Exploring financial sentiment analysis with the Financial Phrasebank dataset

Hugo Veríssimo
Complements of Machine Learning 24/25
University of Aveiro
Aveiro, Portugal
hugoverissimo@ua.pt

João Cardoso
Complements of Machine Learning 24/25
University of Aveiro
Aveiro, Portugal
joaopcardoso@ua.pt

***Abstract*—abstratotootototot**

Keywords: key, word, number, 1

I. Introduction

With the ever increasing volume of information created and distributed by the minute, it is more important than ever to have access to fast and reliable analysis of any information we may come across. Especially with the democratized access to financial instruments and capital markets, where individuals have the possibility to invest in virtually any company on the stock exchange, it is important to have ways to leverage against giant institutions with hundreds of financial analysts at their disposal.

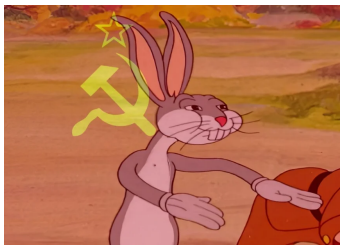

Fig. 1: Power to the people, colorized (circa 1917).

Historically, financial analysis relied heavily on fundamental analysis (examining earnings, balance sheets, annual financial reports), which required extensive knowledge in the field (also the strategy that made Warren Buffett one of the richest men in the world), along with technical analysis (studying price and volume trends). Around 2010, after the 2008 global financial crisis, there was a surge in news analysis to evaluate the tone and derive investment strategies from it. Due to the lack of domain specific lexicon these analysis were falible, until the work by Loughran and McDonald was published, a financial lexicon based on 10-K forms (i.e., annual financial reports) and dictionaries. This allowed to use more sofisticated analysis rather than using the presence of negative words as a signal to sell.

Upon the launch of Twitter, information streams increased dramatically, making more and more data available for analysis. But, machine learning was not heavily used, as most data was not annotated, or there was very little data with high-quality annotations. In 2014, P. Malo *et al.* published a fundamental dataset for financial sentimental analysis, that is still used, the Financial Phrasebank. It is unique, for the inclusion of important aspects as directional expressions (e.g., profits decreased), entity polarity shifts (e.g. profits may be negative if decreased), and phrase level context.

With this, machine learning models started finding their place in research, as the field of natural language processing (NLP) grew and niche fields such as financial investments found useful data. In this work we explore the Financial Phrasebank dataset, by implementing different machine learning and deep learning models to evaluate the sentiment of sentences related to financial news.

II. State of The Art

The field of NLP has grown drastically in the past decade, progressing from recurrent neural networks (RNN) and related models such as Long-Short Term Memory (LSTM), to the transformers-type models, large language models (LLM) and text generative models as ChatGPT. With the Financial Phrasebank the field expanded into financial analysis, with several works of relevance being published in recent years.

In the work of Araci (2019), the author developed a BERT-based model trained specifically on texts with financial data. BERT, Bidireccional Encoder Representations from Transformers, is a large language model developed by Google (2018), benefitting largely from the fact that it can ”hold” in memory large chunks and in both directions, simultaneously. The fact that it is built on the Transformer encoder architecture, it can weigh the importance of different words in a sentence by using a self-attention. The model is pre-trained on large unlabelled corpora (e.g., Wikipedia, BookCorpus), and can be fine-tuned for specific purposes. In this work, the end model was trained on domain specific corpus such as TRC2-financial data and financial specific texts (over 440 000 sentences), and then fine tuned with the Financial Phrasebank. The model achieved an accuracy of 97% on the Financial Phrasebank dataset with 100% agreement, but 86% and 85% respectively on the dataset with all levels of agreement (the

agreement will be further detailed in the Methodology section). Later on the model was further improved by Sun *et al.* (2025), EnhancedFinSentiBERT, by including dictionary embeddings, expanded corpus that deversified the pre-training stage drastically, and a novel neutral sentiment module, that further enhanced the distinction between neutral and weak sentiments, resulting in improved accuracy (89%) and F1-score (88%). The pre-training stage benefited from the large diversity of the corpus, going from a few million tokens to 2.4 B tokens with the latest version. In a similar direction, but at the fine tuning level, Atsiwo (2024) improved the data used in fine tuning, considering that most datasets have relatively short sentences ($<$ 100 tokens), failing to leverage the full context window of LLMs like BERT (512 tokens). This was achieved by augmenting the training data with synthetic sentences generated by GPT-4, with accuracy of 89% and F1-score of 88% for 50% agreement dataset.

