Sentiment and Company Classification

Group 69

Gustav Enocson and Adyuth Hisham

November 6, 2025

***Abstract*—This study investigates the classification of financial tweets about publicly traded companies, focusing on two sub-tasks: sentiment analysis and company identification. The motivation stems from the significant impact that tweets from influential individuals or large volumes of similar tweets can have on stock market dynamics. We explored two approaches to address these tasks. Approach A involved fine-tuning two pre-trained models—FinBERT for sentiment analysis and BERTweet for company identification—and combining their outputs using a fusion mechanism. Approach B extended a pre-trained FinTwitBERT model by adding a dual-head architecture, enabling simultaneous sentiment and company classification with a single encoder. One head was fine-tuned for sentiment analysis, while the other was trained from scratch for company identification. Using two financial tweet datasets, we evaluated both approaches. The individual models in Approach A achieved an accuracy of 80% for sentiment analysis and 67% for company identification. The dual-head model in Approach B yielded slightly higher accuracies, reaching an accuracy of 83% for sentiment analysis and 68% for company identification. These results demonstrate the potential of multi-task learning for financial tweet analysis, offering insights into scalable methods for monitoring social media's influence on markets.**

I. Introduction

IN today's financial markets, social media platforms such as X (formerly Twitter) play a major role in shaping investor sentiment and influencing stock prices. With the help of social media, large groups of people—and especially highly influential individuals—can cause significant market fluctuations. Being able to understand the content and sentiment of financial tweets can therefore provide valuable insights for traders, analysts, and automated systems. This project aims to classify tweets about publicly traded companies through two sub-tasks: sentiment analysis and company classification.

Two approaches were investigated. Approach A involved fine-tuning two pretrained models and combining them through a helper function. For sentiment analysis, the FinBERT [1] model was used, and for company classification, BERTweet [2] was applied. Each model was trained on task-specific datasets after extensive preprocessing and label normalization. Approach B developed a multi-task model extending the pre-trained FinTwitBERT [3] model. The model employs a multi-task design with a shared encoder and two heads: the original head that gets fine-tuned for the sentiment analysis and the newly added company-classification head trained from scratch.

Both approaches used the same preprocessed data, with the only difference being that for Approach B the datasets were merged. During pre-processing, the data was cleaned, balanced, and normalized by removing mentions, hashtags, URLs, and emojis. Rare companies were grouped into one category called "Other," and the sentiment labels were standardized into three classes. This setup made it possible to directly compare the two models after training.

II. Related Work

Knowledge Assembly Due to the specificity of this task, it is quite difficult to identify a dataset labeled for all tasks, and generating a new dataset is time-consuming. Proposed in the paper [4] is a training strategy that leverages unlabeled data using semi-supervised learning, without requiring any additional network designs.

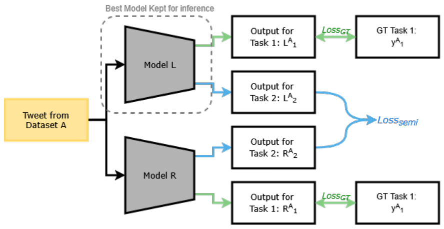

Best Model Kept for inference
Tweet from Dataset A
Model L
Model R
Output for Task 1: L^A_1
Output for Task 2: L^A_2
Output for Task 2: R^A_2
Output for Task 1: R^A_1
$Loss_{GT}$
$Loss_{GT}$
$Loss_{semi}$
GT Task 1: y^A_1
GT Task 1: y^A_1

Fig. 1: KA Method Overview.

During training, two copies of the same model, L and R, are initialized. Each mini-batch, containing samples from datasets A and B, is fed through the networks to predict tasks T1 and T2. If a sample has a ground-truth (GT) label, the supervised loss L_{GT} is applied; otherwise, the semi-supervised loss L_{semi} enforces consistency between the two networks. Figure 1 shows the forward pass for a sample from dataset A (yellow) with

GT labels only for task T1 (green). During inference, the better-performing model is retained.

III. Method

A. Overview

Two approaches were designed to tackle the task of sentiment and company classification of financial tweets. Approach A used two separate pre-trained models and combined them through a helper function while approach B implemented a unified multi-head architecture based on one pre-trained model, capable of performing both tasks simultaneously through a shared encoder.

B. Dataset

Datasets

Two public datasets were used in this study: one for sentiment analysis and one for company classification.

