Sector-level sentiment analysis with deep learning

Ioannis Almalis[a], Eleftherios Kouloumpris[a,*], Ioannis Vlahavas[a]

[a]*School of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece*

Abstract

This paper presents new machine learning methods in the context of Natural Language Processing (NLP), in order to extract useful information from financial news. Traditional NLP approaches, based on the use of lexicons or standard machine learning algorithms, have ignored the importance of position and word combination in texts, resulting in reduced performance. More recently, NLP empowered by deep learning has achieved remarkable results in various tasks, such as sentiment analysis. This paper proposes a deep learning solution for sentiment analysis, trained exclusively on financial news, that combines multiple recurrent neural networks. Following this, our sentiment analysis models are further enhanced via a semi-supervised learning method that relies on the detection and correction of presumably mislabeled data. The performance of our proposed solution is compared favourably against both traditional and state-of-the-art models based on its performance of previously unseen tweets data. The paper also provides novel research towards the prediction of the specific economic sectors affected by news articles. Finally, we propose an ensemble of the sentiment and sector models in order to provide sector-level sentiment analysis with potential applications in the context of sector fund indices.

Keywords: Natural Language Processing, Machine Learning, Financial Sentiment Analysis

*Corresponding author

Email addresses: ialmalis@csd.auth.gr (Ioannis Almalis), elefthenk@csd.auth.gr (Eleftherios Kouloumpris), vlahavas@csd.auth.gr (Ioannis Vlahavas)

1. Introduction

Financial markets are volatile and at the same time susceptible to global events and phenomena, such as pandemics and political crises, trade competition, innovation and scientific discoveries. The spread of the COVID-19 pandemic has caused considerable disruption in the markets. The negative effects of this situation are reflected in global supply chains, as well as in declining global demand for goods and services. Uncertainty has increased in financial markets, with the US 10-year interest rate falling to a record low and at the same time business and consumer confidence has declined.

Financial investors trade on the basis of available information, especially those that consider information on corporate financial analysis, with the potential to influence the market. The effort to generate future revenue based on stock price behavior has affected numerous research areas. Initial research on the subject argued that stock movements do not follow specific patterns or trends, with the result that their past behavior is not a valid criterion for predicting their future value. Later studies have moved in a similar direction, indicating that emotions affect rational thinking and social behavior to such an extent that the stock market itself can be considered as a measure of social mood (Audrino et al., 2020).

Based on the above view, it could be considered that the analysis of public mood can be used to predict the movement of stock market prices. Bollen et al. (2011) report that changes in a particular public mood state are able to influence daily fluctuations in the closing prices of the Dow Jones Industrial Average. They also used graphs to study the correlation of micro-blogging activity in Twitter with changes in stock prices and the corresponding trading volumes.

As the stream of textual data such as news articles or tweets continues to expand rapidly, recent research suggests that the analysis of online texts on blogs, websites and social networks is useful for predicting a variety of financial trends. A large number of websites that publish and collect such financial news and articles are active on the internet.

Each one of these numerous sites manages a variety of financial articles, where in turn, each article contains tens or hundreds of words, which due to the inherent linguistic complexity, are difficult to process. Therefore, there is a need to collect, process and analyze such news using methods from modern computing fields, in order to handle countless pages of digitized texts and uncover the useful information that is hidden in plain sight. Consequently, the outputs produced by these automated methods enable investors to make informed decisions (Feldman, 2013; Day and Lee, 2016).

Natural Language Processing (NLP) is one of the aforementioned fields and includes a range of computational methods for analyzing and encoding natural texts at one or more levels of language analysis, in order to achieve human-level language processing for a range of tasks or applications (Farkash et al., 2015), such as emotion analysis, summary creation and keyword extraction. Sentiment analysis is a well-known and multifaceted task studied by NLP researchers (Jain et al., 2017). The goal of sentiment analysis is to locate and extract subjective information from text sources, by detecting the attitude, polarity (positive or negative) or opinion that is being communicated in the text.

Traditional approaches for developing sentiment detection models include dictionary based methods that rely on lexicons e.g. WordNet (Miller, 1995), as well as traditional machine learning algorithms. The latter category includes both supervised learning, with techniques such as Naive Bayes (Zhang, 2004), and unsupervised learning techniques like k-Means (Arthur and Vassilvitskii, 2007).

However, these methods often fail to accurately predict the polarity of financial texts, as they do not take into account the interrelationship of words, their position in a financial article and the semantics of the specific field of finance. Traditional NLP algorithms find it difficult to analyze "silent" concepts such as ambiguities, ironic expressions, idioms, metaphors, etc. In the context of finance, while these methods go as far as to extract sentiment from a particular text, the sentiment is not automatically associated with one or more corresponding stocks or stock sectors.

Deep learning later became the basis for the development of new sentiment analysis models in finance, but was mainly based on pretrained text embeddings, such as Google-News-Word2Vec and Stanford's GloVe. Pretrained word embeddings are dense vector representations of text that capture semantic relations. These embeddings are typically learned from a single language modelling task, later to be used as an effective data representation for solving other tasks. Large corpora of text, such as the Google News, fastText WIKI, TREC and IMDb datasets, have been used for learning text embeddings. Yet, those databases' general purpose content is the source of the weakness in approaching the complexity of economic articles, achieving hardly 70% accuracy in finding positive-negative sentiment.

In this paper, we propose a system consisting of three interconnected modules, where at the heart of each one, there is a deep learning architecture that uses reccurrent neural networks (RNNs). The first module performs a preliminary label correction task, by detecting and relabeling news that has most likely been mislabeled as neutral. The second module is used for the classification of news as positive or negative, at the same time leveraging the most likely mislabeled examples for semi-supervised learning. The third module is used for the prediction of the particular economic sectors affected by news, based on text content. Finally, the proposed interconnected system combines the above modules in order to perform sector-level sentiment analysis. The contributions of the paper are further explained in the following paragraphs.

Firstly, we propose a sentiment analysis model that, according to our experiments, is competitive against a state-of-the-art method for financial texts, namely FinBERT. While our model is slightly outperformed by DistilBERT in terms of accuracy, it is several times faster in terms of training and prediction time. As it will be further explained in a later section, we consider the shorter prediction time as relatively more beneficial for algorithmic trading applications, especially when the accuracy is comparable. Similarly, Scholtus et al. (2014) concluded that a time delay in trading activity can significantly reduce returns of news-based trading strategies, with a delay of 300ms (1s) incurring about 10.85% (20.05%) losses per year.

Secondly, we introduce a semi-supervised learning approach that takes advantage of neutrally labeled data. Specifically, we detect neutral data that are most likely mislabeled, which are then relabeled as positive or negative in order to augment the dataset used for binary sentiment analysis. Two deep learning ensembles are used to achieve this augmentation step. The first is an LSTM/GRU ensemble trained with neutral/not-neutral labeled data that can be used to detect neutrals that are most likely mislabeled. The second is another LSTM/GRU ensemble, trained with positive/negative labels, that relabels the previously detected data as positive or negative. Our experiments show that this label correction method marginally improves the results of our sentiment analysis model, after the latter is retrained on the augmented dataset.

Thirdly, we investigate the problem of sector prediction based on news data. To the best of our knowledge, we propose the first NLP application based on deep learning that can extract the particular sectors affected by news. The experimental results show that this model significantly outperformed DistilBERT and can effectively predict the sector that is most relevant to each news item.

Lastly, we propose a novel system named Sector-Level Sentiment Analysis (SLSA) that combines the previous models in order to extract the overall sentiment that affects each sector (i.e. the sector-level sentiment). As far as we are aware, the combination of sector prediction and sentiment analysis models in order to derive sector-level sentiment analysis has not been previously proposed in the literature. To evaluate our system, we also propose three performance measures for this task: Sector Sentiment Accuracy (SSA), Sector Sentiment Percentage Error (SSPE) and Mean Sector Sentiment Percentage Error (MSSPE). Our experiments show that this combined scheme is an effective method for determining sector-level sentiment.

The remainder of the paper is structured as follows: Section 2 covers related work from literature. Section 3 presents the learning methods and data used. Section 4 presents the proposed machine learning modules and interconnected system. Section 5 provides the experimental results. Section 6 includes a discussion on the findings. Section 7 concludes the paper.

2. Related Work

Sentiment analysis of financial texts is an NLP task, the goal of which is to interpret and classify emotions in financial data expressing a positive or negative sentiment. A survey from Klein and Prestbo (Klein and Prestbo, 1974), using an ontology-guided semantic technique, established a theory regarding how a pessimistic financial report can affect the markets with results that strongly support this kind of correlation between markets and financial news.