GPT has been used with different purposes, as in the work by Zhou *et al.* (2023), where GPT-3.5Turbo was used for zero-shot sentiment classification. However, it performed worse than finely tuned models (accuracy of 75% and F1-score of 74%).

The BERT model was revisited by different researchers, but a new iteration from Facebook AI was proposed (2019) named RoBERTa (Robustly Optimized BERT Approach) was developed, that used a signficantly larger corpus for training (10x larger), and a dynamic masking technique during training, that allowed the model to learn new contextual relations while using the same sentences, making it more robust. This model was used as BERT to develop financial models, where the work Choe *et al.* (2023) is worth mentioning, where a large corpus of financial texts were fed to the model for training, from a range of sources (e.g., Reuters, SEC fillings, EIA). The model (FiLM, Financial Language Model) benefited from the diversity of training data, rather than simply focusing on fine tuning with highly curated data, showing improved generalization and better metrics than FinBERT and RoBERTa (accuracy 88%, F1-score 87%). These models are improving substantially over the years, but it is different to put them to test against a controlled dataset from using them in real life, and the variety included as consequence. Competitions such as FinNLP help drive research in this field, by posing ever more diversified test sets, aiming to improve the robustness of models, and the solutions developed by the researchers.

III. Methodology

To address the problem of sentiment classification in sentences related to financial investment, we set up a pipeline for training and testing using the Financial PhraseBank for three types of models, where the best was further explored and tuned for different tests. The dataset and setup is detailed in the following sections.

A. Dataset

The Financial PhraseBank is a widely used benchmark dataset for financial sentiment analysis. It consists of roughly 4,840 English sentences (mostly news headlines or short statements) about companies, drawn from financial news articles and press releases. Each sentence is labeled with one of three sentiment classes – positive, negative, or neutral – representing the sentence's sentiment from the perspective of an investor [1], [2].

TABLE I: Financial PhraseBank distribution. Four possible sets within the dataset, depending on how many financial experts agreed with the attributed label. The dataset with 50% agreement corresponds to the entire dataset.

Sentiment	**Agreement**			
	50%	**66%**	**75%**	**All**
Negative	604	514	420	303
Neutral	2879	2535	2146	1391
Positive	1363	1168	887	570
Total	4846	4217	3453	2264

The dataset was labeled by 16 finance professionals, each responsible for labelling a subset of sentences. Each sentence was labelled by 5-8 annotators, and the resulting agreement score was a result of the fraction of annotators that labelled the sentence in the same manner. This resulted in 4 different subsets, where 50% agreement corresponds to the entire dataset, with the dataset size decreasing as the level of agreement increased. It is important to mention that the agreement level corresponds to the least allowed, so the 50% agreement level dataset contains the other subsets. The dataset sizes and class proportion can be consulted in Table I, along with sample sentences and the attributed sentiment classification in Table II.

This subset strategy allows researchers to find a balance between the amount and the quality of data, representing a common trade-off in the field of machine learning.

TABLE II: Example sentences from the Financial PhraseBank with annotated sentiment labels and annotator agreement levels.

Sentence	**Sentiment**	**Agreement level (%)**
According to Gran, the company has no plans to move all production to Russia, although that is where the company is growing.	Neutral	100%
The fair value of the company 's investment properties went down to EUR 2.768 billion at the end of 2009 from EUR 2.916 billion a year earlier.	Negative	75%
Basic banking activities continued as normal .	Neutral	66%
In banking , Sampo A was unchanged at 14.24 eur and Nordea rose 0.42 pct to 9.51 eur.	Positive	50%

B. Exploratory Data Analysis

Taking into consideration the different possible subsets, we selected the one with 75% agreement, as it is a balance

between quantity and quality in the dataset, also taking into consideration the proportion between classes.

In Fig. 2 the number of sentences per class is evidence of how imbalanced the dataset is. As a result, we had to balance the dataset by undersampling all classes to the amount of examples for the 'negative' class, keeping 336 sentences per class.

