CENG 396

Literature Review of

Entity Based Turkish Financial

Sentiment Analysis System

October 2024

Team 20

Taner Onur Uyar

Ahmet Gökay Ürkmez

Mert Şerafettin Kargı

Ahmet Eren Yağlı

Baha Öçalan

08/11/2024

1. Introduction

Turkish Financial Sentiment Analysis (SA) is crucial for understanding the impact of investor and public sentiment on stock prices and market dynamics. On digital platforms like social media, investors' emotions can directly influence market behavior. In the Turkish financial sector, sentiment analysis aids in predicting investor behavior, managing risk, and supporting strategic decision-making processes.

The goal of this project is to analyze Twitter/X data related to Turkish companies to identify positive, negative, or neutral sentiments. This analysis helps in gaining insights into market trends and measuring public perception of companies, providing a strategic information source for financial decision-makers and investors.

2. Sentiment Analysis

Sentiment analysis is an approach used to understand the mood and attitude expressed in texts. Twitter can be thought of as a microblogging platform with a diverse demographic structure, consisting of users from various cultures with countless different dialects and jargon. When texts are written informally or contain social media jargon, sentiment analysis requires a more careful and detailed approach [1], often necessitating the inclusion of emojis and abbreviations in the algorithms. The texts under consideration are classified as positive, neutral, or negative. In this section, we will discuss sentiment analysis, Turkish sentiment analysis, and sentiment analysis in financial data.

2.1 Sentiment Analysis Levels

Sentiment analysis can be approached at four levels: Document Level, Sentence Level, Phrase Level, and Aspect Level [2]. For our project, the Aspect Level approach is more suitable, as it is preferred for more specific subjects such as products, stocks, and sectors. Therefore, it is deemed appropriate for tracking the sentiment associated with BIST100 companies. These levels are summarized below.

a. Document Level

At this level, sentiment analysis is performed on the entire document, assigning a single polarity (positive, negative, or neutral) to the entire text. It is commonly used to classify large blocks of text, like book chapters or pages. Both supervised and unsupervised learning methods can be applied at this level. However, this approach is often domain-dependent and may be inadequate for more focused analyses.

b. Sentence Level

In sentence-level analysis, each sentence is evaluated individually, and a sentiment polarity is assigned to each. This is especially useful when a document has a mix of sentiments. Sentence-level analysis requires more training data and processing resources than document-level analysis. It is also crucial for handling more complex tasks, such as conditional sentences or ambiguous statements.

c. Phrase Level

Phrase-level analysis identifies and classifies the sentiment of phrases within the text. This approach is beneficial for texts like multi-line product reviews, where each phrase may convey a distinct sentiment. Phrases are generally shorter and may carry clearer sentiments than full sentences, and this level can also provide insights into demographic and socio-psychological characteristics.

d. Aspect Level

In aspect-level analysis, sentiment is assigned to each aspect within a sentence (such as specific products or stocks). Multiple aspects may exist within a single sentence, and each aspect is analyzed individually for sentiment. This allows for a detailed understanding of sentiment towards each aspect, resulting in an aggregated sentiment for the entire sentence.

2.2 Sentiment Analysis Process

a. Data Extraction

The first step in Sentiment Analysis is collecting or creating text data for analysis. This data can come from third-party sources, web scraping, or various data types.

b. Data Pre-processing

Unstructured data is cleaned and reduced in size to prepare it for analysis. Steps include:

- **Tokenization:** Breaking text into smaller units.
- **Stop Words Removal:** Removing irrelevant words.
- **PoS Tagging and Lemmatization:** Identifying structural elements and root forms of words.

c. Feature Extraction

i. Bag of Words (BoW)

The BoW model converts text into a numerical vector by creating a vocabulary of unique words and encoding sentences based on word frequency or count. However, it ignores word order, sentence structure, and syntax, which can reduce contextual understanding. TF–IDF is a common extension of BoW that considers the importance of words in context.

ii. Distributed Representation

Distributed representation (or word embedding) represents words in vector space where each vector position contributes information. Common methods include:

- **Word2Vec:** A neural network model with **CBOW** (predicts current word from context) and **Skip-Gram** (predicts context from current word).
- **GloVe:** Generates word embeddings by capturing global word co-occurrence information from a corpus, with fast parallelized training.