Ederington and Lee (1993) suggested that market volatility could be based on financial texts and especially on press releases. Wüthrich et al. (1998) developed a computational linguistics system, based on financial articles from five popular financial websites, to improve stock market predictive models and Melvin and Yin (2000) stated the importance of the financial news headlines for investors.

Chan (2003) noticed that stocks mentioned in news releases performed significantly better than others at the same time period. Loughran and McDonald (2013) designed a common-term-weighting scheme to analyze the sentiment of financial texts and suggested that low level prices in many initial public offerings (IPOs) could be explained by the existence of news presenting uncertainty or negative sentiment. Baker and Wurgler (2006) and Kothari et al. (2009) investigated the correlation of stock price volatility with the sentiment inherent in financial news.

Ahmad (2011) introduced the notions of return and volatility in the context of sentiments extracted from news and also examined whether these can improve the estimation of financial risk. Généreux et al. (2011), assuming that the sentiment in news carry information about the future direction of prices, explored the short-term impact of financial news items on the stock price of companies. They proposed an effective Support Vector Machine (SVM) model that was trained on news labelled according to market reactions.

While we can't deny that investors base their decision on hard facts, such as company earnings and price signals, it is also true that they are, at some extent, influenced by the prevailing sentiments that surround them. Cambria et al. (2017) highlights that automatic and effective processes for capturing public sentiments have significant implications for financial market predictions. Given that multiple channels of information (text, audio, images etc.) influence market sentiment, a multi-modal approach is promising for gaining an edge on competition.

Later works established the fact that sentiment analysis is relevant to many challenges of finance. Some important problems include volatility forecasting, trend forecasting and portfolio management. Based on the assumption of a bidirectional interaction between asset prices and market sentiment, Xing et al. (2019) proposed the Sentiment Aware Volatility Forecasting (SAVING) model that incorporates market sentiment signals in a neural network architecture. Regarding the prediction of price fluctuation, it was shown that SAVING outperforms statistical methods and RNNs that rely solely on historical price data.

Xing et al. (2018) proposed a novel portfolio management method by combining market sentiment views with modern portfolio theory via a Bayesian approach. A hybrid approach that combined evolved clustering with a Long Short-Term Memory (LSTM) model (Huang et al., 2015), an improved RNN architecture, was used to extract market sentiment views. The proposed portfolio management method demonstrated higher profits compared to several benchmarks. Malandri et al. (2018) applied machine learning to directly learn the best asset allocation strategy based on historical prices and public mood features. Experimenting with five portfolios, they highlighted the addition of sentiment features for notably increasing revenues, while the same features were best utilized by LSTM.

Picasso et al. (2019) followed another machine learning approach to forecast the trend of a portfolio consisting of the twenty most capitalized companies listed in NASDAQ100. Specifically, they proposed a time series classification system based on market data, fundamental data and sentiment features that were extracted from news articles. The effectiveness of this forecasting approach was further demonstrated in a simulated High Frequency Trading (HFT) scenario.

As far as machine learning approaches are concerned, Pang et al. (2002) introduced empirical methods in NLP using sentiment classification based on movie reviews. They used traditional machine learning methods, like Naive Bayes and SVM, and the results showed that the latter method was the best performing in any experiment. While the utility of extracting sentiment from online texts was proven, the previously mentioned methods could not capture the object of each sentiment in detail.

Schumaker and Chen (2009a,b), applied sentiment analysis on news, using bag-of-words, noun phrases and named entities as text representation, in order to integrate stock prediction models with textual content. Furthermore, Linear Regression and SVM were the machine learning classifiers used as predictive models. Zhang and Swanson (2010) analyzed online financial texts and estimated that the sentimental content of these texts presented an additional value.

Identification of phrase subjectivity was a key factor that Wiebe and Mihalcea (2006) pointed out through machine learning methods based on word polarity. Their effort indicated that a negative sentiment does not mean the pessimistic mood of the article's author. Chua et al. (2009) deployed a sentiment extraction engine from internet stock message boards, which consisted of a variation of Naive Bayes classifiers and produced an accuracy of 78.72%.

Thelwall et al. (2010) implemented SentiStrength, an algorithm to evaluate sentiment levels from stock news written in informal English. This approach failed to achieve accurate sentiment analysis for specific domains of financial activity. Knowledge that derives from specific financial domains could affect the sentiment polarity in financial articles. Thus, if common financial terms are treated as simple words, it is difficult to extract the real sentiment in domain-specific expressions.

Wang et al. (2014) proposed a method based on ensemble machine learning techniques, supporting the idea that sentiment analysis should focus more on phrases than on individual words, since the phrases correspond to the most meaningful parts of a text. Cohen et al. (2011) implemented a series of pre-processing steps in order to filter out unrelated information that exist in tweets due to their informal structure. These steps improved the quality of the extracted tokens enough to improve the performance of the classifier. Similarly, Srividhya and Anitha (2010) investigated the contribution of stemming and stop word removal to text analysis.

However, Saif et al. (2012) mentioned that the removal of stop words could probably reduce the accuracy of sentiment analysis, since these words may have a specific role for sentiment classification. Rechenthin et al. (2013) used a variety of classification models to predict stock trends, based on Yahoo Finance Message Board. A keyword based algorithm was proposed to classify tweets as positive, neutral or negative. The proposed model achieved almost 75% accuracy.

Neutrality is frequently ignored in sentiment analysis due to its vagueness and lack of information. In Valdivia et al. (2018), however, it is considered to be the main key for distinguishing between positive from negative classes and improving sentiment classification. Neutrality is considered as potential noise, so from a noise filtering point of view, the detection and removal of noise can improve performance. To this end, a neutrality proximity function is introduced that assigns weights to polarities according to its proximity to a neutral point.

Wang et al. (2020) proposed a new sentiment analysis scheme, namely multi-level fine-scaled sentiment sensing with ambivalence handling. The ambivalence handler is described, indicating strength-level tuning settings for analyzing the strength and fine-scale of both positive and negative attitudes. When both positive and negative co-exist, ambivalence is configured as a combination of mixed-negative (stronger weighting of negative sentiments), mixed-positive (stronger weighting of positive sentiments) and mixed-neutral (equal weighting of positive and negative sentiment).

LSTM and Gated Recurrent Unit (GRU) (Dey and Salem, 2017) have been a cornerstone for NLP research due to their ability to overcome vanishing or exploding gradients in longer texts. Basiri et al. (2021) explain several issues that appeared in previous sentiment analysis systems that used RNNs like LSTM or GRU, more importantly, their high dimensional output when used as feature layer and the fact that they consider all words as of equal importance.

To address these issues, they proposed an Attention-based Bidirectional CNN-RNN Deep Model (ABCDM) for sentiment analysis on both long product reviews, as well as shorter tweets. The Convolutional Neural Network (CNN) layer reduced the dimensionality of the output, while an attention mechanism was employed to learn which parts of the input were relevant. ABCDM outperformed several state-of-the-art models at short tweet polarity classification, which is interesting in view of the fact that Twitter significantly contributes to the formation of market sentiment.

While there is still ongoing research towards improving RNNs, Vaswani et al. (2017) followed another approach and introduced the Transformer, a neural architecture that is based on the attention mechanism without the need for recurrent layers. Subsequent papers proposed several pre-trained Transformer based models for language tasks that received significant popularity, such as BERT (Devlin et al., 2019) and GPT-3 (Floridi and Chiriatti, 2020).

The Neural Tensor Network (NTW) has also been of particular interest due to its ability to learn multiple relationships between entities (e.g. words), which can then be used as features for sentiment analysis or other NLP tasks. Li et al. (2021) provided a mathematical analysis of NTW based on Taylor's theorem to shed light on the connection between NTW and traditional neural networks.

3. Methods and Materials

This section presents the data and machine learning algorithms employed in our experiments. Firstly, the sources of text data are given, and all data preprocessing steps are revealed. Secondly, we provide references for the machine learning algorithms that were used in the experiments.

3.1. Dataset

In order to construct our datasets, we use a financial news network website, called StockNewsApi.com which offers videos and articles from more than thirty (30) news sources.[1]

[1] The news' sources include The Street, CNBC, Zacks, Benzinga, Bloomberg, Engadget, Forbes, MarketWatch, The Motley Fool, Investor Business Daily, Seeking Alpha, 24/7 Wall Street, Business Insider, Business Wire, CNET, CNN, Forbes, Fox News, GeekWire, Huffington Post, NYTimes, Reuters and The Guardian. The news text data used for this research are provided in the following link: `https://drive.google.com/file/d/1bP6D_k7bfkaQLCB5D27Buz_Ag-X6ar7o/view?usp=sharing`

Table 1: Datasets details

TechNews		AllTickers	
Number of Examples	43189	Number of Examples	133743
Number of Examples, no duplicates	25547	Number of Examples, no duplicates	74595
Number of words before cleaning	855938	Number of words before cleaning	2635769
Number of words after cleaning	557857	Number of words after cleaning	1717037
Negative Examples	11372	Negative Examples	37278
Positive Examples	14175	Positive Examples	37317

In this paper, we constructed and used two different datasets. We named them the TechNews dataset, which consists of financial news articles exclusively related to technology companies, and the AllTickers dataset, which includes general market news from a variety of economic sectors. Details for both datasets are shown in Table 1.