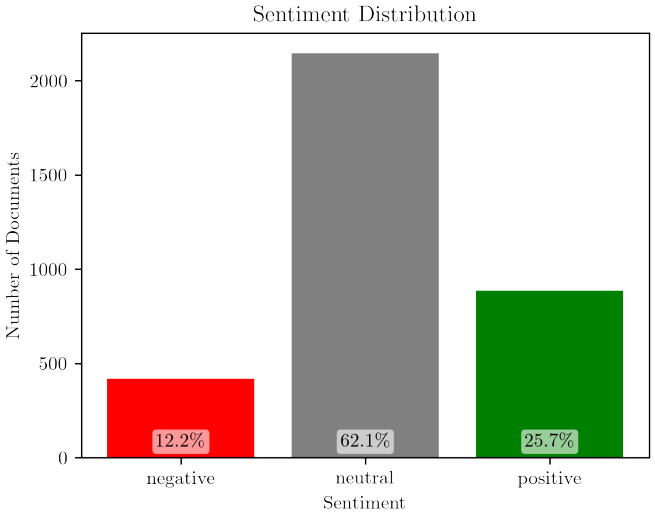

Fig. 2: Class distribution in the 75% agreement dataset.

TThe frequency distribution of document lengths helped determine the maximum number of tokens to use (considering the limit is 512 for BERT), as shown in Fig. 3.

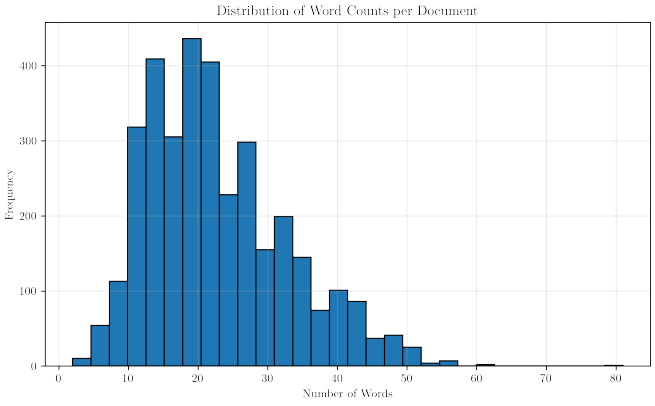

Fig. 3: Word count distribution per document for the 75% agreement dataset.

Fig. 4: Most frequent words in the dataset, visualized as a word cloud.

To identify words that are most indicative of sentiment class, we compared word frequencies across classes. Terms such as *down*, *decreased*, *profit*, *fell*, and *rose* showed significantly higher frequency in specific sentiment categories, making them particularly relevant for classification.

C. Preprocessing

The selected dataset (75% agreement) was split 80/20 for training/testing. The testing dataset was unique, meaning that all the sentences present in this subset were removed from any other dataset (of all the possible agreement levels), to prevent data leakage.

The preprocessing changed slightly between models, and is adequately detailed in their section. The preprocessing here mentioned was carried before any model training.

D. Model Evaluation and Validation Strategy

Prior to the model training we have performed 5-fold cross validation for hyperparameter training, followed by training on the full dataset. The models were continuously evaluated in terms of their learning curves and comparative metrics (i.e., confusion matrix, accuracy).

IV. Model Architectures: BERT, LSTM, and FastText

With the maining goal of classifying the sentiment of sentences, we have selected three models for an initial assessment, and proceeded with more complex iterations on the best model. The models selected were: Long Short-Term Memory (LSTM), fastText, and BERT.

1) Long Short-Term Memory: LSTM is a type of recurrent neural network (RNN), introduced by Hochreiter and Schmidhuber (1997). Its architecture was though to solve the vanishing gradient problem in most RNN, by introducing memory cells and gating mechanisms (input, output, and forget gates), to retain long term relations in sequential data (such as time series, or sentences).

2) fastText: Developed by Joulin *et al.* at Facebook AI (2016), fastText is built on the Word2Vec (word representation in a vectorial space) and extended it by incorporating subword information. Rather than representing each word as a single entity, it breaks it down to character n-grams. This allows to

represent sentences by averaging word embeddings, making it very lightweight and fast to train on large datasets, with minimal tuning. The lightweightness and little tunability makes it less differenciate and harder to adapt to specific cases.

3) Bidireccional Encoder Representations from Transformers: BERT was introduced by Devlin *et al.* and colleagues at Google (2018), and is a deep-transformer model pre-trained on large corpora using masked language modeling (hiding one word in the sentence for the model to predict) and next sentence prediction. BERT is capable of considering both forwards and backwards dependencies with a word, simulatenously. This allows for much better understanding of nuanced language patterns and semantics. Despite the higher computational requirements, it still is manageable at a local level, and benefits heavily from fine tuning for specific NLP tasks.