Other methods include **Doc2Vec** and **FastText**, which capture more contextual information.

d. Feature Selection

Feature selection removes irrelevant or redundant features to improve classification accuracy. Main methods include:

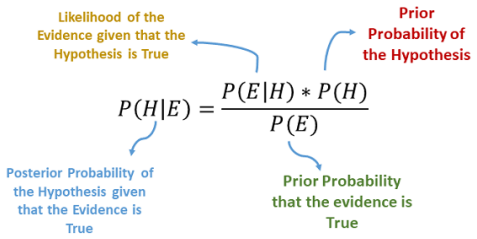
- **Filter Approach:** Selects features based on data properties, without using machine learning. Common measures include **Information Gain**, **Chi-square**, and **Mutual Information**.
- **Wrapper Approach:** Evaluates feature subsets based on machine learning performance, but is computationally intensive. Uses methods like **Naïve Bayes** and **SVM** with feature subset generation strategies.
- **Embedded Approach:** Integrates feature selection into the model training process, often using decision tree algorithms (e.g., **CART**, **C4.5**) and **LASSO**.
- **Hybrid Approach:** Combines filter and wrapper methods to balance performance and efficiency, often yielding high accuracy for sentiment analysis.[3]

2.3 Sentiment Analysis Methods

a. Machine learning-based approaches

Traditional methods like Naive Bayes and Logistic Regression are widely preferred in sentiment analysis projects due to their speed and relatively low computational requirements [4].

Naive Bayes is known for its efficiency, especially with large datasets. This method classifies text based on probabilities, assuming that each word is independent of the others. Despite its simplicity, it can yield effective results in text classification and sentiment analysis. However, its assumption of word independence can sometimes limit its sensitivity, as it overlooks dependencies within the text.

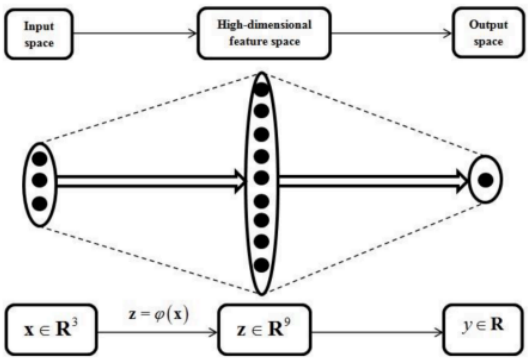


Logistic Regression attempts to classify data using a linear model and calculates the probabilities for class labels. This model can perform well with text data, particularly on datasets with a limited number of features. According to the article, both methods are especially favored for quick prototyping and basic sentiment analysis tasks, though they may fall short compared to deep learning-based methods like LSTM and CNN.

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

$$J(\theta) = \frac{1}{m}[\sum_{i=1}^{m} -y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)}) log(1 - h_\theta(x^{(i)}))]$$

Support Vector Machines (SVMs), developed by Vapnik, have gained increasing acceptance due to their appealing features and strong performance. This method is based on the principle of structural risk minimization, rather than the empirical risk minimization used in traditional machine learning methods, resulting in superior performance. Initially designed to solve classification problems, SVMs have since been adapted to the domain of regression problems as well.



b. Deep learning-based approaches

Deep learning, a subset of machine learning, utilizes deep neural networks and has been widely adopted in sentiment analysis due to its capacity to model complex patterns in text data. Numerous studies review and compare deep learning methods for sentiment analysis. For example, Dang et al. (2020) analyzed 32 papers and assessed the performance of models such as Deep Neural Networks (DNN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN) on multiple datasets. They found that RNN models, particularly with word embeddings, achieved the best results, though at a much higher computational cost compared to CNNs.

Common deep learning models applied in sentiment analysis include CNNs, Long Short-Term Memory (LSTM) networks, and Gated Recurrent Units (GRU). LSTM networks are especially effective for capturing long-term dependencies, making them suitable for sentiment analysis tasks, as noted by Yadav and Vishwakarma (2019), who also highlighted the promise of attention-based networks and Capsule Networks in improving sentiment classification accuracy.

Table 3 Distribution of deep learning papers published in 2020

ID	Algorithm	# of usages	Percentage (%)
1	LSTM	81	35.53
2	CNN	76	33.33
3	GRU	20	8.77
4	RNN	18	7.89
5	Bidirectional Encoder Representations from Transformers (BERT)	7	3.07
6	DNN	4	1.75
7	ReNN	4	1.75
8	Graph Convolutional Neural Network (GCN)	3	1.32
9	Capsule Network (CapsN)	2	0.88
10	Recurrent Convolutional Neural Network (RCNN)	2	0.88
11	Distillation Network (DN)	2	0.88
12	Generative Adversarial Network (GAN)	1	0.44
13	Gated Alternate Neural Network (GANN)	1	0.44
14	Category Attention Network (CAN)	1	0.44
15	Recurrent Memory Neural Network (ReMemNN)	1	0.44
16	Interactive Rule Attention Network (IRAN)	1	0.44
17	Self-Attention based Hierarchical Dilated Convolutional Neural Network (SA-HDCNN)	1	0.44
18	Fusion-Extraction Network (FENet)	1	0.44
19	Deep Q-Network	1	0.44
20	Autoencoder	1	0.44
	Total	228	100

c. Lexicon-Based Approaches

Lexicon-based approaches identify sentiment by scanning for positive or negative words listed in lexicons. This method doesn't require training data but is domain-sensitive, as different domains (e.g., movie reviews vs. Twitter) need tailored lexicons. Lexicons are built via dictionary-based or corpus-based approaches, where the former uses synonym/antonym expansion and the latter identifies sentiment words through co-occurrence in domain-specific corpora.

d. Hybrid Approaches

e. Hybrid approaches combine machine learning and lexicon-based techniques to leverage the strengths of both methods. Recent research integrates symbolic and subsymbolic AI, applying deep learning to detect patterns and symbolic AI to build commonsense knowledge bases like SenticNet. These approaches aim to enhance natural language understanding by combining pattern recognition with structured knowledge. Hybrid models like LSTM-CNN ensembles also show improved sentiment analysis performance by integrating both model types' strengths.[5]

2.4 Financial Sentiment Analysis

What is Financial Sentiment Analysis?

