicating a superior capacity for generalizing compared to its counterparts.

However, one might still consider selecting Model 2 due to its unique features and its promise despite indications of overfitting. The argument for Model 2 could be based on several merits:

Firstly, Model 2 shows rapid advancement in training and validation accuracy early on, suggesting an adequate learning curve. Secondly, it employs learnable positional encodings rather than fixed ones. This flexibility allows the model to adapt to the specific sequential patterns in the data, which is particularly advantageous when dealing with the unpredictable nature of Twitter data, where standard patterns may not be evident.

Moreover, the presence of overfitting implies that there is room for further model refinement. Implementing regularization, fine-tuning the learning rate, or increasing the dataset could improve its generalization capabilities. Observed stability in its validation accuracy signals its potential to maintain performance consistency, a critical aspect for models in production environments.

Model 2's framework, incorporating transfer learning with its adaptable positional encodings, might show advantages when fine-tuned on specialized tasks or datasets, catering to more specific needs. This adaptability could be paramount in addressing the intricacies of the data and capturing subtle patterns, thereby rendering Model 2 more robust against data variability.

Lastly, the contextual relevance of the predictions made by Model 2 is another factor to consider. If the learnable positional encodings enable the model to recognize contextually important nuances, even at the expense of some accuracy, it could be deemed more suitable for specific applications where contextual precision is paramount.

Since Twitter feeds feature various language styles, including casual speech, abbreviations, hashtags,

and unique platform-specific expressions, the challenge is to convey messages within the concise character limit. This results in a tapestry of compact and diverse linguistic forms. Learning positional encodings provides a substantial benefit because they can adapt to Twitter's particular language structures and sequences. These encodings can capture the distinctive positional patterns and subtleties of how information is organized within a tweet's length confines. This enables models to develop more effective representations of position tailored to the specific quirks of tweet content.

Moreover, the importance of word order and the context of their position changes a lot depending on the task, whether it's understanding feelings in text, sorting topics, or spotting false information. Learnable positional encodings are adaptable. They can be fine-tuned to make the model as sensitive as needed to where words are placed, depending on the task's requirements.

In our research, we used Model 2. We utilized Optuna, a tool that makes it easier to choose the right settings for a model by automating the hunt for the best hyperparameters for our Transformer model. We set up an Optuna study that ran multiple tests. Optuna's algorithm gave us different values for things like how many layers to use, the size of the model, the number of attention heads, the size of the feed-forward network, and the dropout rate. With each set of hyperparameters, it suggested, it created, trained, and tested the Transformer model and recorded the best score for correctly validating data. Once we completed several tests, Optuna showed us the best combination of settings it found during the search.

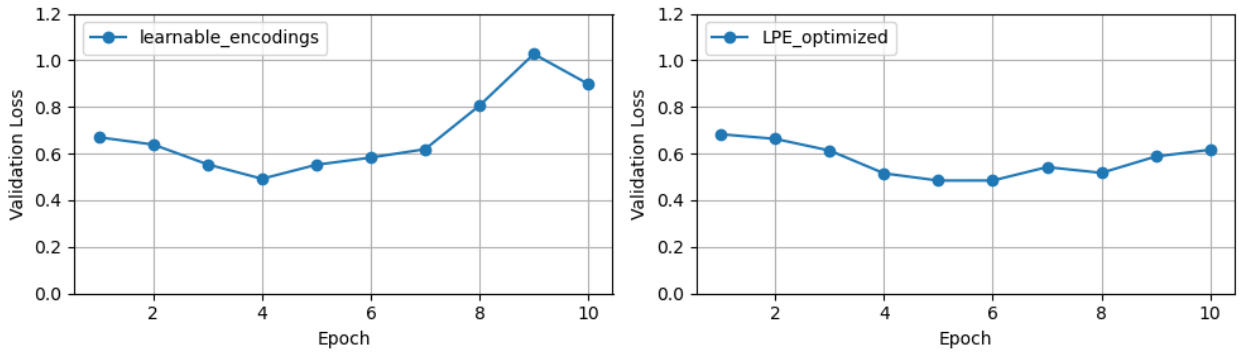

Figure 11: Hyperparameter Optimization via Optuna

The outcome of this exercise was revealing in terms of model performance. Upon optimizing Model 2 with Optuna, our final epoch metrics showed an increase in validation accuracy from 0.7776 to 0.7907, signaling that the optimized model had a superior ability to generalize when exposed to new data. Additionally, the optimized model indicated a lower validation loss, decreasing from 0.6846 to 0.6165, which suggests that the optimized model's predictions were more closely aligned with the actual values (Figure 11). This loss reduction represents an improved performance of the model. In summary, the optimization of Model 2 resulted in a model that was more precise in terms of accuracy and generated predictions with greater fidelity, as demonstrated by the enhanced metrics post-optimization.