A. Initial benchmark

The initial models went through rounds of 5-fold cross validation, with the hyperparameter search spaces as indicated in Table III.

TABLE III: Hyperparameter search space for the initial models.

Hyperparameter	Possible Values
Epochs	$\{20, 21, \ldots, 99\}$
Learning rate	$[10^{-5}, 10^{-2}]$
Embedding dimension	$\{100, 200, 300\}$

(a) fastText hyperparameters.

Hyperparameter	Possible Values
Epochs	$\{2, 3, 5, 8, 10, 15\}$
Learning rate	$[10^{-5}, 10^{-3}]$
Embedding dimension	$\{32, 64, 128, 256\}$
LSTM units	$\{32, 64, 128, 256\}$
Dropout	$[0, 0.5]$
Recurrent dropout	$[0, 0.5]$

(b) LSTM hyperparameters.

Hyperparameter	Possible Values
Epochs	$\{1, 2, 3, 4, 5\}$
Learning rate	$[10^{-5}, 10^{-2}]$
Weight decay	$[0, 0.5]$

(c) BERT hyperparameters.

After fitting the models with the best hyperparameters, the models were trained on the complete training set, with the results in Table IV.

TABLE IV: base model metrics across models

Model	Accuracy		F1 (macro)	
	Train	Test	Train	Test
fastText	0.54	0.65	0.45	0.44
LSTM	0.67	0.66	0.63	0.63
BERT	0.97	0.92	0.97	0.91

From the results, there is a clear gap between the models, with BERT being the best, followed by LSTM and fastText. These are aligned with the literature, although LSTM can perform better if we consider bi-LSTM, but the purpose here was to evaluate *vanilla* models as an initial assessment.

The hyperparameters for BERT were: number of training epochs, 2; learning rate, 0.0001; and weight decay, 0.1.

From the learning curve (Fig. 5) the model seems to learn well, albeit the validation loss does increase slightly towards the end, overlapping the training curve.

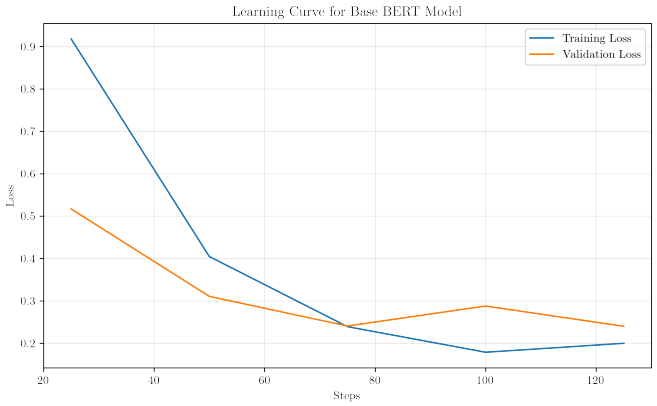

Fig. 5: learning curve

From the confusion matrix and training metrics (Fig. 6 and Table V)

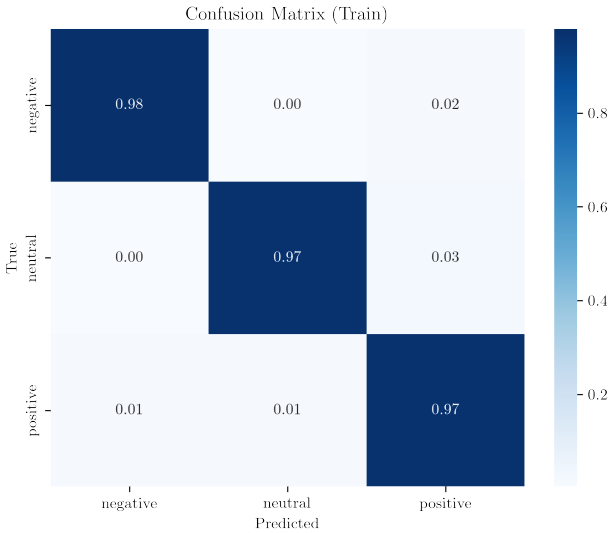

Fig. 6: base bert conf matri train

TABLE V: Classification report for BASE BERT on training data.