All news items include a sentiment tag that can be described as positive, negative or neutral. The last description is used if the title or the main content of an article cannot be clearly defined as positive or negative. The data also includes the following fields for each news item: The news title, the main informative content (text), the source, the publication date, the derived sentiment (positive-negative-neutral) and the stock tickers referred to in the news.

Furthermore, the collected news items have additional labels with respect to their related economic sectors. The sector areas along with the respective number of examples for each one are presented in Table 2.

Table 2: Economic Sectors

Sectors	Number of Examples
Technology	26934
Healthcare	26934
Financial	19270
Consumer goods	20279
Energy	3965
Commodity	5743

3.2. Preprocessing

In the first phase, we remove the duplicate news records, as data overlaps may happen during the download process in StockNewsApi.com. The main reason for the overlap is the fact that the main API call filter is the publication date and the time difference with the USA does not allow us to cover all the day's articles. Inevitably, if we want to get the missing articles of the previous day, we will also receive already acquired ones. The remaining data is then subjected to a series of processes.

- Converting all words to the lower case so that there is no distinction of words depending on how they are written (uppercase or lowercase).

- Removal of special characters. Special characters are non-alphanumeric symbols that are most often found in comments, references, currency symbols, etc. Such characters do not provide any additional value to the text's semantic content and provoke noise in machine learning algorithms.

- Removal of Numbers and Dates: As we deal with texts, numbers as well as dates may not add any significant informational value to linguistic analysis. Regarding the analysis of financial texts, Sun et al. (2014) and Pejic Bach et al. (2019) mention that numbers add noise and should be removed during data preprocessing. In this work, we conducted experiments that confirmed the above fact, so we decided not to include numbers.

- Removal of stop words: Stop words are often used in order to make sentences grammatically correct, for example, words like a, is, an, the, etc. These words are of minimal to no importance in text sentiment analysis and are available in abundance in open texts, articles, comments, etc. It is considered semantically correct to remove these words as well as those with a character length of two or less, so that machine learning algorithms can focus better on words that define the informative content of the article.

- Tokenization: It is a way of splitting a text into smaller sections called tokens. Here, tokens can be either words, characters or parts of words, therefore, tokenization can be broadly classified into 3 types - word, character, and subword (n-gram characters). The result is a list of tokens that have been separated without punctuation.

- Stemming: Is defined as the process of converting words into the form of their stem, base or root. The stem does not have to be identical to the original word. There are many ways to implement stemming, such as algorithms based on search tables and suffix removal. These are mainly based on removing letters, such as 's', 'es', 'ed', 'ing', 'ly', from the end of words.

An exception is made for the input of Transformer based models (Section 3.3), namely DistilBERT and FinBERT, for which only the recommended tokenizer-encoder implementations are applied.

For each dataset, we proceed by splitting the data into training set at a rate of 70%, and testing set at the remaining rate of 30%. Especially, for the purpose of model selection, we use 20% of the training data as a validation set. We consider two methods in order to handle class imbalance. In the first method, the training data is subject to oversampling of the minority class examples, while in the second method, we apply cost-sensitive learning. The latter method is achieved by using weights either as a parameter in machine learning classifier or in neural network's loss functions.

When data separation is completed, we perform data vectorization [2] for Naive Bayes, Random Forest (RF) and SVM estimators, using Term Frequency - Inverse Document Frequency (TF-IDF). TF-IDF is a particular method for vectorizing text based on the following information: (a) the frequency of each term (i.e word or token) in each text and (b) the number of texts in which each term appears.

3.3. Learning methods

We consider two families of learning methods, namely traditional machine learning and the more recent deep learning approaches for sequential data. The traditional machine learning algorithms that we use in each task are Naive Bayes (Zhang, 2004), RF (Xu et al., 2012), SVM (Vishwanathan and Narasimha Murty, 2002) and Extremely Randomized Trees (ERT) (Geurts et al., 2006). Our deep learning approaches include LSTM and GRU, which are compared to two state-of-the-art models, namely DistilBERT (Sanh et al., 2019) and FinBERT (Huang et al., 2020). DistilBERT uses distillation to produce a lighter and faster version of the Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019), which belongs to the

[2] In this context, data vectorization is the essential process of representing textual data as numerical vectors. The transformation is a necessary step as traditional machine learning algorithms work on numerical data.

Table 3: Hyperparameter search space and best configuration for LSTM and GRU

Hyperparameter	**Search space**	**Comment**
Learning rate	[0.0005, **0.001**, 0.0015, 0.01]	Best performing for all experiments
Epochs w/ early stopping	[1, 2, .., **15**, **16**, **17**, .., 30]	For all experiments fell within 15-17
Dropout	[0.3,0.4,**0.5,**0.6,0.7]	Best performing for all experiments
Number of recurrent layers	[**2**,3]	Best performing for all experiments
Units per recurrent layer	[**256**,300,512]	Best performing for all experiments

Table 4: Hyperparameter search and best configuration for RF and SVM

Learning Algorithm	**Hyperparameter**	**Search space**
Random Forest	Max estimator depth	[10, 20, 30, 40, 50, 60, 70, 80, **90**, 100, 110, None]
	Min samples per leaf	[1,**2**,4]
	Min samples per split	[2,5,**6**,10]
	Number of estimators	[200, 288, 377, 466,555, 644, **600**,733, 822, 911, 1000]
	Bootstrap samples	[True,**False**]
	Class weights	[None, **Balanced**]
SVM	C	[0.001,0.01,0.1,**1**, 10, 100, 1000]
	Kernel	[**linear**]
	Decision function shape (multiclass)	[**One versus One**, One Versus Rest]
	Class weights	[**Balanced**, None]

category of neural networks known as Transformers (Vaswani et al., 2017). For the sentiment analysis task on tweets, we also compare our results with FinBERT, a state-of-the art Transformer particularly fine-tuned for sentiment analysis of financial texts.

The work utilized two popular machine learning frameworks. The traditional learning models were implemented with Scikit-Learn (Pedregosa et al., 2012), while deep learning algorithms were implemented with PyTorch (Paszke et al., 2019). For DistilBERT and FinBERT we also used the HuggingFace Transformers package (Wolf et al., 2020).

3.4. Hyperparameter search

In terms of hyperparameter tuning[3] we applied grid search to choose hyperparameters. This section presents the hyperparameters that were considered, the search space used for each hyperparameter, and the best configuration for each model.

Concerning our neural network models, LSTM and GRU, we experimented with learning rate, batch size, dropout rate (Srivastava et al., 2014),

[3]Machine learning models require the specification of both parameters and hyperparameters. While the former are estimated by the training algorithm itself, the latter have to be set by the experimenter. Thus, in this paper we employ grid search for hyperparameter tuning.

number of recurrent layers and number of units per recurrent layer. The best hyperparameters were found to be the same for LSTM and GRU, while there was a small variation in the number of epochs for sentiment analysis and sector prediction. Moreover, early stopping was used to select the number of epochs. The search space for each hyperparameter is provided in Table 3, with the corresponding best configuration for each hyperparameter appearing in bold text. In summary, we opted for architectures with 2 recurrent layers of 256 units each, trained for 15-17 epochs with the use of a 0.001 learning rate and 0.5 dropout rate.

The aforementioned models are bidirectional, i.e. each sentence is examined from two hidden levels, where the first parses the sentence from the first word to the last one, and the second parses the semantic content of the sentence in reverse order, from the last word to the first. The final decision for the dependent target variable is produced by combining the decisions of the two hidden levels.

With regard to the traditional methods, for RF we try to obtain the best combination of the following hyperparameters: number of trees, maximum tree depth, maximum number of features consider for best split, minimum number of samples for node splitting, minimum number of samples for node to be considered a leaf and whether to use bootstrap samples. The grid search results for RF are provided in Table 4. The best configuration used 600 estimators of depth less than 90, with a minimum of 2 samples per leaf and 6 per split.

Regarding the SVM estimator, we perform a grid search process to tune the regularization parameter C, using a linear kernel. We note that polynomial and RBF kernels were not investigated further due to our limited computational resources.

