Decoding Emotions: A Multi-Approach Sentiment Classification of Tweets

Bakiri Ayman, Ben Mohamed Nizar, Chahed Ouazzani Adam
CS-433 : Machine learning, EPFL

***Abstract*—This project investigates sentiment classification of tweets, focusing on predicting whether a tweet conveyed positive :) or negative :(emotions based on its text. Multiple approaches were explored, ranging from traditional machine learning methods such as TF-IDF and GloVe embeddings combined with logistic regression, to advanced deep learning models including FastText, DistilBERT, and RoBERTa. The study highlights the importance of robust preprocessing, efficient hyperparameter tuning using Optuna, and the integration of context-aware models to improve classification performance. The results demonstrate the trade-offs between simplicity and accuracy, with advanced models like DistilBERT achieving high validation accuracy of 88.7% and an F1 score of 88.9%. Ethical concerns such as the misclassification of sarcasm and irony were addressed, ensuring the model is both effective and responsible.**

I. Introduction

The objective of this project is to develop a sentiment classification model to predict whether a tweet previously contained a positive :) or negative :(smiley, based on the remaining text. Our planned methodology involves starting with basic approaches, such as GloVe embeddings or TF-IDF combined with logistic regression. Subsequently, we aim to explore more advanced methods, including FastText, and ultimately experiment with deep neural networks such as DistilBERT, leveraging cluster resources for training.

II. Exploratory Data Analysis

During the exploratory data analysis (**EDA**), we observed that the dataset is well-prepared for modeling, with no missing values and **balanced classes**. Each line in the dataset represents one tweet, making it straightforward to analyze.

We began by examining the **most common words, bigrams, and trigrams** in the positive and negative training data. Initially, many frequent terms such as "user," `<user>`, and even repetitive patterns like "user user user" were identified, alongside placeholders like "URL." These terms, while dominant, were deemed uninformative for sentiment classification. Hence, we removed them and reanalyzed the data for more meaningful insights.

Upon cleaning, the most common words in positive tweets included emotionally positive terms such as **"love," "good," "thanks," and "haha"**. In contrast, negative tweets highlighted terms like **"cant," "miss," and "really,"** reflecting more negative sentiments. Interestingly, some words such as "im" and "x" appeared frequently in both positive and negative contexts, indicating their **neutral or context-dependent nature**.

We also analyzed **hashtags**, which revealed insightful patterns. For instance, hashtags like `#sadtweet` appeared predominantly in negative tweets, while others like `#waystomakemehappy` were associated with positive contexts.

Further, **bigram and trigram analysis** revealed context-rich patterns such as *"thank you"* in positive tweets and *"want to go"* in negative ones.

III. Data Preprocessing

We employed light preprocessing for every method to remove irrelevant words identified during EDA, such as "user," "url," and "<user>" which were uninformative for sentiment classification. Removing these terms improved input quality and reduced noise, complementing the dataset's partially cleaned state.

Each method leveraged its own tokenization mechanism. Minimal preprocessing was applied for GloVe, as its embedding process handles text variability. In contrast, TF-IDF required additional steps to optimize feature representation.

GloVe-Specific Preprocessing

No further preprocessing was conducted for GloVe embeddings. Tweets were directly converted into feature vectors using pre-trained embeddings, which can be downloaded following the instructions provided in the README.

TF-IDF-Specific Preprocessing

To enhance feature representation for TF-IDF, we applied a more stringent preprocessing routine:

- Removed stopwords using NLTK's stopword list.
- Removed punctuation and normalized text to lowercase.
- Removed repeating characters (e.g., "heeellooo" $\rightarrow$ "hello") and numeric values.
- Tokenized the text into words.
- Applied stemming and lemmatization to reduce words to their base forms.

FastText, DistilBERT and RoBERTA

For advanced neural network-based methods, such as **FastText** and **DistilBERT**, minimal preprocessing was applied. These methods include their own tokenization processes:

- **FastText**: This method works efficiently with words split into subwords and does not require removing rare tokens.
- **DistilBERT**: Initial experiments with extensive text preprocessing (e.g., removing stopwords, punctuation) led

to poor performance with DistilBERT. This model inherently handle raw text effectively through it's tokenization mechanisms. Therefore, for these approaches, we decided to provide the **text data** directly to the tokenizer, retaining maximum contextual information.

IV. Methods

GloVe + Logistic Regression

To evaluate the effectiveness of GloVe embeddings in tweet sentiment classification, we utilized the pre-trained GloVe Twitter embeddings (glove.twitter.27B.100d.txt) with a Logistic Regression model from scikit-learn. The embeddings were first mapped to a 100-dimensional space, and the Logistic Regression model was trained with different regularization strengths using GridSearchCV, also from scikit-learn.