Class	Precision	Recall	F1-Score	Support
Negative	0.98	0.98	0.98	336
Neutral	0.98	0.97	0.98	336
Positive	0.96	0.97	0.96	336
Accuracy			0.97	1008
Macro avg	0.97	0.97	0.97	1008
Weighted avg	0.97	0.97	0.97	1008

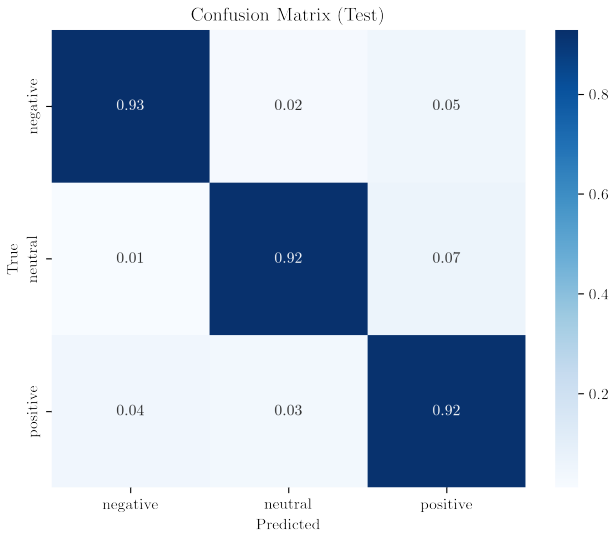

Fig. 7: base bert cong matrix test

TABLE VI: Classification report for BASE BERT on test data.

Class	Precision	Recall	F1-Score	Support
Negative	0.86	0.93	0.89	84
Neutral	0.98	0.92	0.95	429
Positive	0.84	0.92	0.88	178
Accuracy			0.92	691
Macro avg	0.89	0.92	0.91	691
Weighted avg	0.93	0.92	0.92	691

ligeiro overfitting e tais

B. data augmention model

para esta abordagem foi usada online data augmention nas várias frases do dataset de treino do 75Agree,, como o bert pela literatura e como pela nossa observacao tem uma performance signigicativamente melhor, é o unico q vamos usar

esta data augmentation consistitiu em:

Back-Translation : Translation-based augmentation using intermediate language pivoting to generate paraphrases.

Lexical Substitution : Random substitution of words with WordNet-based synonyms.

Template-Based Augmentation : Named Entity-aware augmentation by replacing entities (ORG, DATE, EVENT) using slot-filling over extracted templates.

alguns exemplos desta augmention são

1. In the building and home improvement trade , sales decreased by 22.5% to EUR 201.4 mn .

-¿ in the building and diy trade, sales decreased by 22. 5% to eur 201. 4 million.

2. In a media advisory , ...

-¿ in a media consultation

3. In January-June 2010 , diluted loss per share stood at EUR0 .3 versus EUR0 .1 in the first half of 2009 .

-¿ in the first half of 2009, diluted loss per share stood atomic number 85 eur0. 3 versus eur0. 1 in the first half of 2009.

TABLE VII: Hyperparameter space for .....

Hyperparameter	Possible Values
Epochs	$\{1, 2, 3, 4, 5\}$
Learning rate	$[10^{-5}, 10^{-2}]$
Weight decay	$[0, 0.5]$

best hyperameters:

- Num train epochs: 2

- Learning rate: 0.0001

- Weight decay: 0.1

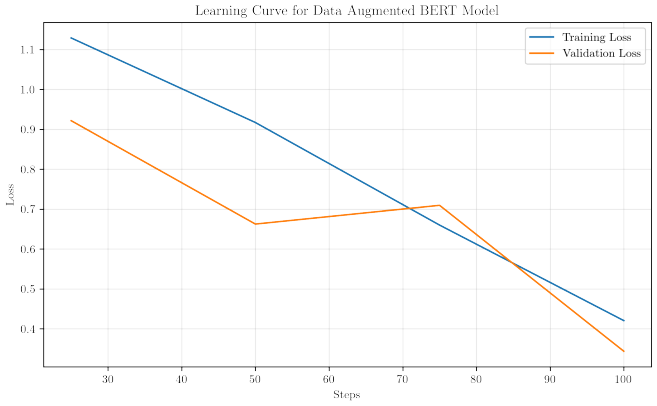

Fig. 8: learning curve

aqui 8 ela ainda nao convergiu, mas pq fizemos cv com o numero de epochs é como se fosse um earlystopping