For ERT, we opted for the default setting since tuning did not bring any significant differences in performance. For Naive Bayes, following an empirical rule based on the number of classes, we set the smoothing operator to 0.2 for sentiment analysis and 0.6 for sector prediction.

4. Machine Learning System Architecture

In this section we present the three machine learning modules that were used for the tasks of label correction, sentiment analysis and sector detection. We also present how these modules were integrated into a sector-level sentiment analysis system.

4.1. Detection and relabeling of presumably mislabeled neutrals

In preparation for this task, the available financial articles are labeled as neutral or not-neutral, with the latter label given to news that were originally labeled positive or negative in our dataset.

Initially, we assume that a number of samples have been mislabeled as neutral. Therefore, our first goal is to detect samples that have most likely been mislabeled as neutral. To this end, we train a neutral/not-neutral ensemble classifier (LSTM and GRU) and consider as presumably mislabeled the samples that were originally labeled as neutral but were predicted as not-neutral by both LSTM and GRU after using a 0.5 probability threshold. We consider it possible that the remaining neutral samples, those that were not detected as potentially mislabeled, are actually neutral in terms of polarity. Thus, due to the fact that we perform binary sentiment analysis, these truly neutral samples were removed and will not be relevant for the remainder of this work.

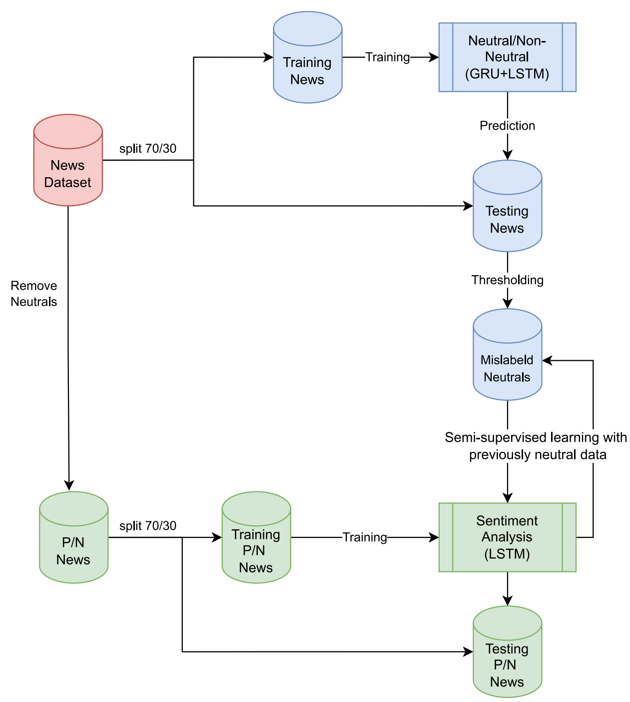

Figure 1: Flow diagram for the label correction task. P/N news refers to the part of the data that either positive or negative news.

Following this, our next goal is to relabel the detected samples as positive or negative. For this purpose, we use an ensemble positive/negative classifier (LSTM and GRU) to relabel the data that will be used for semi-supervised learning (see Section 4.2). The rules for relabeling are the following: (a) If both LSTM and GRU predicted probabilities higher than 0.9, the sample is relabeled as positive (b) If both LSTM and GRU predicted probabilities lower than 0.1, the sample is relabeled as negative. Otherwise, the sample is considered irrelevant for the remainder of this work and is removed from the dataset. Figure 1 presents a graphical illustration for the module that detects and relabels potentially mislabelled data.

4.2. Sentiment analysis for polarity detection

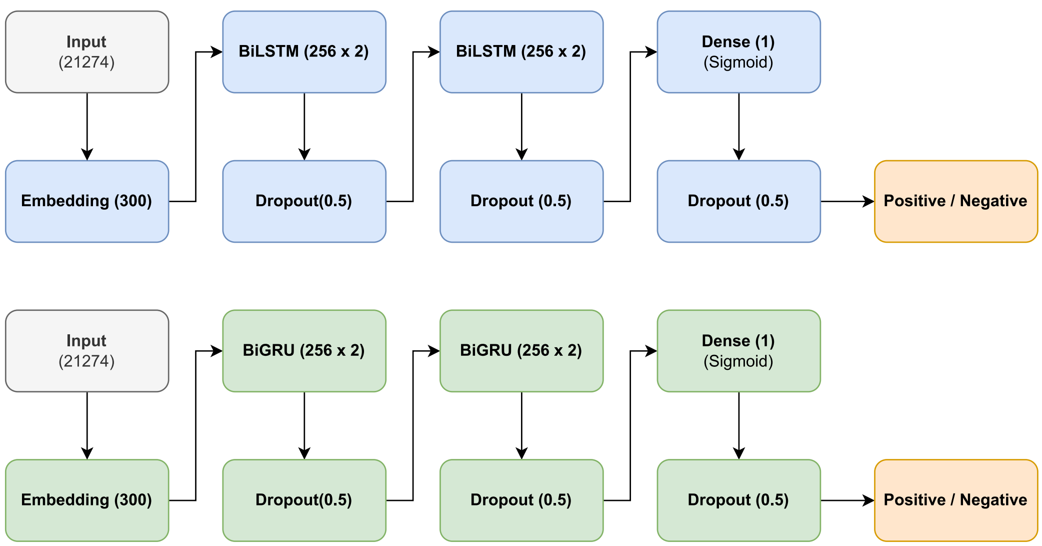

Figure 2: Deep learning architectures for the LSTM and GRU models of the sentiment analysis task.

In this task, our goal is to classify financial articles as positive or negative, depending on the sentiment that is derived from their textual content. To this end, we train both LSTM and GRU models for binary classification on data labeled as positive or negative.

The deep learning architectures for both models are provided in Figure 2, while the remainder of this paragraph will provide more detail. In the input layer, the vocabulary initially has 21274 words, which are then embedded as

300 dimensional dense vectors. Both architectures use two stacked bidirectional recurrent layers (BiLSTM or BiGRU) of 256 units, followed by a fully connected layer with 1 unit. Dropout at 0.5 is used for the recurrent and fully connected layers. The last layer uses a logistic activation function to output scores between 0 and 1, which are interpreted as sentiment polarity scores.

We note that the same architectures are used for the detection and relabeling tasks mentioned in Section 4.1. A few differences are that (a) in these tasks LSTM and GRU are used as an ensemble and (b) for the detection task only the label is changed to neutral/not-neutral. As mentioned in the same section, additional samples, originally labelled neutral, were relabeled as positive or negative. The same were added in the positive/negative training set in order to retrain the sentiment analysis model with more examples.

For the sentiment analysis task, we compare and evaluate the methods given in Section 3.3. In addition, we evaluate each method on a completely unknown set of Tweets data, which includes economic news and is used exclusively as a second testing dataset.

4.3. Sector prediction

The objective of sector prediction is to detect the economic sector that is related to each news item. The sectors used as labels in this task are technology, healthcare, financial, consumer goods, energy and commodities. Furthermore, we evaluate and compare the same methods mentioned in the previous section, excluding FinBERT.

A graphical illustration for the sector prediction deep learning architectures are provided in Figure 4. While the architectures are similar to the ones mentioned in 4.2, the output layer is different. Specifically, we use a fully connected layer with 6 units (i.e. one for each sector) and softmax activation in order to output sector membership scores that add to 1.

4.4. Sector-level sentiment analysis

As it was revealed in the Introduction, we combine the previously mentioned modules (see Section 4.1, Section 4.2 and Section 4.3) in order to develop an interconnected solution. Initially, this solution detects the sector of economic activity that is directly affected by each news article. The following step consists of the classification of the same news items as positive or negative. The final output is an estimation of the sector-level sentiment, with the computational steps described in the following paragraph.

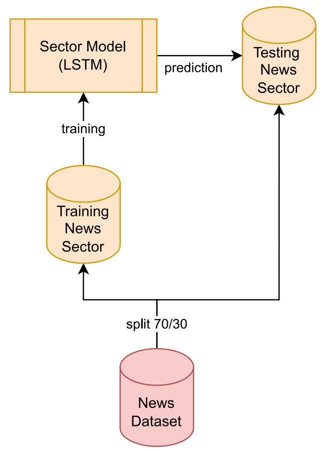

Figure 3: Flow diagram for the sector prediction task.