1) Methodology

Each tweet was represented as the mean of the GloVe word vectors for the tokens it contained. We employed a 5-fold cross-validation strategy, optimizing the regularization parameter C for the Logistic Regression model. The tested values of C ranged from 0.01 to 10, and the best model was determined based on cross-validated accuracy.

2) Results

The Logistic Regression model with GloVe embeddings achieved a highest validation accuracy of **76.2%**, with the optimal regularization parameter being $C = 1$. The Validation accuracy increased consistently as the regularization parameter grew from 0.01 to 1, plateauing beyond this point.

Training accuracy remained slightly higher than validation accuracy, indicating minimal overfitting.

To analyze the effect of embedding dimensions, we evaluated the Logistic Regression model with GloVe embeddings of different dimensions: 50, 100, and 200. The optimal regularization parameter $C = 1$, identified through GridSearchCV, was used for all configurations. Table I summarizes the validation accuracy and F1 scores for each embedding dimension.

TABLE I: Performance of Logistic Regression with GloVe Embeddings Across Different Dimensions

Embedding Dimension	Validation Accuracy (%)	F1 Score (%)
50d	74.5	74.8
100d	76.2	76.3
200d	77.6	77.2

The results show that increasing the embedding dimension improves both validation accuracy and F1 score. The 200-dimensional embeddings achieved the highest performance, suggesting that larger embeddings better capture semantic nuances in the text.

3) Discussion

The results indicate that increasing the regularization parameter C improves validation accuracy up to a certain point, stabilizing at $C = 1$. Training accuracy remained slightly higher than validation accuracy, indicating minimal overfitting and good generalization.

Embedding dimension significantly impacted performance. The default 100-dimensional embeddings balanced accuracy and efficiency, while higher-dimensional embeddings (200d) further improved accuracy by capturing richer semantic information. However, this improvement comes at the cost of greater computational complexity.

Overall, the combination of GloVe embeddings and Logistic Regression was effective for sentiment classification, serving as a robust baseline for comparison with more complex models.

TF-IDF + Logistic Regression

To evaluate the effectiveness of TF-IDF vectorization in tweet sentiment classification, we applied a Logistic Regression model using TF-IDF features. Each tweet was represented as a sparse vector generated by the TF-IDF transformation, capturing the importance of terms relative to the corpus.

1) Methodology

The Logistic Regression model was trained with different regularization strengths using GridSearchCV from scikit-learn. We conducted a 5-fold cross-validation to optimize the regularization parameter C, testing values ranging from 0.01 to 10. The model's performance was assessed based on cross-validated accuracy.

2) Results

The Logistic Regression model with TF-IDF features achieved a highest validation accuracy of **82.0%**, with the best performance observed at a regularization parameter of $C = 1$. The validation accuracy steadily improved as the regularization parameter increased from 0.01 to 1, stabilizing thereafter.

Training accuracy followed a similar upward trend, remaining consistently higher than validation accuracy by a small margin, indicating that the model was well-regularized and generalized effectively to unseen data.

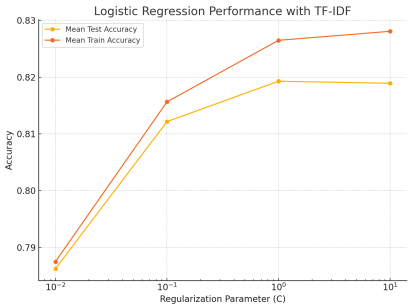

Fig. 1: Performance of Logistic Regression with TF-IDF features. The x-axis represents the regularization parameter (C), while the y-axis indicates the accuracy. A log-scale is used for C.

3) Discussion

The results demonstrate that validation accuracy increases with larger values of the regularization parameter C, peaking at $C = 1$, and stabilizing thereafter. The gap between training and validation accuracy is minimal, indicating that the model generalizes effectively.

The combination of TF-IDF vectorization and Logistic Regression provides a robust approach for tweet sentiment classification, delivering competitive accuracy and interpretability.

4) Comparison Between GloVe and TF-IDF Representations

TF-IDF outperformed GloVe embeddings in tweet sentiment classification, achieving higher validation accuracy across all regularization values. This suggests that TF-IDF's ability to capture corpus-specific term importance was more effective than the semantic relationships encoded in GloVe embeddings for this task. While GloVe offered competitive performance, its fixed, lower-dimensional representation may have limited its ability to fully capture dataset-specific nuances. These results highlight the importance of tailoring feature representations to the specific characteristics of the dataset.