Financial Sentiment Analysis is a technique used to understand investor sentiment in financial data, particularly in stock predictions. By analyzing data from platforms like Twitter, this approach examines the relationship between stock prices and investor moods. Studies show that positive sentiment in tweets can correlate with upward trends in certain stocks, making sentiment analysis a valuable tool in financial forecasting [6].

Key Studies and Methods

Research in this field explores how social media sentiment affects stock prices, using techniques from natural language processing (NLP) and machine learning. A study published by PLOS ONE analyzed the effects of Twitter sentiment on stocks within the Dow Jones Industrial Average. This research utilized event-based analysis to identify significant short-term correlations between sentiment and stock returns during specific events. Such event-based studies indicate that sentiment-driven data, especially during key events, can provide valuable insights into stock valuation [7].

In another notable study, the Federal Reserve developed the Twitter Financial Sentiment Index (TFSI) to observe correlations between financial sentiment and economic variables such as equity returns, bond spreads, and monetary policy. The TFSI leverages FinBERT, a specialized NLP model designed to analyze sentiment in financial text. Models like FinBERT, RNNs, and LSTMs are effective in capturing long-term dependencies in text, making them suitable for understanding market reactions based on sentiment.

These studies highlight the role of advanced NLP and deep learning methods in financial sentiment analysis, demonstrating how models like LSTM, RNN, and BERT can help predict sentiment-based stock movements by analyzing social media data.

3. Natural Language Processing (NLP) in Turkish Texts and Special Challenges

Studies in Turkish natural language processing (NLP), especially in certain areas such as financial texts, are quite interesting and complex. The agglutinative structure of Turkish, ambiguities and implicit emotional expressions are some of the main difficulties encountered in natural language processing processes.

- ## Agglutinative Structure of Turkish

  Derivation of roots with various suffixes in Turkish offers many possibilities that change the meaning and function of the word. This makes morphological parsing difficult, especially for language models. Any word can have hundreds of derivations, which makes word vectors and language comprehension models more complex. Morphological parsing and root finding methods for NLP models should therefore be considered more comprehensively.

- ## Ambiguities of Meaning

  In Turkish, the same word can have different meanings with different suffixes or even with the same suffix. In financial texts, terms like “interest” or “stock market” can have different meanings depending on the context. This can be challenging for NLP models trying to understand the correct context. Semantic parsing techniques in particular need to be effective in resolving these ambiguities.

- ## Implicit Emotional Expressions

  Emotional expressions in Turkish texts are usually indirect and can be interpreted differently depending on the context. Identifying positive or negative emotional expressions in financial texts is important for decision support systems. Therefore, emotional analysis in Turkish NLP usually requires higher-level features and requires models that require fine-tuning.

3.1 Examples of NLP Studies in Turkish Financial Texts in Literature

a. Morphological Segmentation Techniques

Due to the rich morphological structure of Turkish, most NLP methods used in financial texts are based on morphological analysis. Tools such as Zemberek or TRmorph are used to separate the roots and suffixes of Turkish words. Especially in the analysis of financial reports, correct parsing is of great importance.[8][9]

- **ZEMBEREK**
  What is Zemberek-NLP?

  Zemberek-NLP is an open-source Turkish natural language processing library. It was specifically developed to understand the complex language structure of Turkish and to make sense of texts. The library can perform morphological analysis of Turkish texts, grammar checking, sentence parsing and many more operations. Zemberek is an indispensable resource for Turkish natural language processing projects.

  Features of Zemberek

  - ✔ Can parse sentences in Turkish texts and identify the structural elements of the sentence.
  - ✔ Perform morphological analysis of Turkish texts. Morphological analysis provides detailed information about the roots, stems, suffixes and grammatical information of a word.
  - ✔ It is effective for correcting spelling errors in Turkish texts.
  - ✔ It breaks the text into understandable parts and separates the words in the text.
  - ✔ It provides enhanced tools for language analysis and machine learning-based tasks.
  - ✔ Easy to integrate and use in language processing projects

- **TRmorph**
  What is TRmorph?