Financial Tweets (Hugging Face): [5] This dataset consists of tweets from financial influencers, originally containing approximately 315,000 tweets. For sentiment analysis, the dataset was pruned to 44,426 entries. Within the sentiment set, about 44% of labels were Bullish, 30% Neutral, and 26% Bearish.

Stock Tweets (Kaggle): [6] This dataset contains 64,479 tweets related to the top 25 most-watched stock tickers on Yahoo Finance, collected between 30-09-2021 and 30-09-2022. It was used for company classification. For the company labels, approximately 46% referred to Tesla, 13% to TSM, and the remaining $\sim 41\%$ to other companies.

C. Data Preprocessing

The preprocessing aimed to clean and standardize the two datasets while mitigating class imbalance.

For the sentiment dataset, all columns except `description` and `sentiment` were removed, and tweets labeled differently from Bearish, Neutral, or Bullish were filtered out. The remaining sentiment labels were mapped to integers: Bearish = 0, Neutral = 1, and Bullish = 2. For the company dataset, all columns except `tweet text`, `stock name`, and `company name` were dropped. To create a unique label for each company, a composite key combining the stock name and company name was generated.

Since both datasets were imbalanced, a custom helper function was implemented to produce roughly balanced datasets. The function identifies all unique classes and samples an equal number of rows from each, without replacement. The sampled rows are combined and shuffled to form the final dataset. Companies appearing fewer than 200 times were grouped into a single class labeled “Other.” The remaining companies, as well as the “Other” category, were assigned numeric labels using a `label2id` mapping. This step helped stabilize the model and prevent overfitting to small classes.

After these steps, the column names in both datasets were standardized to `text` and `label` for compatibility with the training pipeline. For the multi-head model, the two datasets were merged into a single table with columns `text`, `label_sen`, and `label_class`, where each row contained exactly one label. This setup allowed the shared encoder to process all examples while maintaining task separation at the head level.

Finally, all tweets were normalized using a function that removed mentions, hashtags, URLs, and emojis while preserving the textual content. This ensured that the models could focus on the semantic meaning of the tweets. The resulting datasets were clean, balanced, and suitable for fine-tuning.

D. Model Architecture

1) Approach A: This approach combines two task-specific transformer models trained independently: FinBERT for sentiment classification and BERTweet for company classification. For both models, a similar training pipeline was followed. The preprocessed data was first split into training and validation sets using an 80/20 ratio while maintaining class proportions through stratification. Model-specific tokenizers were then loaded to tokenize the respective datasets. Tokenization was performed with truncation and padding, capping sequences at 96 tokens—a length sufficient to cover most tweets while maintaining computational efficiency. The tokenized outputs were wrapped in simple PyTorch Dataset objects to facilitate loading during training.

Both model backbones were initialized from pre-trained checkpoints and configured with consistent label mappings. All layers were unfrozen to enable full fine-tuning, allowing the models to adapt to the financial tweets. Training was conducted using Hugging Face’s Trainer API with the AdamW optimizer and a cosine learning-rate scheduler to gradually decay the learning rate over time. The Trainer uses the default loss function, CrossEntropyLoss, which is suitable for single-label, multi-class classification. For FinBERT, the learning rate was set to 2e-5, with a weight decay of 0.01, a warm-up ratio of 0.1, and a maximum of 10 epochs, stopping early if the F1-macro score did not improve for two consecutive epochs. For BERTweet, the same learning rate and weight decay were used, with a warm-up ratio of 0.06, a label smoothing factor of 0.1 applied to to cross-entropy loss, and gradient accumulation over two steps to effectively increase the batch size without exceeding memory limits. Training was allowed to continue for up to 15 epochs, with early stopping triggered after four epoch if no improvement in F1-macro score.

The F1-macro score was chosen as the primary optimization metric to prevent overfitting and encourage balanced performance across classes. The batch size was set to 16 for both training and evaluation. The batch size was chosen as a balance between training stability, generalization, and hardware limits. Finally, a helper function was implemented to combine the two fine-tuned models, enabling sentiment analysis and company classification of tweets.

2) Approach B: The model is based on a pre-trained transformer architecture for sequence classification, specifically a FinTwitBERT model [3] fine-tuned for sentiment analysis. To enable multi-task learning, a parallel classification head was added to predict company labels alongside sentiment.

Shared Encoder: The core of the model is a pre-trained BERT model, which serves as a shared encoder for both tasks. Input tweets are tokenized and passed through BERT to obtain contextualized embeddings. The embedding corresponding to the `[CLS]` token is used as a fixed-length representation of the entire sequence.