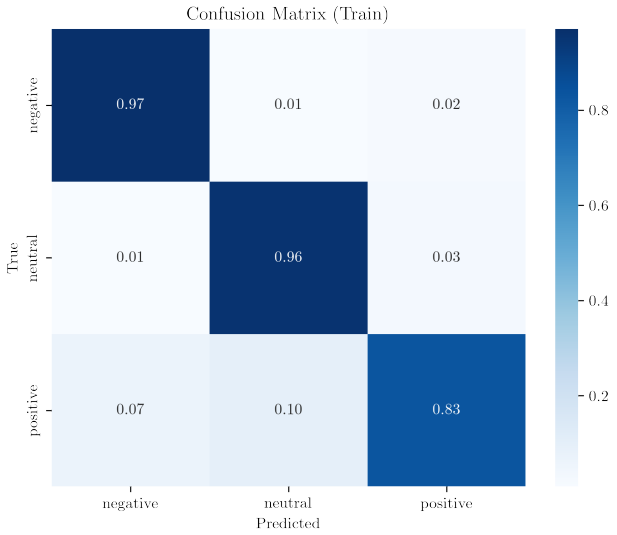

Fig. 9: data augm bert conf matri train

TABLE VIII: Classification report for data augm BERT on training data.

Class	Precision	Recall	F1-Score	Support
Negative	0.92	0.97	0.95	336
Neutral	0.90	0.96	0.93	336
Positive	0.94	0.83	0.88	336
Accuracy			0.92	1008
Macro avg	0.92	0.92	0.92	1008
Weighted avg	0.92	0.92	0.92	1008

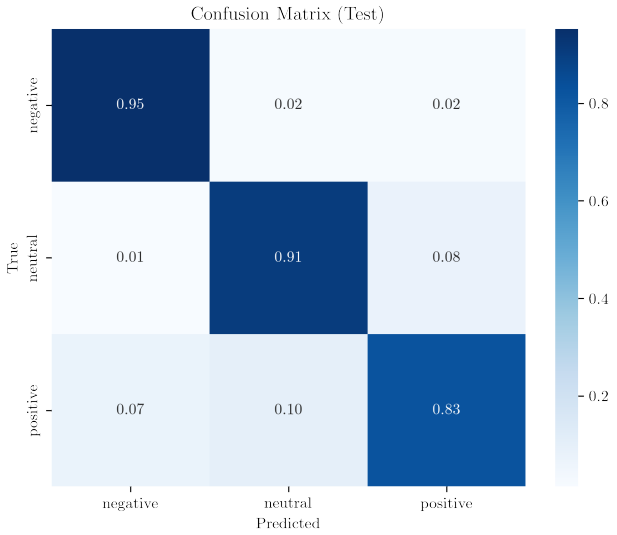

Fig. 10: data augm bert cong matrix test

TABLE IX: Classification report for data augm BERT on test data.

Class	Precision	Recall	F1-Score	Support
Negative	0.81	0.95	0.87	84
Neutral	0.95	0.91	0.93	429
Positive	0.80	0.83	0.81	178
Accuracy			0.89	691
Macro avg	0.85	0.89	0.87	691
Weighted avg	0.90	0.89	0.89	691

os resultados nao apresentam indicios de overfitting e vao ao encontro da literatura

tabela do research, 0.90 de accuracy do finbert q era finetunning do finbert neste dataset ou qq coisa assim

C. weighted model

a ideia desta abordagem é q havendo o equilibrio entre quantidade e qualidade dos dados, ao inves de ter de haver uma filtragem dos documentos usadas tendo por base o nível de concordância, por que não usar todos mas atribuindo pesos diferentes aos diferentes niveis de agreement

isto é, enquanto o modelo aprende, ser mais penalizado ou dar mais importância quando classifica mal uma instância com maior nível de agreement, e não tanta aos níveis mais baixos

após algumas experiencias empiricas, chegou-se a seguinte formula para o weight da classe i, com $i \in \{50\text{Agree}, 66\text{Agree}, 75\text{Agree}, \text{AllAgree}\}$.

$$w_i = \text{scale} \cdot p_i = \left(\frac{N}{\sum_{j=1}^{k} n_j \cdot p_j} \right) \cdot \left(\frac{e^{a_i}}{\sum_{j=1}^{k} e^{a_j}} \right),$$

where p_i is the softmax weight for class i, a_i the raw agreement scores for class i, n_i the number of samples in class i, and N the total number of instances.

The rationale behind this formulation starts from a set of empirically defined raw agreement scores, for each agreement class:

$$\{0.50, 0.66, 0.75, 1.0\}$$

These values represent the degree of annotator consensus and are transformed exponentially using the softmax function

$$p_i = \frac{e^{a_i}}{\sum_j e^{a_j}}$$

to capture differences in confidence in a non-linear manner.