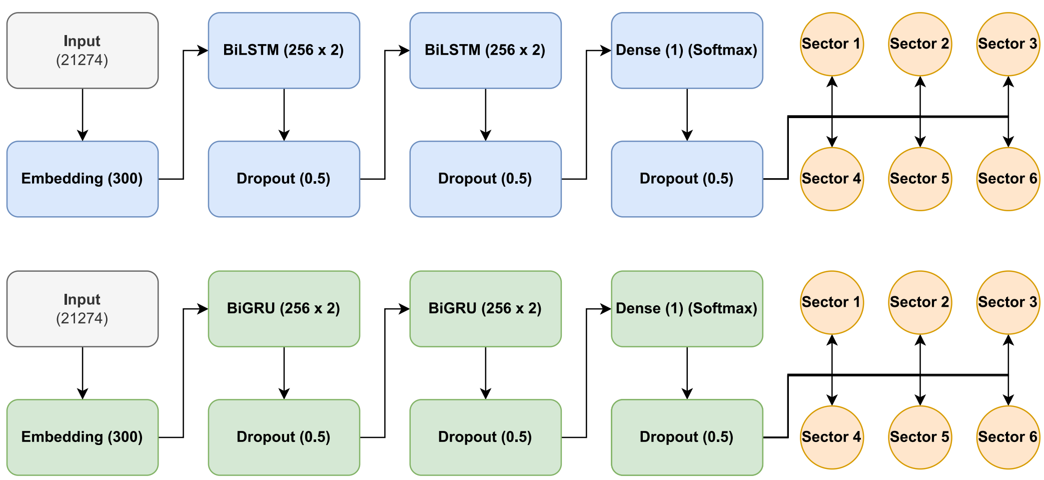

Figure 4: Deep learning architecture for the sector detection task.

Firstly, all news items are grouped according to the predicted sector. Secondly, for each sector's group we aggregate the predicted sentiment for the news that belong to this group, specifically by averaging[4], to derive the sen-

[4]An interesting alternative would be to compute the weighted average with regard to the news source or author popularity.

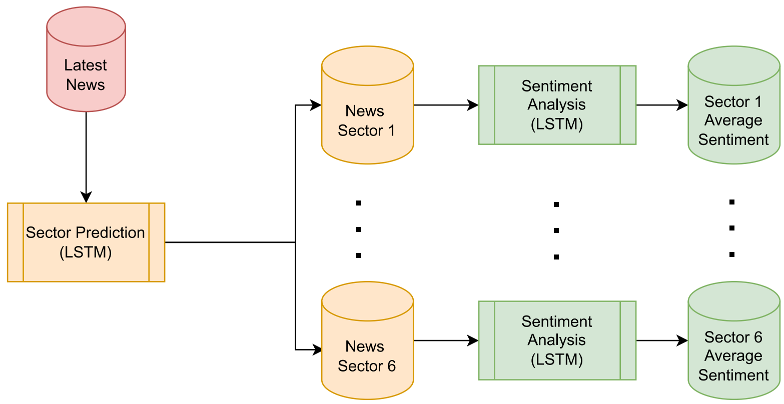

Figure 5: Flow diagram of the interconnected sector-level sentiment analysis system.

timent of the corresponding sector. The architecture for this interconnected system, with respect to its application on previously unseen news data (latest news), is provided in Figure 5.

While the previous approach combines the models of Section 4.2 and Section 4.3, we also consider an alternative approach to sector-level sentiment forecasting with a single multi-class model. The latter method directly classifies news into one among twelve classes formed by the Cartesian product of the different sectors and sentiment levels. The twelve classes are {*Consumer-Negative, Consumer-Positive, Financial-Negative, Financial-Positive, Technology-Negative, Technology-Positive, Health-Negative, Health-Positive, Energy-Negative, Energy-Positive, Commodity-Negative, Commodity-Positive*}, such that a single model simultaneously classifies both sentiment and sector. After decomposing these predictions into a sentiment and sector (e.g. *Energy-Positive* is decomposed into the labels *Energy* and *Positive*), the estimation of sector-level sentiment is identical to the previous approach, that is the computation of the average predicted sentiment for each predicted sector.

To elaborate, the first approach is a two-step method that classifies sector and sentiment with independent models, and combines the results thereafter. In contrast, the second multi-class approach simultaneously classifies both sentiment and sector in a single pass. A hypothesis in favour of the second approach is that the concept of sentiment may differ along the various sectors. For instance, what might be considered as positive sentiment in the context

Table 5: Sentiment analysis test set statistics

Task	Dataset	Positive	Negative
Sentiment Analysis (Binary)	General News	11135	11164
	Tech News	4235	3399
	Tweets	604	1363

Table 6: Sector prediction test set statistics

Task	Technology	Healthcare	Financial	Consumer Goods	Energy	Commodity
Sector Prediction (Multiclass)	8080	5404	5781	6084	1190	1723

of the fast-growing and unpredictable technology sector might be considered negative in a more established and stable sector. A benefit in favour of the first approach is that there are more examples per class, as there are two and six classes in the constituent sentiment and sector models, respectively.

Essentially, both methods begin with a different approach to predict a sector and sentiment label for each news text. The remaining pipeline, that is the estimation of the sector-level sentiment by averaging the sentiment of news over each sector, is identical for both methods.

4.5. Evaluation process

As mentioned in Section 3.2, for each dataset we split the available financial articles into a training set at a rate of 70%, and a testing one at a rate of 30%. Especially, in order to find the optimal hyperparameters of neural network models, we use a 20% of the training data as a validation set.

As far as sentiment analysis is concerned, we use an additional testing set that exclusively contains financial tweets, which also mentioned in Section 4.2. When the training process is completed, we estimate each model's performance on a set of unknown instances (testing set) with respect to the following metrics suitable for classification problems: Accuracy, Balanced Accuracy, Precision, Recall and F1. The label distribution for the sentiment analysis test set is provided in Table 5, while the corresponding information for the sector prediction test set is provided in Table 6.

For sector-level sentiment analysis, we evaluated the performance of both the two-step and multi-class approaches as follows. Firstly, we estimate the accuracy in computing simultaneously both sector and sentiment labels over N news articles, and denote this quantity as Sector Sentiment Accuracy (SSA) and estimate it with the formula given in Equation 1.[5] For the k-th

[5]The indicator function $\mathbb{1}(e)$ is equal to 1 if the logical expression e is true, or 0 if it

news article, $Sect_k$, $\hat{Sect_k}$, $Senti_k$, $\hat{Senti_k}$ denote the actual sector, the predicted sector, the actual sentiment and the predicted sentiment, respectively. The set of all sectors is denoted by S, while s is a particular sector.

Secondly, we define Sector Sentiment Percentage Error (SSPE) as the percentage error between the actual and predicted sentiment scores for each sector $s \in S$, as in Equation 2.[6] While the former quantity is estimated with the actual sector and sentiment labels, the latter uses predicted sector and sentiment labels. Finally, we computed the Mean Sector Sentiment Percentage Error (MSSPE) by averaging the SSPE over all sectors, as in Equation 3.

$$SSA = \frac{1}{N}\sum_{k=1}^{N}(\mathbb{1}(\hat{Sect_k} = Sect_k \wedge \hat{Senti_k} = Senti_k)) \tag{1}$$

$$SSPE(s) = 100 \times \frac{\mathbb{E}_{Sect_k=s}\left[Senti_k\right] - \mathbb{E}_{\hat{Sect_k}=s}\left[\hat{Senti_k}\right]}{\mathbb{E}_{Sect_k=s}\left[Senti_k\right]}, s \in S \tag{2}$$

$$MSSPE = \mathbb{E}_{s\in S}[SPE(s)] \tag{3}$$

5. Experimental Results

We conducted several experiments on the technological and general news datasets. In this section, we present the performance results for the sentiment analysis module, the sector detection module and the interconnected sector-level sentiment analysis system.

5.1. News-level sentiment analysis results

The sentiment classification process consists of two main steps. In the first step, we use two different datasets, TechNews and AllTickers.

We proceed to the phase of training and evaluation of our models, based on the above datasets, presenting the corresponding results in Table 7 and Table 8 respectively. Furthermore, in the second step, we proceed to an independent evaluation of our pretrained models with an unknown dataset

false. The logical AND operation $p \wedge q$ is true only if both operands p and q are true.

[6]The expression $\mathbb{E}_{e(x)}[f(x)]$ denotes the expected average of $f(x)$ over the set $\{x|e(x)\}$.

that includes tweets of financial content. Additionally, we also evaluate FinBERT with pretrained weights and after being fine-tuned with 10,000 financial statements for a sentiment prediction task. Results for the tweets test set are shown in Table 9.

As shown in the evaluation tables, DistilBERT is the best performing model across most metrics for both TechNews and AllTickers. It is closely followed by GRU and LSTM, with most differences across metrics being in the range of 2-4%. We observe a marginal improvement in the performance of LSTM and GRU when the data are enriched with the presumably mislabeled neutrals detected in the preliminary phase. Regarding the other machine learning methods, ERT is the best performing model in both datasets.