Sentiment Head: The original classification head of the pre-trained BERT model is used to predict sentiment labels (*bearish*, *neutral*, *bullish*). The number of output neurons in this head is set to 3 to match the sentiment classes. This head remains trainable and leverages the pre-trained BERT representations.

Company Classification Head: A new parallel head, `HeadNN`, was introduced to perform company classification. This head consists of:

- A fully connected layer mapping the BERT embedding to 1024 hidden units.
- A ReLU activation function for non-linearity.
- A dropout layer with a probability of 0.2 for regularization.
- A final fully connected layer mapping to the number of company classes.

The output of this head provides logits for each company class.

Forward Pass: During the forward pass, input token IDs and attention masks are passed through the shared BERT encoder. The `[CLS]` token embedding is extracted and fed into both the sentiment and company classification heads simultaneously. The model returns two sets of logits: one for sentiment prediction and one for company classification.

This architecture allows the shared encoder to learn generalized textual representations while maintaining task-specific outputs for both sentiment and company classification, enabling efficient multi-task learning on the merged dataset.

Training: To implement Knowledge Assembly, two models were initialized with different random weights.

To fine-tune the new company classification head, the sentiment head and the shared BERT backbone were frozen, and only the classification head was trained. An Adam optimizer with a learning rate of 5×10^{-3} was used, and training was performed for 30 epochs with a batch size of 64.

The loss was calculated using cross-entropy and during training, checkpoints were saved, and the model with the minimal validation loss was selected for subsequent steps.

Separate optimizers were used for each model to update only the parameters of the classification head, and learning rate schedulers with linear warm-up ensured stable training.

This setup allowed the classification head to adapt quickly to the new task while keeping the pretrained sentiment head and backbone weights fixed, stabilizing training and preventing overfitting.

Using the best models trained for both instances, the sentiment head and backbone is unfrozen. Both instances are now trained with Knowledge Assembly.

The multi-task semi-supervised consistency loss combines supervised and consistency terms for both sentiment and company classification. For the supervised loss, cross-entropy is used, while for the consistency loss, Kullback-Leibler (KL) divergence is used to encourage both models to be consistent. Here, i denotes data from Dataset A and j denotes data from Dataset B. L and R are the two instances of the model, and Task 1 and Task 2 are represented as 1 and 2, respectively.

$$\mathcal{L} = \mathcal{L}_{\text{GT}} + \lambda \, \mathcal{L}_{\text{semi}}, \tag{1}$$

where λ balances the contribution of the semi-supervised component.

Supervised Loss ($\mathcal{L}_{\text{GT}}$): For labeled examples, cross-entropy loss is computed for both models and both tasks:

$$\begin{aligned}\mathcal{L}_{\text{GT}} = \sum_{i \in \mathcal{A}} & \text{CE}(L_1^i, Y_1^i) + \text{CE}(R_1^i, Y_1^i) \\ & + \sum_{j \in \mathcal{B}} \text{CE}(L_2^j, Y_2^j) + \text{CE}(R_2^j, Y_2^j).\end{aligned} \tag{2}$$

Consistency Loss ($\mathcal{L}_{\text{semi}}$): For both labeled and unlabeled data, symmetric KL divergence enforces agreement between model predictions:

$$\begin{aligned}\mathcal{L}_{\text{semi}} = \sum_{i \in \mathcal{A}} & \text{KL}(L_1^i \,||\, R_1^i) + \text{KL}(R_1^i \,||\, L_1^i) \\ & + \sum_{j \in \mathcal{B}} \text{KL}(L_2^j \,||\, R_2^j) + \text{KL}(R_2^j \,||\, L_2^j).\end{aligned} \tag{3}$$

This formulation allows the model to leverage both labeled and unlabeled data while maintaining task-specific predictions that are consistent across the two model instances.

To train the multi-task model, both classification heads were updated with a learning rate of 1×10^{-4}, while the shared backbone was updated with a learning rate of 1×10^{-5}. Training was performed for 20 epochs with a batch size of 32. The semi-supervised consistency loss (Eq. 1) was used with λ set to 1.0.

IV. Results

The models were evaluated on the test set using accuracy, F1-weighted and F1-macro scores. These were chosen since they a good insight to the performance. Accuracy shows the overall correctness, F1-weighted accounts for class imbalance and F1-macro ensures balanced evaluation across all classes.