  TRmorph is a relatively complete finite-state morphological analyzer for Turkish. The current version of the analyzer is licensed under the terms of GNU LGPL.

b. Sentiment Analysis

Studies on Turkish financial news and social media posts often include sentiment analysis algorithms. Language models such as Turkish BERT or XLM-R, trained according to the specific language structure of Turkish, are used to predict sentiment trends in financial news.[10][11]

- **BERT**
  What is BERT?

  The BERT Algorithm, like many other algorithm updates by Google, was developed to better understand queries and provide more accurate results to its users. The BERT algorithm can be explained as a natural language processing technique that uses artificial intelligence and machine learning technologies together.
  For example, before the BERT algorithm update, when a Brazilian traveler made this query on Google about a US visa, they would encounter the Washington Post's news. Because Google could not correctly understand the user's intention here and presented the user with such a result. However, after the BERT algorithm understood the query more accurately, especially with the "to" conjunction in the query, and started to directly present the embassy and consulate pages to the users.

- **XLM-R**
  What is XLM-R?

  It is a natural language processing model developed by Facebook AI and designed to improve the performance of language modeling across languages. As a multilingual version of the RoBERTa model, it has been pre-trained on 100 different languages, including Turkish. XLM-R shows strong performance on texts in different languages, especially by bridging the gap between language structures and meanings. The model is suitable for working on morphologically rich and structurally diverse languages such as Turkish.

  In context-sensitive texts such as Turkish financial news, XLM-R can analyze sentiment trends more accurately thanks to its contextual interpretation power. This model is widely used in tasks such as sentiment analysis, text classification, and translation.

c. Word Embeddings

Word representation models such as Word2Vec and GloVe are used to better capture the meaning in Turkish financial texts. In particular, domain-specific embeddings are useful in capturing the complex semantic relationships in Turkish financial language.[12][13]

- **Word2Vec**
  What is Word2Vec?

  Word2Vec is an unsupervised and prediction-based model that tries to express words in vector space and there are two main methods used to convert words into vectors.

  CBOW: This method uses surrounding words to guess a particular word, thus learning its meaning by considering the context around the words.
  Skip-Gram: This method tries to predict surrounding words using a selected word. It can be more effective than CBOW, especially on low datasets.

- **GloVe**
  What is GloVe?

  It is a model developed by Stanford University that considers the global context of words. Instead of relying on neighboring words in context like Word2Vec, GloVe creates a vector based on the common occurrences of words in the entire text. In this way, it determines the meanings of words by obtaining more information from their overall distribution.

4. Named Entity Recognition (NER)

Named Entity Recognition is used to extract information, question answering and sentiment analysis by getting strings which includes sentence, paragraph, or document, recognizing and classifying the key entities that belong to each category. Names of people, organizations, locations, stocks and dates are some examples.

4.1. NER in Financial Texts

The need for clear identification and classification of entities like market terms, companies, financial reports, and news makes Named Entity Recognition more crucial in finance. Because NER allows efficient data extraction by identifying and categorizing key entities, this supports a wide range of financial tasks like sentiment analysis, news and report classification and risk prediction.

4.2. NER in Turkish Texts

Using NER for Turkish language is quite challenging compared to English. This is because for few reasons. [14][15]

A. Agglutination

In Turkish, words are formed by adding multiple suffixes to the root word. Also, a word can have more than one suffixes, and this makes morphological variations that affects the complexity of NER system. Sometimes this can create long and hard to understand word forms for traditional tokenizers.

B. Word Order

Turkish is flexible about word order. It means position of key entities can change sentence to sentence.

C. Lack of Resources

Turkish has fewer pre-existing resources for Named Entity Recognition like pre-trained models and articles. The lack of resources makes harder to train a NER model for Turkish language.

4.3. NER Methods[16]

A. Rule-Based Methods

This approach is about identifying and creating a set of rules for the grammar of that language. These rules are used to classify entities in the text considering the grammar rules. Rule base methods are great in specific domains where entities are identified clearly, but they can be time consuming and hard to work with large data.

B. Statistical and Machine Learning Methods[17]

Statistical Methods predicts named entities based on training data. Like Hidden Markov Models (HMM) and Conditional Random Fields (CRF). HMM predicts sequence of tags based on observed data. CRF captures dependencies between words in text. They are great as long as training data is good. In Machine learning various algorithms like decision trees or support vector machines (SVM) can be used

C. Deep Learning Based Methods

Recurrent Neural Networks (RNN) and transformer models have ability to model long-term dependencies in the text. To handle with rare words, they capture sequential data. Their best use case is large scale tasks with abundant training data.

D. Hybrid Methods

Combines rules-based and machine learning methods to use the benefits of both. Despite the complexity, this method offers flexibility for various sources. Rule-based can quickly identify entities that are easy to recognize and ML to identify entities that are more complex.

4.4 NER Process Steps[18]

A. Data Collection

The first step is making a clear dataset. Dataset contains examples of text with named entities.

B. Data Preprocessing

After the data is collected, data should be cleared from unnecessary words and characters. Then split the text into tokens to work with each part individually. Lemmatization means reducing the word to their root forms and it can be done if it is necessary.