Table 7: Pos-Neg Technology New

	B. Accuracy	Accuracy	Precision	F1	Recall
DistilBERT (finetuned)	**0.943**	**0.914**	**0.955**	**0.948**	**0.941**
GRU	0.913	0.914	0.914	0.915	0.915
LSTM with neutrals	0.911	0.910	0.911	0.911	0.911
GRU with neutrals	0.910	0.912	0.912	0.912	0.912
LSTM	0.905	0.904	0.905	0.905	0.905
ERT	0.893	0.896	0.896	0.895	0.893
Random Forest	0.885	0.885	0.885	0.885	0.885
Naive Bayes	0.837	0.837	0.837	0.837	0.837

Table 8: Pos-Neg General News

	B. Accuracy	Accuracy	Precision	F1	Recall
DistilBERT (finetuned)	**0.928**	**0.928**	**0.936**	**0.927**	**0.918**
LSTM with neutrals	0.906	0.906	0.906	0.905	0.905
LSTM	0.905	0.905	0.905	0.905	0.905
GRU with neutrals	0.901	0.900	0.900	0.900	0.900
GRU	0.900	0.900	0.900	0.900	0.900
SVM (Linear)	0.874	0.874	0.874	0.874	0.874
ERT	0.874	0.874	0.874	0.874	0.874
Random Forest	0.865	0.865	0.866	0.865	0.865
Naive Bayes	0.82	0.82	0.82	0.82	0.82

As expected, in the last evaluation with the tweets dataset (Table 9), the models pretrained with general news performed significantly better than the respective pretrained with technology news, due to the fact that the

Table 9: Tweets Dataset

	B. Accuracy	Accuracy	Precision	F1	Recall
DistilBert trained in General news	**0.902**	**0.914**	**0.943**	**0.937**	**0.932**
DistilBert trained in Tech news	0.821	0.873	0.872	0.913	0.957
LSTM trained in General news	0.834	0.839	0.850	0.842	0.838
GRU trained in General news	0.821	0.824	0.838	0.828	0.824
SVM (linear) in General news	0.820	0.821	0.791	0.801	0.820
FinBERT (fine-tuned)	0.779	0.730	0.816	0.740	0.730
GRU trained in Tech news	0.755	0.735	0.786	0.745	0.735
LSTM trained in Tech news	0.751	0.748	0.780	0.756	0.748
ERT train in General News	0.788	0.829	0.788	0.788	0.788
Random Forest trained in General news	0.748	0.8037	0.7745	0.7583	0.7477
Naive Bayes trained in General news	0.739	0.7092	0.7036	0.696	0.739
ERT trained in Tech News	0.725	0.776	0.737	0.730	0.725
Naive Bayes trained in Tech news	0.7144	0.6977	0.6836	0.6804	0.7144
Random Forest trained in Tech news	0.6879	0.7307	0.6848	0.6862	0.6879

former offers better generalization towards a larger variety of news. Once more, DistilBERT finetuned on general news achieves the best results, being about 8% ahead in Balanced Accuracy and 9% in F1 score compared to LSTM. Furthermore, DistilBERT, LSTM and GRU, as well as linear SVM, significantly outperforms FinBERT, even though the latter is finetuned for sentiment analysis on financial texts.

While it is clear that DistilBERT achieves the best metrics for both news and tweets data, we highlight that it does so at a certain cost. Specifically, the execution time results of LSTM, GRU and DistilBERT, provided in Table 10, do not favour DistilBERT. Compared to the training times of LSTM and GRU, DistilBERT is about 40% slower in general news and 145% slower in tech news. More importantly, inference for DistilBERT is about 13 times slower for general news, 8 times slower for tech news and 20 times for tweets.

Subsequently, there is a trade-off between a 2-4% increase in predictive performance of news sentiment analysis against several times faster inference, which is considerable given the time-critical nature of financial applications. With respect to highly competitive financial markets, more so for HFT environments, reaction time is paramount for capturing profit opportunities in time (Scholtus et al., 2014). Thus, financial experts that want to integrate sentiment analysis in algorithmic trading systems should further investigate the most profitable resolution to this trade-off.

Table 10: Execution times for training and inference for the sentiment analysis task (in seconds)

	General			Tech		
	LSTM	GRU	DistilBert	LSTM	GRU	DistilBert
Training (News)	291.933	**280.571**	398.135	126.907	**106.874**	260.282
Inference (News)	2.088	**1.704**	28.371	**1.26**	1.29	10.275
Inference (Tweets)	0.107	**0.102**	2.401	0.230	**0.104**	2.501

5.2. Sector prediction results

This section presents the performance results for the sector detection module, the goal of which is to retrieve the economic sector derived from the analysis of financial news content.

Notably, the dataset is imbalanced and we address this problem by applying appropriate weights to each classifier's training process. Regarding Naive Bayes, SVM and RF, we proceed to an extra method for handling imbalanced data, oversampling, but it lagged behind in performance against the weight method. Sector-level analysis results for the task of predicting the affected sector are presented in Table 11.

In this task, LSTM and GRU were the most reliable solution with remarkable balanced performance in any area of economic interest, closely followed by DistilBERT. Since LSTM and GRU are marginally ahead DistilBERT in terms of predictive performance, execution time should be the deciding factor for the best model, for reasons explained in the previous section.

The execution times are provided in Table 12. DistilBERT requires 3% and 65% more training time compared to LSTM and GRU, respectively. More importantly, inference for DistilBERT is about 70 times slower compared to LSTM and GRU, which could pose a significant impact for algorithmic trading systems. Overall, the experiments establish LSTM as the best option for sector prediction, achieving about 1% increase in most metrics compared to GRU and DistilBERT, with only marginally slower inference than GRU. Finally, linear SVM is the best performing model among the remaining methods.

5.3. Sector-level sentiment analysis results

The final step is to combine the previous models in an interconnected system, where our effort aims towards predicting both the sector of economic interest and sentiment. The combined predictions for both the sector and

Table 11: Sector Analysis

	B. Accuracy	Accuracy	Precision	Recall	F1
LSTM	**0.892**	**0.882**	**0.884**	**0.882**	**0.882**
GRU	0.888	0.876	0.879	0.876	0.877
DistilBERT	0.885	0.879	0.878	0.878	0.878
SVM (Linear - weights)	0.870	0.860	0.853	0.870	0.861
SVM (Linear - oversampled)	0.865	0.858	0.853	0.865	0.859
ERT (oversampled)	0.811	0.806	0.830	0.819	0.811
Naive Bayes	0.85	0.83	0.838	0.830	0.831
Random Forest (weights)	0.821	0.796	0.801	0.821	0.810
Random Forest(oversampled)	0.803	0.784	0.810	0.803	0.804

Table 12: Execution times for training and inference for the sentiment analysis task (in seconds)

	LSTM	GRU	DistilBERT
Inteference	1.228	**1.180**	85.618
Training	373.299	**232.405**	384.827

sentiment labels are aggregated in order to estimate the sector-level sentiment. Details regarding the computation of sector-level sentiment are given in Section 4.4. Furthermore, the interconnected system is compared to a single model multi-class approach for the simultaneous classification of sector and sentiment, which is also explained in Section 4.4. For the evaluation and comparison of the two approaches, we present the results for the metrics SSA, SSPE and MSSPE (see Section 4.5) in Table 13, as specified in Section 4.5. For the sentiment analysis and sector detection tasks of the interconnected system, as well as for the multi-class approach, we used LSTM.

In terms of accuracy, that is the percentage of news which had correct predictions for both sector and sentiment, the hybrid approach achieved about 79.3%, being noticeably better than the single multi-class model. Similarly, there is a difference of about 10% in the average sector-level sentiment percentage error between the two approaches in favour of the hybrid model.

With regards to the different sectors, the hybrid model features significantly better performance in all but the Technology sector. Furthermore, the multi-class approach does remarkably better in the Technology sector compared to its performance in other sectors. This result is in agreement with a previous assumption, that the definition of positive and negative sentiment

Table 13: Evaluation results for the two sector-level sentiment analysis approaches.

	Hybrid	Multiclass
SSA	**79.29%**	77.18%
Sector	SSPE	
Technology	5.097%	**4.538%**
Healthcare	**2.974%**	14.959%
Financial	**6.111%**	18.269%
Consumer goods	**2.792%**	13.102%
Energy	**2.755%**	19.480%
Commodity	**4.643%**	14.753%
MSSPE	**4.06%**	14.18%

may depend on the underlying sector, especially when comparing fast paced to more traditional sectors.