Learnable Positional Encoding (LPE) Transformer Classifier We selected Model 2 for final evaluation. Model 2 achieved a validation loss of 0.4276 and an accuracy of 0.8215. The accompanying attention score heatmap comparison between the Model 2 and the baseline illustrates the LPE's superior grasp of word positional relationships, as evidenced by its distinctive checkerboard attention pattern. Also, the Model 2 showcased a more balanced performance between false positives (154) and false negatives (200) (Figure 12).

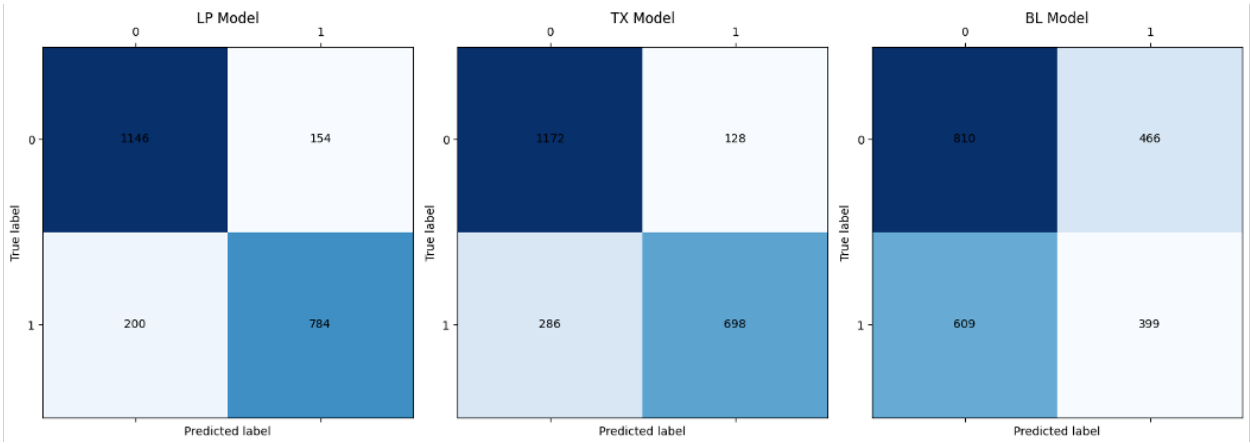

Figure 12: Comparison of Model Performance

We also found Model 2 to change the position of the embeddings after training by twice as much as the baseline (~0.42 to 0.22). On the left, we see the top changed embeddings after training. Both models affected almost the exact words, but the change was higher for the LPE. Furthermore, the probability of TRUE was, on average, higher than a TRUE classification for the same sentence against the baseline (Figure 13).

While the Baseline model demonstrated a slightly higher accuracy than Model 2, the negligible margin suggests that the quality of input data and the model's ability to generalize are pivotal to real-world applications. This observation aligns with the insight that there are diminishing returns in accuracy gains when using increasingly complex models for social media data analysis.

Using FastText embeddings, trained on a Twitter corpus, gave our models a nuanced understanding of disaster-related discourse Figure 14 & Figure 15). This highlights the importance of domain-specific embeddings in improving classification tasks on social media platforms.

However, our study moves the needle by demonstrating that while such embeddings are beneficial, the type of architecture that deploys the embeddings can dramatically impact their effectiveness. Model 2's balanced performance in reducing false positives and negatives emphasizes the value of a model's internal architecture in understanding context—a nuance not fully explored previously.

Discussion

The implications of our findings extend beyond incremental improvements in model metrics; they invite a re-examination of prevailing assumptions in using NLP for disaster response. While the Baseline model demonstrated a slightly higher accuracy than the LPE model, the negligible margin suggests that the quality of input data and the model's ability to generalize, are pivotal to real-world applications.

Our research corroborates the growing consensus that adaptive attention mechanisms contribute to model robustness. The enhanced performance of Model 4, which incorporated learnable positional encodings, suggests that the ability to discern the relevance of different parts of a tweet adaptively is critical in a domain characterized by noisy and informal language.