Following this transformation, a normalization factor is applied to account for agreement imbalance in the training dataset

$$\text{scale} = \frac{N}{\sum_i n_i \cdot p_i}$$

ensuring that the average weight across all training instances equals 1, preserving the comparability of the loss function with a weightless scenario, such as the test dataset.

This approach leverages confidence information from agreement levels without distorting the overall loss scale during training.

esta formula é usada para atribuir os pesos aos diferentes niveis de agreemente que por sua vez, este peso é usado para multiplicar pela loss de cada instancia (tendo em conta a classe de agreement onde se encontra), fazendo com que a loss total ao longo do treino tenho mais um maior contributo dos maiores niveis de agreement face aos menores. ou seja usamos uma custom loss fucntion

pelo facto de ja ter em conta o dataset todo, o class balance resultatou num total de 520 obs por classe de sentimento ao invés das 336 habituais (qnd era so 75 agreement), aumentado o conjunto de treino num total de 552 obs

lalalala hyp

TABLE X: Hyperparameter space for .....

Hyperparameter	Possible Values
Epochs	$\{1, 2, 3, 4, 5\}$
Learning rate	$[10^{-5}, 10^{-2}]$
Weight decay	$[0, 0.5]$

best hyperameters:
- Num train epochs: 3
- Learning rate: 0.0001
- Weight decay: 0.1

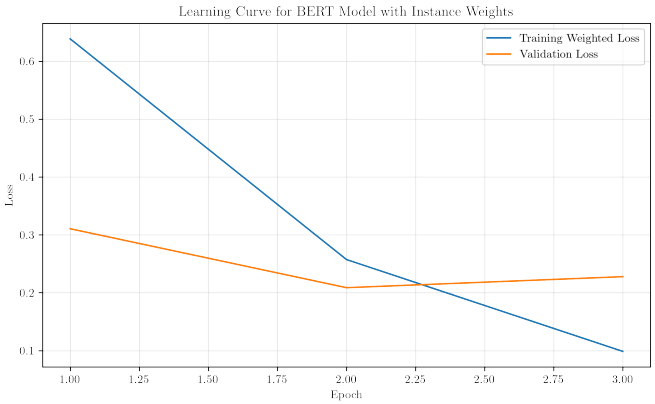

Fig. 11: learning curve

na fig 11 pode se ver a learning curve que representa a weighted loss fucntion dos dados de treino comparada com a loss fucntion dos dados de teste, sem qualquer peso, visto ser o nosso teste então é supsto dar a mesma importancia a tudo (ns explicar)

bla bla comprar metricas de teste e treino, mas por dificuldades de usar os pesos nesta analise foi ent selecionada só a porcao de dados de treino cujo agreement era supior a 75% usado nos dados de treino, visto q o foco do trabalho se cinge neste conjunto, tendo todas as instancias o mesmo peso nestas metricas fig. 12 e table XI. e até pq tp por exemplo na confusion matrix por ser quantidades de instancias seria injunto por exemplo instancias de baixo agreement penalizarem tanto como instancias de alto agreement.

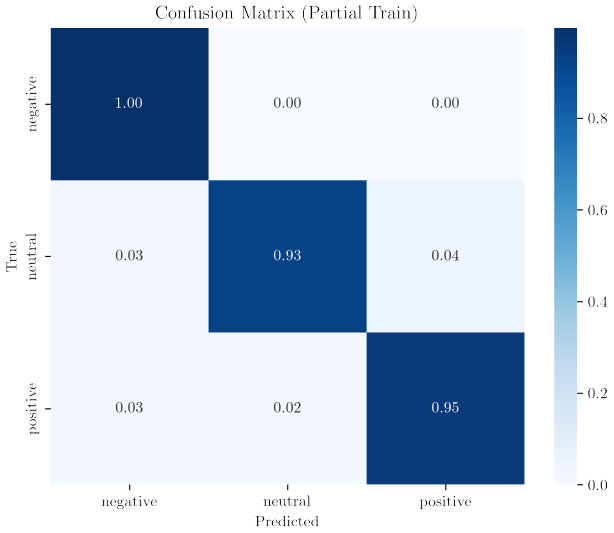

Fig. 12: weighted bert conf matri PARTIAL train

TABLE XI: Classification report for weight BERT on PARTIAL training data.