FastText

1) Methodology

FastText, a word-embedding-based approach developed by Facebook AI, is highly efficient for text classification due to its use of subword embeddings and hierarchical softmax. We decided to switch from basic Logistic Regression to FastText as it captures subword information, making it more robust to spelling variations and rare words commonly found in tweets. To optimize its performance, we applied Optuna to tune the following hyperparameters:

- **Learning Rate**: Log-uniform sampling between 0.001 and 0.5.
- **Epochs**: Integer range between 5 and 100.
- **Word N-grams**: Integer range between 1 and 4.
- **Embedding Dimensions**: Categorical choice of $\{50, 100, 200, 300\}$.
- **Loss Function**: Categorical choice of `softmax`, `hierarchical softmax (hs)`, and `negative sampling (ns)`.

2) Results

After hyperparameter tuning, the best-performing configuration achieved the following:

- **Learning Rate**: 0.00347
- **Epochs**: 69
- **Word N-grams**: 4
- **Embedding Dimensions**: 50
- **Loss Function**: `softmax`

The model achieved:

- **Validation Accuracy**: 83.9%
- **F1 Score**: 84.0%

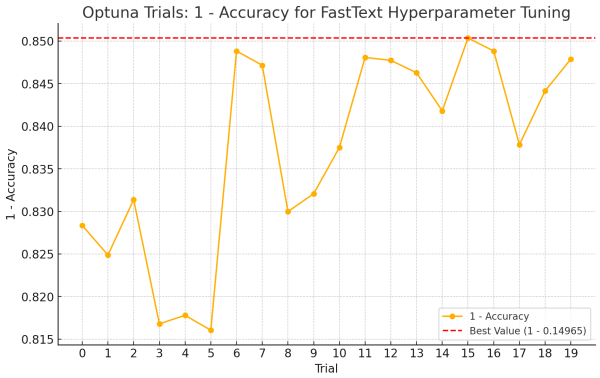

Fig. 2: Accuracy Scores Across Optuna Trials for FastText Hyperparameter Tuning

3) Discussion

FastText provided robust results, outperforming traditional models like Logistic Regression with GloVe embeddings. Its subword embeddings captured semantic relationships effectively. However, it fell slightly short of transformer-based models like DistilBERT and RoBERTa in terms of accuracy and F1 score. Despite this, its computational efficiency makes it a favorable choice for applications requiring low resource usage or rapid deployment.

DistilBERT

1) Methodology

DistilBERT, a distilled version of BERT, is a transformer-based model designed to retain 97% of BERT's performance while being 60% faster and requiring fewer computational resources. It achieves this by leveraging knowledge distillation during training, reducing the number of layers from 12 to 6 while maintaining the same hidden size and attention heads. DistilBERT was fine-tuned for sentiment classification using the Hugging Face `Trainer` API. The training process involved:

- Tokenizing tweets with a maximum length of 128 using DistilBERT's tokenizer.
- Fine-tuning the pre-trained `distilbert-base-uncased` model on the tweet data.
- Optimizing hyperparameters via Optuna.

2) Results

The optimal configuration for DistilBERT achieved:

- **Validation Accuracy**: 88.7%
- **F1 Score**: 88.9%

(Submission ID: 277867)

3) Discussion

DistilBERT offered a strong trade-off between accuracy and computational efficiency, achieving nearly the same performance as RoBERTa while being 60% faster and lighter. Its contextual understanding of text through WordPiece tokenization contributed to superior handling of nuanced tweets. The minimal preprocessing requirement further streamlined its integration into the sentiment classification pipeline.

Hyperparameter Tuning for DistilBERT

Hyperparameter tuning for DistilBERT was conducted using Optuna. The search space included:

- **Learning Rate**: Log-uniform sampling between 1×10^{-5} and 5×10^{-4}.
- **Batch Size**: {8, 16, 32}.
- **Weight Decay**: Log-uniform sampling between 0.01 and 0.1.
- **Number of Epochs**: {3 to 7}.

The best hyperparameters identified were:

- **Learning Rate**: 2×10^{-5}
- **Batch Size**: 16
- **Weight Decay**: 0.01
- **Number of Epochs**: 5

These hyperparameters allowed DistilBERT to achieve optimal results with validation accuracy and F1 scores of 88.7% and 88.9%, respectively.