A possible explanation is that the multi-class approach is given the opportunity to model the particular *Technology-Positive*/*Technology-Negative* concepts, which probably differ from the general *Positive*/*Negative* concepts. In contrast, the hybrid model does not consider any sector information during the sentiment analysis stage, and thus captures only the latter, more general concepts. Nevertheless, the hybrid model is the best performing overall, which indicates that the concept of sentiment does not vary as much for the remaining five sectors.

6. Discussion

In each task of the experimental process, our main goal was to achieve the best possible results by using a large number of experiments and by exploring an exhaustive combination of hyperparameters for each classifier. Common key factors, such as the train-test split used for each method, were held fixed in order to ensure unbiased results.

In the sentiment analysis task, DistilBERT stood out significantly in predictive performance compared to the other classifiers, presenting consistently remarkable performance at each stage of the experimental process. Yet, it was several times slower than LSTM and GRU, which could pose a limit with respect to capturing market opportunities in time. All three models were able to outperform FinBERT on the independent tweets dataset, even though the latter has been pretrained on a large corpus consisting of financial texts. When preceded by the additional task of label augmentation, by correcting news miss-classified as neutral, LSTM and GRU featured marginally improved performance. In the sector detection task, LSTM was clearly the best performing model, achieving the best combination of performance and execution time.

Furthermore, we argue that a system that can only extract sentiment from financial news is incomplete. The rationale of the argument is that in order to be useful, the extracted sentiment still has to be associated with specific sectors, industries or stocks. Towards this direction, our sector detection module proved effective in matching news with the sectors affected by the same news. We consider our work of detecting the affected sectors as a stepping stone towards a more granular detection of the particular industries or stocks.

Sector detection, while being a natural step towards predicting the affected industries or stocks, also has its own significant applications. By combining the sector detection model with the news-level sentiment analysis model, the designed solution was able to extract the general sentiment that prevailed in six sectors. The evaluation of the sector-level sentiment analysis approach yielded promising results. Thus, we consider the proposed sector-level sentiment analysis system as useful towards understanding broad-level sentiment trends, with potential application in forecasting the behavior of sector Exchange Traded Funds (sector ETFs).

7. Conclusion

Financial news is becoming an increasingly important source of data for investors who want to determine market sentiment. As the accuracy and speed of understanding these texts are paramount capabilities, both research and industry are considering computational methods that can automatically extract valuable information. This paper proposes three applications of NLP in this domain. Additionally, even though a multitude of machine learning algorithms were evaluated for all applications, our results showed that methods based on deep learning prevailed in every case.

The initial application regards the sentiment analysis of financial news. In this direction, we propose two computational modules, one for a preliminary label augmentation task and the other for the actual sentiment analysis task. To begin with, we use an ensemble of RNNs to detect news that has most likely been mislabeled as neutral. Secondly, we use LSTM to classify news as positive or negative, which also considers the mislabeled neutrals with a semi-supervised learning approach. Using an independent test set consisting of financial tweets, the predictive performance of our sentimental analysis module is favourably compared against a cutting-edge pre-trained language model, namely FinBERT.

The second problem we addressed is the detection of the particular sector(s) affected by news. We presume that this task is an indispensable complement to the prior sentiment analysis task, because the extracted sentiment still has to be associated with a particular sector, industry or ticker. For this purpose, we propose an additional multi-class LSTM model trained to classify news into six economic sectors. The results showed adequate and balanced performance for all sectors, proving that deep learning can be used to predict the affected sectors.

In the last direction, the sentiment analysis and sector detection models have been combined into a hybrid system. The purpose of this system is to perform sector-level sentiment analysis, with potential applications in gauging broad market sentiment trends and the behavior of sector ETFs. The hybrid system outperformed a single multi-class model in the task of predicting sector-level sentiment, reaching about 80% accuracy.

Based on the aforementioned results, we suggest several directions for future research. First of all, we consider the sector detection model as a stepping stone towards more fine-grained models, that can predict industries or particular stocks affected by news. Essentially, the proposed sector detection model can narrow the search space of these tasks. This could result in systems that can automatically associate the sentiment extracted from news with particular industries or stocks.

Secondly, as sector-level sentiment analysis predictions are designed to gauge market sentiment, they could be further studied as potential indicators for forecasting the price trends of sector ETFs. Thirdly, an investigation of the trade-off between execution time and predictive performance in sentiment analysis of financial texts should be done, in the context of algorithmic trading profits. Finally, we plan to explore whether the addition of CNN layers and attention mechanisms in the proposed system improves sector-level sentiment analysis results.

Acknowledgements

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Declarations of Interest

None

References

Ahmad, K., 2011. The 'Return' and 'Volatility' of Sentiments: An Attempt to Quantify the Behaviour of the Markets?. Springer Netherlands, Dordrecht. pp. 89–99. URL: https://doi.org/10.1007/978-94-007-1757-2_8, doi:10.1007/978-94-007-1757-2_8.

Arthur, D., Vassilvitskii, S., 2007. K-means++: The advantages of careful seeding, in: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, USA. p. 1027–1035. doi:10.1145/1283383.1283494.

Audrino, F., Sigrist, F., Ballinari, D., 2020. The impact of sentiment and attention measures on stock market volatility. International Journal of Forecasting 36, 334–357. URL: https://www.sciencedirect.com/science/article/pii/S0169207019301645, doi:https://doi.org/10.1016/j.ijforecast.2019.05.010.

Baker, M., Wurgler, J., 2006. Investor sentiment and the cross-section of stock returns. The Journal of Finance 61, 1645–1680. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1540-6261.2006.00885.x, doi:https://doi.org/10.1111/j.1540-6261.2006.00885.x.

Basiri, M.E., Nemati, S., Abdar, M., Cambria, E., Acharya, U.R., 2021. Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis. Future Generation Computer Systems 115, 279–294. URL: https://www.sciencedirect.com/science/article/pii/S0167739X20309195, doi:https://doi.org/10.1016/j.future.2020.08.005.

Bollen, J., Mao, H., Zeng, X., 2011. Twitter mood predicts the stock market. Journal of Computational Science 2, 1–8. URL: https://www.sciencedirect.com/science/article/pii/S187775031100007X, doi:https://doi.org/10.1016/j.jocs.2010.12.007.

Cambria, E., Das, D., Bandyopadhyay, S., Feraco, A., 2017. Affective Computing and Sentiment Analysis. Springer International Publishing, Cham. pp. 1–10. URL: https://doi.org/10.1007/978-3-319-55394-8_1, doi:10.1007/978-3-319-55394-8_1.

Chan, W., 2003. Stock price reaction to news and no-news: Drift and reversal after headlines. Journal of Financial Economics 70, 223–260. doi:10.1016/S0304-405X(03)00146-6.

Chua, C., Milosavljevic, M., Curran, J.R., 2009. A sentiment detection engine for internet stock message boards, in: Proceedings of the Australasian Language Technology Association Workshop 2009, pp. 89–93.

Cohen, M., Damiani, P., Durandeu, S., Navas, R., Merlino, H., Fernández, E., 2011. Sentiment analysis in microblogging: a practical implementation, in: XVII Congreso Argentino de Ciencias de la Computación.

Day, M.Y., Lee, C.C., 2016. Deep learning for financial sentiment analysis on finance news providers, in: 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), pp. 1127–1134. doi:10.1109/ASONAM.2016.7752381.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1, Association for Computational Linguistics, Minneapolis, Minnesota. pp. 4171–4186. URL: https://aclanthology.org/N19-1423, doi:10.18653/v1/N19-1423.

Dey, R., Salem, F.M., 2017. Gate-variants of gated recurrent unit (gru) neural networks, in: 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), pp. 1597–1600. doi:10.1109/MWSCAS.2017.8053243.

Ederington, L.H., Lee, J.H., 1993. How markets process information: News releases and volatility. The Journal of Finance 48, 1161–1191. URL: http://www.jstor.org/stable/2329034, doi:10.2307/2329034.

Farkash, E., Magen, N., Waisbard, E., Hibshoosh, E., 2015. Computer-implemented method and apparatus for encoding natural-language text content and/or detecting plagiarism. US Patent 9,213,847.

Feldman, R., 2013. Techniques and applications for sentiment analysis. Commun. ACM 56, 82–89. doi:10.1145/2436256.2436274.

Floridi, L., Chiriatti, M., 2020. Gpt-3: Its nature, scope, limits, and consequences. Minds and Machines 30, 681–694. doi:10.1007/s11023-020-09548-1.

Geurts, P., Ernst, D., Wehenkel, L., 2006. Extremely randomized trees. Machine learning 63, 3–42. doi:https://doi.org/10.1007/s10994-006-6226-1.