Class	Precision	Recall	F1-Score	Support
Negative	0.83	1.00	0.91	336
Neutral	0.99	0.93	0.96	1717
Positive	0.90	0.85	0.93	709
Accuracy			0.94	2762
Macro avg	0.91	0.96	0.93	2762
Weighted avg	0.95	0.94	0.94	2762

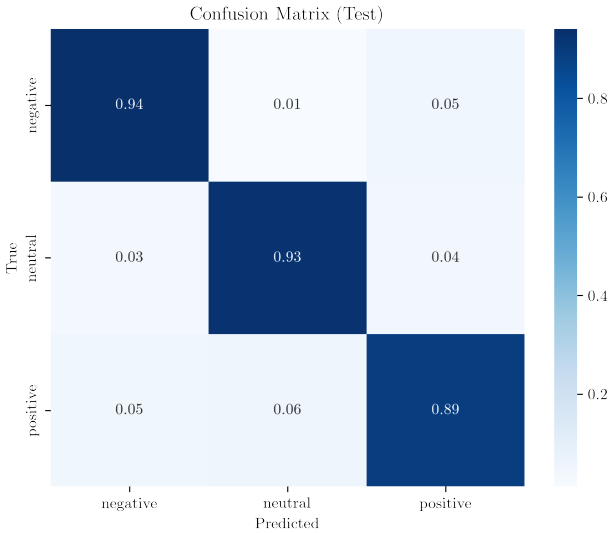

Fig. 13: weight bert cong matrix test

TABLE XII: Classification report for weight BERT on test data.

Class	Precision	Recall	F1-Score	Support
Negative	0.78	0.94	0.85	84
Neutral	0.97	0.93	0.95	429
Positive	0.88	0.89	0.89	178
Accuracy			0.92	691
Macro avg	0.88	0.92	0.90	691
Weighted avg	0.93	0.92	0.92	691

para melhorar estes resultados poderia ser levado a cabo um fine tuning ou a escolha por parte de um expert dos raw agreement scores ou uma melhoria da formula dos pesos

V. Discussion

results discuttion

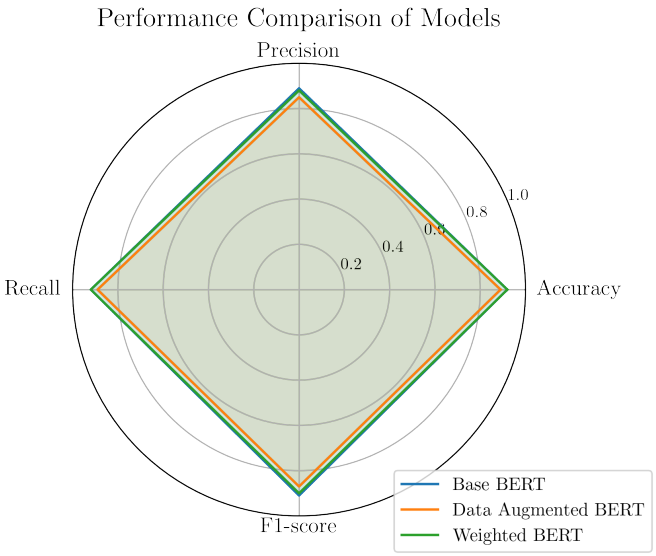

Fig. 14: radar chart results in test set (métricas são MACRO avg)

14 mostra que as tres abordagens criam modelos com metricas comparativas, apesar de neste caso o modelo Base BERT mostrar um desempenho superior a todos os outros em todas as metricas, mas tendo em conta o seu possivel overfifting, acaba por haver mais destaque para o weighted bert que o acaba por bater em algumas métricas tlvz sem o problema do overtffintg

tbm temos os resultados em tabela:

TABLE XIII: prefomece comparacion int test set vs literature aka soota

Model	Accuracy	F1 (macro)
base BERT	0.92	0.91
data augm BERT	0.89	0.87
weifhted BERT	0.92	0.90
literaturas1	xxx	ver
literaturas2	yyy	GPT
literaturas3	zzz	research??

VI. Conclusion

conclisao

Fig. 15: Enter Caption

Work Load

Both authors contributed equally to the project.

References

[1] P. Malo, A. Sinha, P. Takala, P. Korhonen, and J. Wallenius, "Financialphrasebank-v1.0," 07 2013.

[2] ——, "Good debt or bad debt: Detecting semantic orientations in economic texts," 2013. [Online]. Available: https://arxiv.org/abs/1307.5336