Using FastText embeddings, trained on a Twitter corpus, provided our models with a nuanced

```
tx_disaster_words = top_changed_embeddings(
tx_disaster_words_list = [item[0] for item
tx_disaster_words
```

```
[('suicide', 0.2542814085893426),
 ('hiroshima', 0.2366639098735998),
 ('derailment', 0.23234185947136063),
 ('positive', 0.2134087088705059),
 ('california', 0.21303300061344782),
 ('bombing', 0.21094468542664885),
 ('debris', 0.21006077369589732),
 ('wildfire', 0.20714493198224423),
 ('killed', 0.200944669891897),
 ('northern', 0.19962536227301594)]
```

```
lp_disaster_words = top_changed_embeddings(
lp_disaster_words_list = [item[0] for item
lp_disaster_words
```

```
[('hiroshima', 0.496638707030925),
 ('spill', 0.4849229581973467),
 ('suicide', 0.448507128728351),
 ('bomb', 0.4369772545959775),
 ('derailment', 0.4233027778885434),
 ('northern', 0.4181329351044703),
 ('california', 0.4133995836162507),
 ('atomic', 0.40671896164405086),
 ('killed', 0.39854748212169594),
 ('wreckage', 0.3949224709598692)]
```

Figure 13: Positional Changes of Embeddings

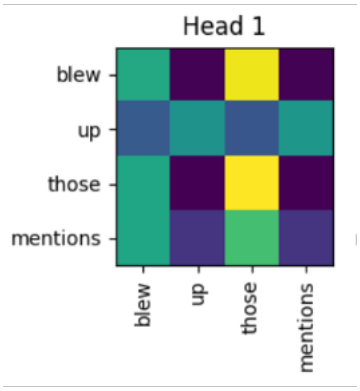

Figure 14: Head1a

Figure 15: Head1b

understanding of disaster-related discourse. This finding echoes the importance of domain-specific embeddings in improving classification tasks on social media platforms.

However, our study moves the needle by demonstrating that while such embeddings are beneficial, the architecture within which they are deployed can dramatically impact their effectiveness. Model 2's balanced performance in reducing false positives and negatives emphasizes the value of a model's internal architecture in understanding context—a nuance not fully explored (Figure 16 & Figure 17).

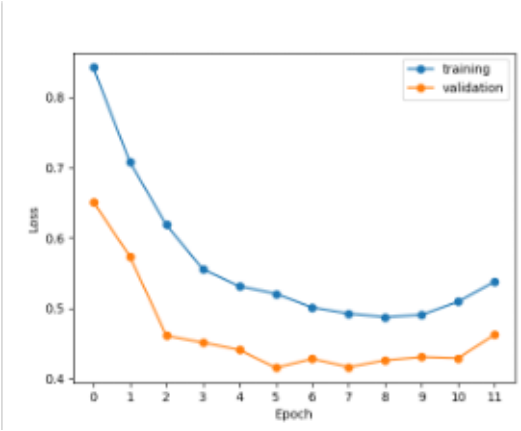

Figure 16: Baseline Model Performance Using Enhanced Data

Conclusion

In the comparative analysis of our study, we observed that the Baseline model slightly outperformed the Learnable Positional Encoding (LPE) model, achieving a validation accuracy of 84% versus the latter's 82%. Despite the marginal difference, it was notable that the Baseline model demonstrated greater resilience against overfitting. This finding is an essential consideration for model reliability. Nonetheless, based on the insights gleaned from this study, we anticipate that with the introduction of a more extensive and more varied real-time data set, the LPE model could potentially surpass the Baseline model in terms of performance due to its adaptive features.

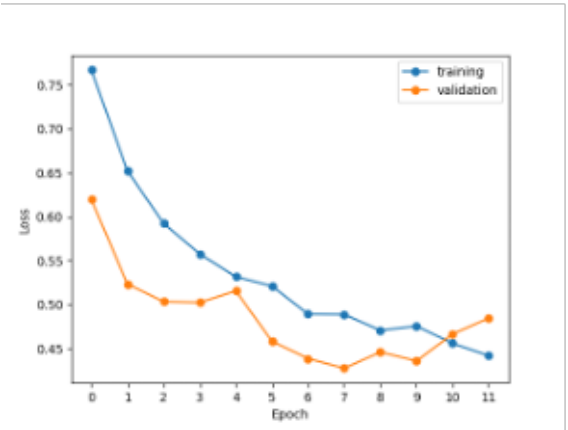

Figure 17: LPE Model Performance Using Enhanced Data

Moreover, our research supports the notion that incorporating adaptive attention mechanisms and learnable positional encodings can substantially improve performance in specialized tasks such as classifying disaster-related tweets. Notably, Model 4, embedded within our series of experiments, shone through by achieving the highest validation accuracy amongst its peers and demonstrating enhanced overfitting prevention. This finding highlights the transformative impact of Transformer-based architectures enhanced with learnable encodings on applications within disaster response scenarios.