Généreux, M., Poibeau, T., Koppel, M., 2011. Sentiment Analysis Using Automatically Labelled Financial News Items. volume 45. pp. 101–114. doi:10.1007/978-94-007-1757-2_9.

Huang, A., Wang, H., Yang, Y., 2020. Finbert—a large language model approach to extracting information from financial text URL: https://ssrn.com/abstract=3910214, doi:http://dx.doi.org/10.2139/ssrn.3910214.

Huang, Z., Xu, W., Yu, K., 2015. Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991. URL: http://arxiv.org/abs/1508.01991, arXiv:1508.01991.

Jain, V.K., Kumar, S., Fernandes, S.L., 2017. Extraction of emotions from multilingual text using intelligent text processing and computational linguistics. Journal of Computational Science 21, 316–326. URL: https://www.sciencedirect.com/science/article/pii/S1877750317301035, doi:https://doi.org/10.1016/j.jocs.2017.01.010.

Klein, F., Prestbo, J., 1974. News and the Market. H. Regnery Company. URL: https://books.google.gr/books?id=B2APAQAAMAAJ.

Kothari, S.P., Li, X., Short, J.E., 2009. The effect of disclosures by management, analysts, and business press on cost of capital, return volatility, and analyst forecasts: A study using content analysis. The Accounting Review 84, 1639–1670. URL: http://www.jstor.org/stable/27784235, doi:10.2308/accr.2009.84.5.1639.

Li, W., Zhu, L., Cambria, E., 2021. Taylor's theorem: A new perspective for neural tensor networks. Knowledge-Based Systems 228, 107258. URL: https://www.sciencedirect.com/science/article/pii/S0950705121005207, doi:https://doi.org/10.1016/j.knosys.2021.107258.

Loughran, T., McDonald, B., 2013. Ipo first-day returns, offer price revisions, volatility, and form s-1 language. Journal of Financial Economics 109, 307–326. URL: https://www.sciencedirect.com/science/article/pii/S0304405X13000603, doi:https://doi.org/10.1016/j.jfineco.2013.02.017.

Malandri, L., Xing, F.Z., Orsenigo, C., Vercellis, C., Cambria, E., 2018. Public mood–driven asset allocation: The importance of financial sentiment in portfolio management. Cognitive Computation 10, 1167–1176. doi:10.1007/s12559-018-9609-2.

Melvin, M., Yin, X., 2000. Public information arrival, exchange rate volatility, and quote frequency. The Economic Journal 110, 644–661. URL: http://www.jstor.org/stable/2565919.

Miller, G.A., 1995. Wordnet: A lexical database for english. Commun. ACM 38, 39–41. URL: https://doi.org/10.1145/219717.219748, doi:10.1145/219717.219748.

Pang, B., Lee, L., Vaithyanathan, S., 2002. Thumbs up? sentiment classification using machine learning techniques, in: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, Association for Computational Linguistics, USA. p. 79–86. URL: https://doi.org/10.3115/1118693.1118704, doi:10.3115/1118693.1118704.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library, in: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., Louppe, G., 2012. Scikit-learn: Machine learning in python. Journal of Machine Learning Research 12, 2825–2830.

Pejic Bach, M., Krstic, Z., Seljan, S., Turulja, L., 2019. Text mining for big data analysis in financial sector: A literature review. Sustainability 11, 1277. doi:10.3390/su11051277.

Picasso, A., Merello, S., Ma, Y., Oneto, L., Cambria, E., 2019. Technical analysis and sentiment embeddings for market trend prediction. Expert Systems with Applications 135, 60–70. URL: https://www.sciencedirect.com/science/article/pii/S0957417419304142, doi:https://doi.org/10.1016/j.eswa.2019.06.014.

Rechenthin, M., Street, W.N., Srinivasan, P., 2013. Stock chatter: Using stock sentiment to predict price direction. Algorithmic Finance 2, 169–196. doi:10.3233/AF-13025.

Saif, H., He, Y., Alani, H., 2012. Semantic sentiment analysis of twitter, in: The Semantic Web – ISWC 2012, Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 508–524. doi:10.1007/978-3-642-35176-1_32.

Sanh, V., Debut, L., Chaumond, J., Wolf, T., 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, in: The 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS. doi:https://doi.org/10.48550/arXiv.1910.01108.

Scholtus, M., van Dijk, D., Frijns, B., 2014. Speed, algorithmic trading, and market quality around macroeconomic news announcements. Journal of Banking & Finance 38, 89–105. URL: https://www.sciencedirect.

com/science/article/pii/S0378426613003841, doi:https://doi.org/10.1016/j.jbankfin.2013.09.016.

Schumaker, R.P., Chen, H., 2009a. A quantitative stock prediction system based on financial news. Information Processing & Management 45, 571–583. URL: https://www.sciencedirect.com/science/article/pii/S0306457309000478, doi:https://doi.org/10.1016/j.ipm.2009.05.001.

Schumaker, R.P., Chen, H., 2009b. Textual analysis of stock market prediction using breaking financial news: The azfin text system. ACM Transactions on Information Systems (TOIS) 27, 1–19. doi:10.1145/1462198.1462204.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. Journal of Machine Learning Research 15, 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Srividhya, V., Anitha, R., 2010. Evaluating preprocessing techniques in text categorization. International journal of computer science and application 47, 49–51.

Sun, F., Belatreche, A., Coleman, S., McGinnity, T.M., Li, Y., 2014. Preprocessing online financial text for sentiment classification: A natural language processing approach, in: 2014 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr), pp. 122–129. doi:10.1109/CIFEr.2014.6924063.

Thelwall, M., Buckley, K., Paltoglou, G., Cai, D., Kappas, A., 2010. Sentiment strength detection in short informal text. Journal of the American Society for Information Science and Technology 61, 2544–2558. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/asi.21416, doi:https://doi.org/10.1002/asi.21416, arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1002/asi.21416.

Valdivia, A., Luzón, M.V., Cambria, E., Herrera, F., 2018. Consensus vote models for detecting and filtering neutrality in sentiment analysis. Information Fusion 44, 126–135. URL: https://www.sciencedirect.

com/science/article/pii/S1566253517306590, doi:https://doi.org/10.1016/j.inffus.2018.03.007.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vishwanathan, S., Narasimha Murty, M., 2002. Ssvm: a simple svm algorithm, in: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), pp. 2393–2398 vol.3. doi:10.1109/IJCNN.2002.1007516.

Wang, G., Sun, J., Ma, J., Xu, K., Gu, J., 2014. Sentiment classification: The contribution of ensemble learning. Decision Support Systems 57, 77–93. URL: https://www.sciencedirect.com/science/article/pii/S0167923613001978, doi:https://doi.org/10.1016/j.dss.2013.08.002.

Wang, Z., Ho, S.B., Cambria, E., 2020. Multi-level fine-scaled sentiment sensing with ambivalence handling. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 28, 683–697. URL: https://doi.org/10.1142/S0218488520500294, doi:10.1142/S0218488520500294.

Wiebe, J., Mihalcea, R., 2006. Word sense and subjectivity, in: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Sydney, Australia. pp. 1065–1072. URL: https://aclanthology.org/P06-1134, doi:10.3115/1220175.1220309.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., Rush, A., 2020. Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Association for Computational Linguistics, Online. pp. 38–45.

URL: https://aclanthology.org/2020.emnlp-demos.6, doi:10.18653/v1/2020.emnlp-demos.6.

Wüthrich, B., Permunetilleke, D., Leung, S., Lam, W., Cho, V., Zhang, J., 1998. Daily prediction of major stock indices from textual www data. HKIE Transactions 5, 151–156. doi:10.1080/1023697X.1998.10667783.

Xing, F.Z., Cambria, E., Welsch, R.E., 2018. Intelligent asset allocation via market sentiment views. IEEE Computational Intelligence Magazine 13, 25–34. doi:10.1109/MCI.2018.2866727.

Xing, F.Z., Cambria, E., Zhang, Y., 2019. Sentiment-aware volatility forecasting. Knowledge-Based Systems 176, 68–76. URL: https://www.sciencedirect.com/science/article/pii/S0950705119301546, doi:https://doi.org/10.1016/j.knosys.2019.03.029.

Xu, B., Guo, X., Ye, Y., Cheng, J., 2012. An improved random forest classifier for text categorization. Journal of Computers 7, 2913–2920. doi:10.4304/jcp.7.12.2913-2920.

Zhang, H., 2004. The optimality of naive bayes, in: Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference, FLAIRS 2004, pp. 1–10.

Zhang, Y., Swanson, P., 2010. Are day traders bias free? evidence from internet stock message boards. Journal of Economics and Finance 34, 96–112. doi:10.1007/s12197-008-9063-1.