C. Feature Extraction

This step includes word embeddings, contextual features, and lexical features. In word embeddings, words are represented as vectors embeddings to capture semantic

meaning. Contextual Features means considering near words for each token, and that gives cues for identifying entities. Feature choices can depend on the NER model that match with the system needs.

D. Model Training

The next step is training a NER model with labeled dataset using ML or deep learning. The Model will learn to recognize patterns and relationships between words in text with their related named entity labels.

E. Model Evaluation

Model evaluated with performance metrics like recall, precision, or F1-score. These metrics Then with the evaluation data, errors or low performance can be revealed and further improvements can be made.

F. Model Improvements

In this step, based on the evaluation data, improvements are made for performance and errors if there is. Training data or model can be modified and adjusted accordingly for the requirements. Entity linking or normalization can be made if output of the NER model needs to be refined for better results.

G. Result

In the last step, model can be tested on a new text. The model will apply preprocessing steps and extract needed data from the text, then the model will predict the named entity labels for each token in the text.

5. Data Collection and Preprocessing

5.1 Data Sources and Data Collection

It is possible to obtain the Twitter (X) and BIST100 data that we will use in this project by various methods. Some of these methods are:

5.1.1 Borsa Istanbul Data API: In the database and API services offered by Borsa Istanbul, APIs allow you to obtain the prices, volumes, and index values that companies trade at daily or at certain intervals. Obtaining data through official APIs or data services can be one of the most definitive ways to ensure the reliability of the data.

5.1.2 Twitter (X) API: It is possible to collect tweets using specific keywords, hashtags or accounts via Twitter API provided by Twitter (X). Access to past tweets can be accessed with Academic Research Track and a wider data set can be obtained. In addition, Twitter's paid Premium Search API can be used to pull even older tweets or more data. Premium Search API allows you to perform more in-depth historical

searches and can provide data over a wider date range. Main API Functions: BIST100 company names, symbols or specific hashtags can be searched with Search Tweets endpoint. (e.g. “$BIST” or “#BIST100”). Tweets containing specific keywords can be continuously pulled with Filtered Stream Endpoint. In addition, access to the API can be provided using Python libraries such as Tweepy or . These libraries make it easier to pull tweets and structure the data.

5.1.3 Web Scraping: Web scraping is defined as the process of extracting data from pages on the internet. These processes are generally used for data analysis and research. Since it is possible to access current or historical data about BIST100 companies on many financial sites, web scraping is one of the alternatives in the data collection process. In order to collect data such as current stock prices, transaction volumes, and price changes of BIST100 companies through web scraping, web pages that provide such data should be determined. Data obtained from financial sites such as TradingView, Investing.com, Yahoo Finance or financial news sites that include current news and economic situation analyses can be used in analyses about BIST100 companies.

The web scraping process generally consists of the following:

- Creating a Site Map: First, it is determined which pages of the target site will be used to extract information and the HTML structure is analysed. It is defined which HTML elements contain data (for example, , , or <table>).
- Determining HTML Tags and XPath: The HTML elements containing the target data are determined. For example, it can extract certain HTML tags with the methods included in the BeautifulSoup library. Specific elements can be accessed with XPath expressions using Scrapy or Selenium.
- Retrieving and Processing Data: A request is sent to the web page and the necessary information is extracted.
- Cleaning and Saving Data: The retrieved data is usually in raw form. For example, price data can be in string (text) type. Converting the data to numerical form when necessary and saving it to a CSV file or database is done at this stage.

5.2 Data Cleaning and Preprocessing

In our project, data preprocessing is a critical step in order to provide meaningful analysis of the data collected from Twitter. Raw data usually contains noise, which reduces the accuracy of the analysis. The proposed data cleaning and preprocessing techniques provide standards for text mining and sentiment analysis.

5.2.1 Data Cleaning

5.2.1.1 Extracting Usernames, URLs, and Numbers: Raw tweet data often contains usernames (@username), URLs, and meaningless numbers. Extracting usernames and URLs reduces text length and noise, resulting in more accurate sentiment analysis results. This problem can be solved by

identifying and removing usernames and URLs using Python Regex or string manipulation methods.

5.2.1.2 Removal of Special Characters and Punctuation Marks:

Special characters (^, &, \, etc.) and punctuation marks that are frequently encountered in the texts we will obtain from Twitter need to be cleaned since they generally do not carry any meaning in data mining. Removing punctuation marks reduces the data size resulting from unnecessary words. The "string.punctuation" and "re.sub" methods [20] can be used to remove special characters and punctuation marks.

5.2.1.3. Uppercase-Lowercase Conversions:

The machine learning algorithms we will use may perceive uppercase and lowercase letters as different words. To prevent this data conflict, it is necessary to convert all text to lowercase and make the data uniform. This conversion can be easily done in Python using the "str.lower()" function.