TABLE II: Hyperparameter Tuning Results for DistilBERT

Learning Rate	Batch Size	Weight Decay	Epochs	Accuracy	F1 Score
1×10^{-5}	8	0.01	3	87.1%	87.3%
3×10^{-5}	16	0.05	5	88.2%	88.5%
2×10^{-5}	16	0.01	5	**88.7%**	**88.9%**
5×10^{-5}	32	0.1	7	87.8%	88.0%
4×10^{-5}	8	0.03	6	88.0%	88.2%

RoBERTa

1) Methodology

RoBERTa, an optimized variant of BERT, was fine-tuned for sentiment classification. The process included:

- Tokenizing tweets to a maximum length of 128 using RoBERTa's tokenizer.
- Training the `roberta-base` model for 5 epochs with a learning rate of 3×10^{-5} and batch sizes of 16 (training) and 32 (evaluation).

2) Results

The fine-tuned RoBERTa model achieved:

- **Validation Accuracy**: 88.4%
- **F1 Score**: 88.7%

(Submission ID: 277552)

3) Discussion

RoBERTa demonstrated high performance, comparable to DistilBERT, but required significantly more computational resources. Its improved pre-training strategy and larger dataset usage enhanced its contextual understanding, making it effective for nuanced tweet analysis. However, the increased training time may limit its applicability in time-sensitive scenarios.

Comparison Between FastText, DistilBERT, and RoBERTa

- **Accuracy and F1 Score**: DistilBERT achieved the highest validation accuracy of 88.7%, outperforming FastText (83.9%). RoBERTa matched DistilBERT but at a higher computational cost.
- **Efficiency**: FastText required significantly less training time and resources than transformer-based models.
- **Complexity Handling**: DistilBERT and RoBERTa excelled in capturing context and nuances, outperforming FastText on ambiguous or sarcastic tweets.
- **Resource Constraints**: FastText is ideal for lightweight tasks, while DistilBERT offers a balance of accuracy and efficiency. RoBERTa is suited for maximum accuracy regardless of resources.

Overall, DistilBERT provides the best trade-off between accuracy, efficiency, and resource usage, making it suitable for most applications. FastText excels in resource-constrained scenarios, while RoBERTa is best for tasks demanding maximum accuracy.

V. Scientific Novelty

This project addresses the challenges of sentiment classification in short and informal texts like tweets. These challenges include:

- **Ambiguity and Sarcasm**: Tweets often lack context, making it difficult for traditional models to understand nuanced expressions.
- **Noise in Text**: Tweets contain informal language, abbreviations, and symbols that complicate feature extraction.

Our approach introduced several innovations to tackle these issues:

- Leveraging **Optuna for Hyperparameter Tuning** ensured optimal configurations for all models, significantly improving performance.
- Employing **context-aware models** (DistilBERT and RoBERTa) enabled better handling of ambiguity and sarcasm.
- Demonstrating the **trade-offs between traditional and deep learning methods**, highlighting the strengths of TF-IDF for interpretable feature selection and the power of transformers for semantic understanding.

VI. Conclusion

In this project, we explored various approaches for sentiment classification of tweets, from traditional models like Logistic Regression with TF-IDF and GloVe embeddings to advanced deep learning models like FastText, DistilBERT, and RoBERTa.

The results highlight trade-offs between simplicity and complexity. TF-IDF and GloVe provided interpretable and efficient solutions, while DistilBERT achieved state-of-the-art performance with 88.7% validation accuracy and an 88.9% F1 score. FastText offered a good balance between simplicity and performance, outperforming traditional methods while maintaining efficiency.

Key factors included robust preprocessing, hyperparameter tuning with Optuna, and context-aware models for capturing semantic nuances. Future work could address sarcasm and contextual ambiguity, potentially leveraging multimodal data like images or user metadata to enhance classification.

Ethical Concerns

1) Overview of Risks

A significant risk in sentiment classification is the misclassification of tweets, particularly when the sentiment conveyed by smileys conflicts with the overall context or intent of the text. Examples include:

- **Sarcasm and Irony**: Tweets such as *"Oh great, everything's ruined :)"*, were often misclassified as positive due to the presence of a smiley, despite the underlying negative sentiment.
- **Ambiguity**: Tweets like *"I can't believe it worked!"* depend on context and were sometimes misclassified by traditional models like Logistic Regression.
- **Bias in Data**: Certain hashtags and phrases disproportionately influenced predictions due to their prevalence in the training data, potentially amplifying biases.

2) Impact of Misclassification

The misclassification of tweets can negatively affect:

- **Content Creators**: Misinterpretation of the author's intent can lead to misrepresentation in sentiment-based analytics.
- **End-Users**: Applications such as content moderation, sentiment tracking, or recommendations may produce misleading results, reducing user trust.