Model	Accuracy	F1-Macro	F1-Weighted
FinBERT	0.80	0.80	0.80
Multi-Task Model	0.83	0.83	0.83

TABLE I: Performance on Sentiment Analysis

Model	Accuracy	F1-Macro	F1-Weighted
BERTweet	0.60	0.67	0.60
Multi-Task Model	0.68	0.68	0.65

TABLE II: Performance on Company Classification

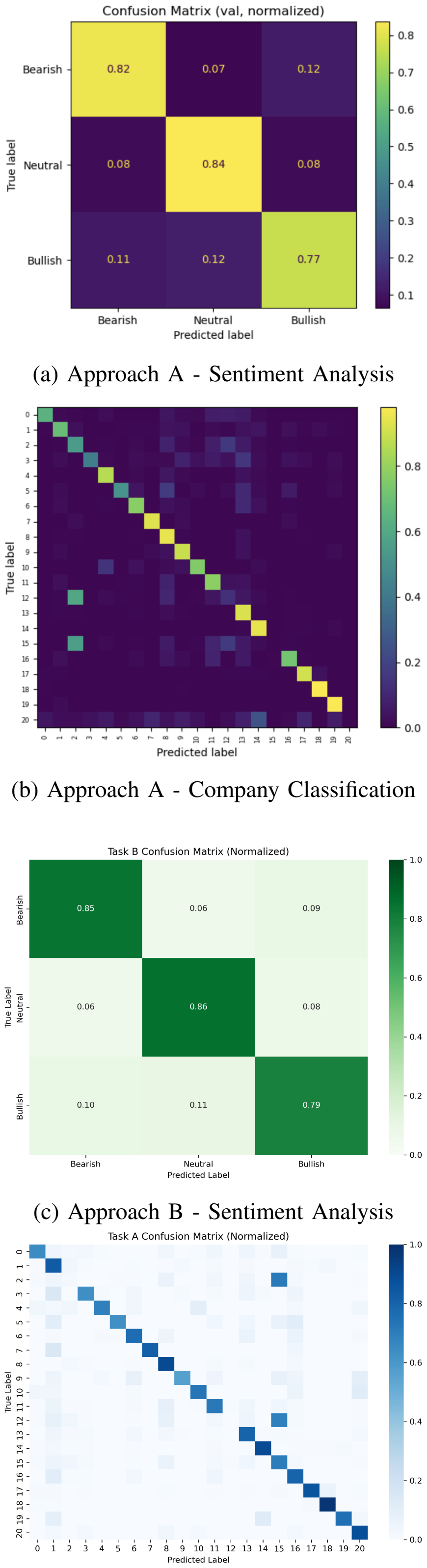
For Sentiment Analysis (I), the multi-task model performed 0.03% better, and for Company Classification (II), it performed 0.08% better in terms of accuracy. We can see that there is a marginal increase in performance for the multi-task model, but this improvement is quite minimal.

V. Conclusion

In this work, we tested two approaches for performing sentiment analysis and company name classification using two disjoint datasets. Both approaches were able to perform both tasks sufficiently well, with the multi-task model performing slightly better. To create a more robust system, Named Entity Recognition (NER) could be used to identify company names and integrate the current models for stock prediction.

References

[1] ProsusAI, “Finbert: Financial sentiment analysis with bert.” https://huggingface.co/ProsusAI/finbert, December 2020. Pre-trained BERT model for financial sentiment analysis, released by ProsusAI on Hugging Face.



(a) Approach A - Sentiment Analysis

(b) Approach A - Company Classification

(c) Approach B - Sentiment Analysis

(d) Approach B - Company Classification

Fig. 2: Heatmaps for Approaches A and B across Sentiment Analysis and Company Classification tasks.

[2] D. Q. Nguyen, T. Vu, and A. T. Nguyen, “Bertweet: A pre-trained language model for english tweets.” https://huggingface.co/docs/transformers/model_doc/bertweet, October 2020. Available on Hugging Face; original paper at arXiv:2005.10200.

[3] S. Akkerman, “Fintwitbert.” https://huggingface.co/StephanAkkerman/FinTwitBERT, September 2024. Hugging Face model repository.

[4] F. Spinola, P. Benz, M. Yu, and T. Kim, “Knowledge assembly: Semi-supervised multi-task learning from multiple datasets with disjoint labels,” *arXiv preprint arXiv:2306.08839*, 2023.

[5] S. Akkerman, “Financial tweets datasets at hugging face.” https://huggingface.co/datasets/StephanAkkerman/financial-tweets, February 2024.

[6] Kaggle, “Stock tweets for sentiment analysis and prediction.” https://www.kaggle.com/datasets/equinxx/stock-tweets-for-sentiment-analysis-and-prediction, December 2022.