5.2.1.4. Converting Emoji and Special Characters to Text:

Emojis, which are frequently used on Twitter as well as on every platform in social media, have meaning in terms of sentiment analysis. Because some users can express their feelings and opinions not only through text but also through tools such as emojis and special text. Converting the meanings of these emojis to text is of great importance in terms of Sentiment Analysis. Emojis can be converted into meaningful words using the Python Demoji library.

5.2.2 Data Preprocessing

5.2.2.1 Removing Stop Words:

Stop words (e.g. "and", "but", "one") are words that do not contribute to sentiment analysis. If these words are removed while preserving the meaning of the text, the data we collect will be free from unnecessary data load. In Python, NLTK or spaCy libraries can be used to identify and remove these words.

5.2.2.2 Tokenization:

Tokenization is defined as the process of separating sentences into meaningful words. Tokenization is very important in ensuring that data is divided into meaningful units for sentiment analysis. Tokenization can be performed using NLTK's "word_tokenize()" function or spaCy in this process.

5.2.2.3 Stemming and Lemmatization:

Stemming and stemming techniques are used to reduce words to their most basic roots. For example, the words "working" and "worked" are converted to the same root ("work"). This makes the data we will use more consistent when processing the data and prevents the model from shifting in meaning. NLTK's "SnowballStemmer" or spaCy's "Lemmatizer" functions can be used for stemming and stemming.

In addition, since the data we will capture will be used in numerical analysis, scaling and normalization are also very important. According to the article Data Preprocessing for Stock Price Prediction Using LSTM and Sentiment Analysis [19], scaling the data within a certain range increases the performance of LSTM models, which are especially strong in time series and text analysis and effective in areas such as stock predictions and sentiment analysis. Tools such as "MinMaxScaler" can help in this process.

As a result, Data Cleaning and Data Preprocessing techniques play a critical role to model the raw data and obtain more consistent results in the project. These techniques that can be used for this project (especially in financial data and social media text analysis) should be adapted with project specific methods to obtain higher quality and meaningful data.

6. Previous Works

There have been previous studies attempting to predict the trends of BIST30[21] or BIST100 indices by examining the stocks of multiple large companies that significantly influence these indices, as well as analyzing news about them through sentiment analysis. In some stocks, an accuracy rate of 86.56% was achieved using FastText and LSTM [22], and here are more examples with different algorithms with pictures[23].

Stocks	Word2Vec	LSTM	Word2Vec+LSTM
AKBNK	84.17	85.39	87.25
ALBRK	87.23	88.05	89.46
GARAN	76.65	78.23	78.98
HALKB	75.96	76.85	77.25
ISCTR	77.16	77.54	79.46
SKBNK	78.57	79.45	79.46
TSKB	74.72	75.93	75.20
VAKBN	78.80	78.64	79.46
YKBNK	77.05	78.12	78.98
Avg.	78.92	79.80	**80.61**

Stocks	FastText	LSTM	FastText+LSTM
AKBNK	89.55	89.23	89.94
ALBRK	89.27	89.00	89.56
GARAN	88.75	88.72	88.15
HALKB	85.43	85.15	85.57
ISCTR	80.27	80.72	80.65
SKBNK	80.78	80.66	81.39
TSKB	81.24	82.80	83.03
VAKBN	87.90	88.24	89.23
YKBNK	88.00	89.37	90.05
Avg.	85.69	85.99	**86.39**

Stocks	Word2Vec	RNN	Word2Vec+RNN
AKBNK	77,25	78,33	80,02
ALBRK	77,10	78,40	80,07
GARAN	78,42	78,77	81,18
HALKB	78,77	78,90	80,11
ISCTR	75,23	75,27	77,00
SKBNK	71,36	72,34	73,87
TSKB	74,30	74,06	75,57
VAKBN	75,50	75,25	76,96
YKBNK	74,80	74,00	75,56
Avg.	75,86	76,15	**77,82**

A comprehensive study has been conducted on sentiment analysis (SA) of financial texts. Sentiment analysis seeks to determine individuals' emotions and attitudes toward specific topics or products using text data. This research extracted sentiment polarity—negative, positive, and neutral—from financial texts through machine learning and deep learning algorithms. The machine learning approach applied Multinomial Naïve Bayes (MNB) and Logistic Regression (LR) classifiers, while deep learning techniques included Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), and Gated Recurrent Unit (GRU) models. MNB and LR achieved good and very good

accuracy rates, respectively, while RNN, LSTM, and GRU demonstrated excellent accuracy rates. The findings suggest that preprocessing steps positively impacted accuracy [24].

In another study For sentiment analysis, FastText was used, a library known for its efficient word vector representations, particularly for morphologically rich languages like Turkish. The FastText model classified Turkish news articles as positive, negative, or neutral based on manually labeled training data of 500 samples [25]. The LSTM-RNN was chosen to capture sequential dependencies in time-series data, making it suitable for stock trend predictions.Experiments on 8 years of data revealed that the model's F1 score, which was around 0.56 without sentiment analysis, increased to 0.65 when sentiment labels were included. The results suggest that the model with sentiment labels better fits actual prices, particularly during periods of high stock price volatility.