3) Evaluation and Metrics

To address these risks, we relied on the adoption of context-aware models (e.g., DistilBERT and RoBERTa) to better handle nuanced contexts like sarcasm and ambiguity. These models showed significant performance improvements over traditional approaches, as shown in Table III.

TABLE III: Performance Improvement with Context-Aware Models

Model	**Accuracy (%)**	**F1 Score (%)**
Logistic Regression (TF-IDF)	82.0	82.0
DistilBERT	88.7	88.9

4) Mitigation Strategies

To minimize misclassification risks, the following strategies were employed:

- **Advanced Models**: Context-aware models like DistilBERT and RoBERTa inherently address risks related to sarcasm and ambiguity by capturing semantic and contextual nuances.
- **Quantitative Validation**: Improvements in accuracy and F1 provided strong evidence of reduced misclassification rates.

5) Conclusion

The adoption of context-aware models mitigated many ethical risks by improving performance on nuanced text. Future work could explore multi-modal sentiment analysis, incorporating additional context from images or metadata to further enhance robustness.

Appendix

References

- **Scikit-learn**: Pedregosa et al., ”Scikit-learn: Machine Learning in Python”, Journal of Machine Learning Research, 2011. [Online]. Available: https://scikit-learn.org/stable/
- **FastText**: Grave et al., ”Learning Word Vectors for Sentiment Analysis”, Facebook AI Research, 2017. [Online]. Available: https://fasttext.cc/
- **Optuna**: Akiba et al., ”Optuna: A Next-generation Hyperparameter Optimization Framework”, 2019. [Online]. Available: https://optuna.org/
- **Transformers Library**: Wolf et al., ”Transformers: State-of-the-Art Natural Language Processing”, 2020. [Online]. Available: https://huggingface.co/transformers/
- **GloVe Embeddings**: Pennington et al., ”GloVe: Global Vectors for Word Representation”, 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove/
- **Datasets Library**: Hugging Face, ”Datasets: A Community Library for Natural Language Processing”, 2020. [Online]. Available: https://huggingface.co/docs/datasets/
- **NLTK (Natural Language Toolkit)**: Bird et al., ”NLTK: The Natural Language Toolkit”, 2009. [Online]. Available: https://www.nltk.org/

DIGITAL ETHICS CANVAS

CONTEXT

Sentiment analysis of tweets with 1 positive and 0 negative

SOLUTION

a context-aware sentiment classification model like distilBERT

BENEFITS

improves sentiment detection accuracy, provide insights into user opinions for researchers, analysts, and companies.

WELFARE

RISK

- Can the solution be used in harmful ways, in particular with regards to vulnerable populations?
- What kind of impacts can errors from the solution have?
- What type of protection does the solution have against attacks or misuse?

Misclassification of sarcastic or ironic tweets leads to misleading sentiment results

MITIGATION

- Use distilBERT to capture context beyond simple words

- Perform error analysis to identify and address ambiguous examples

FAIRNESS

RISK

- How accessible is the solution?
- What kinds of biases may affect the results ?
- Can the outcomes of the solution be different for different users or groups?
- Could the solution contribute to discrimination against people or groups?

Underrepresented groups or dialects could have higher misclassification rates

MITIGATION

conduct bias testing on underrepresented contexts or languages

balance training data and improve preprocessing to ensure fairness across groups

AUTONOMY

RISK

- Can users understand how the solution works and what its limits are?
- Are users able to make choices (e.g. consent, settings) in their use of the solution and how?
- How does the solution affect user autonomy and agency?

users may not understand how the model makes predictions, reducing trust and transparency

MITIGATION

Implement explainability tools to clarify predictions

PRIVACY

RISK

- What data does the solution collect
- Is it collecting personal or sensitive data
- Who has access to the data?
- How is the data protected?
- Could the solution disclose / be used to disclose private information?

- Although tweets are public, text may contain indirect personal identifiers.

- Unauthorized access to tweet data could pose risks.

MITIGATION

- Ensure all tweets are anonymized during preprocessing

- restricts data access to authorized personnel only and store data securely

SUSTAINABILITY

RISK

- What is the carbon footprint of the solution?
- What types of resources does it consume (e.g. water) -and produce (e.g. waste)?
- What type of human labor is involved?

Training and fine-tuning large models like distilBERT consume significant computational ressources and energy.

MITIGATION

- Use pretrained model to minimize training time.

- Optimize resource usage by leveraging efficient hardware and scheduling runs during off-peak hours for the cluster.

Digital Ethics Canvas 2024 - C. Hardebolle, V. Macko, V. Ramachandran, Y. Do, N. Bartlomiejczyk, A. Holzer, P. Jermann