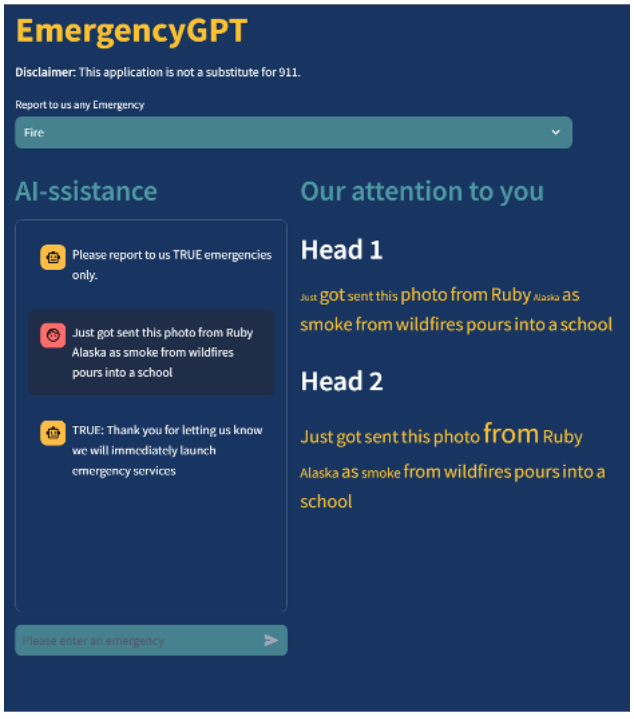

Figure 18: EmergencyGPT Application

The study also draws attention to the formidable capabilities of text enhancement methods, including pre-trained embedded vectors and agile transformer models. Impressively, the performance

of these methods is on par with that of the large language models in the context of the Kaggle competition. We achieved an accuracy of 84%, which secured us a position within the top 2% of the competition's leaderboard. Remarkably, with model sizes around 7.87 MB, our models are competitive in accuracy and sufficiently lightweight to ensure deployability across a wide range of platforms, demonstrating both efficacy and efficiency.

We completed this project by deploying our model via a web interface and trying it out with real word input (Figure 18). The model takes in the sentence from the user, tells the user if it is an actual disaster, and provides a formatted text whose word size represents the attention given to each word by each head.

The findings suggest an avenue for future research to extend these models to a broader array of emergencies, aiming to tailor and calibrate them for real-time application. The goal will be to fine-tune the models' interpretive accuracy and operational readiness, ensuring they are reliable tools for first responders and disaster management professionals in live scenarios. The quest to refine these models is more than an academic pursuit—it is a step towards harnessing the power of machine learning in service of global humanitarian efforts.

References

1. Vaswani, A., et al. (2017). *Attention Is All You Need.* Advances in Neural Information Processing Systems.
2. Pak, A., & Paroubek, P. (2010). *Twitter as a Corpus for Sentiment Analysis and Opinion Mining.* LREC.
3. Nguyen, D., & Gravel, R. (2018). *TweetEmo: Development and Validation of a Self-Report Scale for Measuring Emotions on Twitter.* PLoS ONE.
4. Rönnqvist, S., Sarlin, P., & Henriksson, R. (2019). *Banking on Twitter - Predicting market movements with sentiment analysis.* Journal of Computational Science.
5. S. Limboi and L. Dioşan, "A Lexicon-based Feature for Twitter Sentiment Analysis," 2022 IEEE 18th International Conference on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania, 2022, pp. 95-102 6.Kusrini and M. Mashuri, "Sentiment Analysis In Twitter Using Lexicon Based and Polarity Multiplication," 2019 International Conference of Artificial Intelligence and Information Technology (ICAIIT), Yogyakarta, Indonesia, 2019, pp. 365-368