Our goal is to improve these accuracy rates by using similar or different models and leveraging the increased amount of data available due to the rise in social media usage. In undertaking sentiment analysis of financial textual data, several challenges are anticipated based on similar studies and prior research in the field. Addressing these issues is essential to enhance model accuracy and efficiency.

First problem we will face is probably Data Quality and Complexity: Financial text data, especially from social media, often includes posts that are either overly lengthy, contain complex or nuanced emotional content, or use non-standard characters and symbols [26]. Such variability in data quality can complicate preprocessing and impact model performance. To mitigate this, rigorous preprocessing steps will be implemented, including data cleaning, tokenization, and feature selection. These steps aim to standardize and simplify the input text data, allowing models to better capture sentiment.

Real-Time Analysis may be the second one, running algorithms on big data such as tweets could be hard to produce some results fast. We can call it as Limited Optimization Techniques: Optimization

techniques, such as swarm or bat optimization, can enhance model accuracy and reduce computation time, but these approaches remain underexplored in financial sentiment analysis [27]. To maximize model performance, our study will implement advanced optimization algorithms during the training phase. These techniques aim to refine the model parameters dynamically, potentially improving sentiment classification without sacrificing speed.

High Volatility in Stock Prices: Stock prices are inherently volatile, and sentiment analysis can sometimes be insufficient to capture rapid market shifts. Studies have shown that sentiment analysis models tend to perform better during stable periods but struggle when there is high volatility [28]. This limitation suggests that while sentiment data adds value, further enhancements in feature engineering and model selection are necessary to account for extreme market changes.

7. Conclusion

This project aims to examine investor behavior and market sentiment by analyzing tweets published on social media about the companies with the highest market capitalization on the Borsa İstanbul (BIST100). A sentiment analysis model based on assets, developed in the field of Turkish Financial Natural Language Processing (NLP), will classify the sentiment of tweets as positive, negative, or neutral. The model will provide a decision support mechanism for investors and market analysts, enabling them to predict market trends more accurately, while also allowing companies to assess public perception and brand strategies.

Using the data collection, processing, and conclusion methods we have gained from previous studies, we hope to bring new solutions to previously encountered problems and create an innovative project for the Turkish stock market in our own work.

References:

[1] Omar Y. Adwan () , Marwan Al-Tawil, Ammar M. Huneiti, Rawan A. Shahin, Abeer A. Abu Zayed, Razan H. Al-Dibsi, "Twitter Sentiment Analysis Approaches: A Survey", pp. 79-93, 2020, https://www.learntechlib.org/p/217980/

[2] Mayur Wankhade1,2 · Annavarapu Chandra Sekhara Rao1,2 · Chaitanya Kulkarni1,2, "A survey on sentiment analysis methods, applications, and challenges", pp. 5731-5780, 2022, https://link.springer.com/article/10.1007/S10462-022-10144-1

[3] Marouane Birjali * , Mohammed Kasri, Abderrahim Beni-Hssane, "A comprehensive survey on sentiment analysis: Approaches, challenges and trends ", pp. 1-26, 2021, https://www.sciencedirect.com/science/article/pii/S095070512100397X

[4] Parastoo Golpour 1 , Majid Ghayour-Mobarhan 2,3, Azadeh Saki 1,* , Habibollah Esmaily 4 , Ali Taghipour 4,5, Mohammad Tajfard 4,6, Hamideh Ghazizadeh 2,7, Mohsen Moohebati 3 and Gordon A. Ferns, "Comparison of Support Vector Machine, Naïve Bayes and Logistic Regression for Assessing the Necessity for Coronary Angiography ", pp. 1-9, 2020 https://www.mdpi.com/1660-4601/17/18/6449

[5] Alexander Ligthart1 · Cagatay Catal2 · Bedir Tekinerdogan1, "Systematic reviews in sentiment analysis: a tertiary study", pp. 4997-5053, 2021, https://link.springer.com/article/10.1007/s10462-021-09973-3

[6] Travis Adams, Andrea Ajello, Diego Silva, Francisco Vazquez-Grande, "More than Words: Twitter Chatter and Financial Market Sentiment ", pp. 1-35, 2023, https://www.federalreserve.gov/econres/feds/files/2023034pap.pdf

[7] Gabriele Ranco1 , Darko Aleksovski2 *, Guido Caldarelli1,3,4, Miha Grčar2 , Igor Mozetič2, "The Effects of Twitter Sentiment on Stock Price Returns", pp. 1-21, 2015, https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0138441

[8] Avar, B., & Ceylan, H., "*Morphological Analysis and Applications in Turkish Natural Language Processing"*,2017

[9] Sak, H., Güngör, T., & Saraçlar, M., "*Turkish Language Resources: A Collection of Tools and Resources for Turkish Natural Language Processing*",2008

[10] Öztürk, N., & Ayvaz, S., "*Sentiment Analysis on Twitter: A Text Mining Approach to the Syrian Refugee Crisis in Turkey*", 4(3),pp. 251-272,2008

[11] Avvad, H., & Ereren, E.,"*Sentiment Analysis in Turkish Tweets Using Different Machine Learning Algorithms"*,pp. 108-120,2024, Artificial Intelligence Theory and Applications » Makale » Sentiment Analysis in Turkish Tweets Using Different Machine Learning Algorithms

[12] Eren, İ., & Yıldırım, Ö., "*Domain-specific Word Embeddings for Turkish Financial Texts*", In Proceedings of the 27th Signal Processing and Communications Applications Conference,2019

[13] Şahin, M., & Gülşen, Ö., "*Analysis of Financial News with Word Embeddings for Turkish Language"*, Journal of Language Technology and Computational Linguistics.2021

[14]Zeynep Banu Özger, Banu Diri, "Türkçe Dokümanlar İçin Kural Tabanlı Varlık İsmi Tanıma". https://dergipark.org.tr/en/pub/tbbmd/issue/22245/238798

[15]Kübra Adalı, A. Cüneyd Tantuğ, "Annotation of Financial Entities Using A Comprehensive Scheme in Turkish". https://ieeexplore.ieee.org/abstract/document/9864782

[16]Zulkarnain, Tsarina Dwi Putri, "Intelligent transportation systems (ITS): A systematic review using a Natural Language Processing (NLP) approach". https://www.cell.com/action/showPdf?pii=S2405-8440%2821%2902718-3

[17] Pierre Lison, Aliaksandr Hubin, Jeremy Barnes, Samia Touileb, "Named Entity Recognition without Labelled Data: A Weak Supervision Approach".

https://arxiv.org/pdf/2004.14723
[18]Abid All Awan, “What is Named Entity Recognition (NER)? Methods, Use Cases, and Challenges”.
https://www.datacamp.com/blog/what-is-named-entity-recognition-ner?utm_source=google&utm_medium=paid_search&utm_campaignid=19589720821&utm_adgroupid=152984011814&utm_device=c&utm_keyword=&utm_matchtype=&utm_network=g&utm_adpostion=&utm_creative=719914245961&utm_targetid=dsa-2222697810918&utm_loc_interest_ms=&utm_loc_physical_ms=9210493&utm_content=DSA~blog~Machine-Learning&utm_campaign=230119_1-sea~dsa~tofu_2-b2c_3-row-p1_4-prc_5-na_6-na_7-le_8-pdsh-go_9-nb-e_10-na_11-na-
[19] Pardeep Kumar, Kanwal Garg, “Data Cleaning of Raw Tweets for Sentiment Analysis”, “IEEE Xplore”: https://ieeexplore.ieee.org/document/9181326
[20] Aditya Singh Rajpurohit, Harshada Mhaske, Pradnya Sangitbabu Gaikwad, “Data Preprocessing for Stock Price Prediction Using LSTM and Sentiment Analysis”, “IEEE Xplore”: https://ieeexplore.ieee.org/document/10112026
[21],[25] Muhlis Sarıyer, Ahmet Akıl, Feyza Nur Bulgurcu, Fatih Emin Öge, Murat Can Ganiz, “Individual Stock Price Prediction by Using KAP and Twitter Sentiments with Machine Learning for BIST30”.
https://ieeexplore.ieee.org/abstract/document/9894172
[22],[23] Zeynep Hilal Kilimci; Ramazan Duvar, “An Efficient Word Embedding and Deep Learning Based Model to Forecast the Direction of Stock Exchange Market Using Twitter and Financial News Sites: A Case of Istanbul Stock Exchange (BIST 100)”.
https://ieeexplore.ieee.org/abstract/document/9218927
[24],[26],[27] Ahmad, H. O., & Umar, S. U. (2023).” Sentiment analysis of financial textual data using machine learning and deep learning models”, pp, 156-157, 2023
https://www.informatica.si/index.php/informatica/article/view/4673
[28]Adnan GUMUS , C. Okan SAKAR, “Stock Market Prediction in Istanbul Stock Exchange by Combining Stock Price Information and Sentiment Analysis”, p, 25, 2021
https://dergipark.org.tr/en/pub/jeps/issue/59877/683952

- OpenAI Chat Gpt: https://chatgpt.com/
- Tweepy: https://docs.tweepy.org/en/stable/
- Beautiful Soup: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
- Scrapy: https://docs.scrapy.org/en/latest/
- Selenium: https://www.selenium.dev/documentation/
- Demoji: https://pypi.org/project/demoji/
- NLTK: https://www.nltk.org/
- spaCy: https://spacy.io/
- IBM: https://www.ibm.com/topics/named-entity-recognition