Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2017.DOI

Sentiment Analysis in Finance: From Transformers Back to eXplainable Lexicons (XLex)

MARYAN RIZINSKI[1, 2], HRISTIJAN PESHOV[2], KOSTADIN MISHEV[2], MILOS JOVANOVIK[2], AND DIMITAR TRAJANOV[2, 1], (Member, IEEE)

[1]Department of Computer Science, Metropolitan College, Boston University, Boston, MA 02215, USA
[2]Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University, 1000 Skopje, North Macedonia

Corresponding author: Maryan Rizinski (e-mail: rizinski@bu.edu).

ABSTRACT Lexicon-based sentiment analysis in finance leverages specialized, manually annotated lexicons created by human experts to effectively extract sentiment from financial texts. Although lexicon-based methods are simple to implement and fast to operate on textual data, they require considerable manual annotation efforts to create, maintain, and update the lexicons. These methods are also considered inferior to the deep learning-based approaches, such as transformer models, which have become dominant in various natural language processing (NLP) tasks due to their remarkable performance. However, their efficacy comes at a cost: these models require extensive data and computational resources for both training and testing. Additionally, they involve significant prediction times, making them unsuitable for real-time production environments or systems with limited processing capabilities. In this paper, we introduce a novel methodology named eXplainable Lexicons (XLex) that combines the advantages of both lexicon-based methods and transformer models. We propose an approach that utilizes transformers and SHapley Additive exPlanations (SHAP) for explainability to automatically learn financial lexicons. Our study presents four main contributions. Firstly, we demonstrate that transformer-aided explainable lexicons can enhance the vocabulary coverage of the benchmark Loughran-McDonald (LM) lexicon. This enhancement leads to a significant reduction in the need for human involvement in the process of annotating, maintaining, and updating the lexicons. Secondly, we show that the resulting lexicon outperforms the standard LM lexicon in sentiment analysis of financial datasets. Thirdly, we illustrate that the lexicon-based approach is significantly more efficient in terms of model speed and size compared to transformers. Lastly, the proposed XLex approach is inherently more interpretable than transformer models. This interpretability is advantageous as lexicon models rely on predefined rules, unlike transformers, which have complex inner workings. The interpretability of the models allows for better understanding and insights into the results of sentiment analysis, making the XLex approach a valuable tool for financial decision-making.

INDEX TERMS Machine learning, natural language processing, text classification, sentiment analysis, finance, lexicons, lexicon learning, transformers, SHAP, explainability

I. INTRODUCTION

The financial industry generates massive amounts of data, from transactional data to news articles and social media posts [1], [2]. This big data poses significant challenges and opportunities for financial institutions as they struggle to extract insights and make sense of the vast amounts of information generated every day. Extracting meaningful trends and actionable knowledge from such an immense quantity of data is so complex and time-consuming that it makes it impossible to perform by any individual actor or stakeholder in the financial market. Thus, automatic approaches for big data analytics are becoming essential in addressing the underlying challenges in finance [3]–[5].

Sentiment analysis can play a crucial role in analyzing, interpreting, and extracting insights from financial big data. Sentiment analysis has become increasingly important in the field of finance and fintech, where it gained popularity in a wide range of applications. One of the main use cases of sentiment analysis in finance is to predict stock market trends [6]–[9]. By analyzing news articles, social media

posts, balance sheets, cash flow statements, and other sources of financial information, sentiment analysis can be used to capture market sentiment, which can help investors in making more informed decisions. For example, if sentiment analysis indicates that the overall market sentiment is negative, investors may choose to sell their stocks to avoid potential losses. Additionally, sentiment analysis can help financial institutions and regulators in monitoring financial markets and investors' behavior to detect potential manipulations, speculations, or fraudulent activities.

Another application of sentiment analysis in finance is to assess the creditworthiness of individuals and companies [10]–[14]. By analyzing social media activity, customer reviews, and other sources of data, sentiment analysis can provide insights into the financial behavior and reputation of borrowers. This can help lenders make more informed decisions about lending and pricing, ultimately reducing the risk of default and improving profitability.

In fintech, sentiment analysis can be used to improve customer experience and engagement [15]–[19]. By analyzing customer feedback, fintech companies can better identify and address customer needs, preferences, and problems. This information can be used to develop personalized products and services that are tailored to customer expectations, thereby resulting in increased customer satisfaction and loyalty. Additionally, sentiment analysis can help fintech companies monitor their brand reputation and detect potential issues before they become widespread, improving overall brand image and customer trust [20]–[22].

Lexicon-based sentiment analysis is a commonly used approach that relies on pre-defined sets of words known as lexicons [23]–[25]. Lexicons are manually annotated by experts in the field and assign sentiment scores to individual words (positive, negative, or neutral). While knowledge extraction using lexicons exhibits a simplistic implementation and fast operation on textual data, considerable manual annotation efforts are required to create, maintain, and update such lexicons. However, even after such laborious annotation, some relevant words may still not be included in the lexicon, potentially leading to reduced sentiment classification accuracy. Furthermore, lexicons tailored for one domain, such as finance, cannot be easily reused in other domains. As indicated in the seminal study by Loughran and McDonald [26], dictionaries developed for other disciplines may misclassify common words in financial texts, highlighting the importance of domain-specific lexicons. There are also generic lexicons used for general-purpose sentiment analysis. However, they are known to be imprecise in various domains, introducing inaccuracies and biases [26].

Another approach to sentiment analysis is by using machine learning (ML) [27]–[30] and deep learning (DL) techniques [25], [31]–[34]. ML/DL techniques are based on sophisticated algorithms that can capture complex linguistic patterns. For example, DL approaches, such as the state-of-the-art (SOTA) transformer models [35], [36], can learn contextual and semantic information as well as capture long-term dependencies in text, making them effective in capturing the nuances of sentiment in text [37]. However, transformer models typically require massive amounts of text data which can be computationally expensive to train and implement [38].

Sentiment extraction from financial texts requires the use of domain-specific language. The traditional approach for sentiment analysis in finance is to use manually annotated lexicons, such as the Loughran-McDonald (LM) lexicon. However, this approach has limitations and manual editing efforts are required to maintain and update such lexicons. While transformers have shown superior performance in sentiment classification tasks, little work has been done to investigate how these approaches can be combined to create better lexicons automatically.

In this paper, we explore the potential of transformers and ML explainability tools such as SHapley Additive exPlanations (SHAP) [39] for automating the creation of lexicons, reducing their maintenance efforts, and expanding their vocabulary coverage. We propose a new methodology for building eXplainable Lexicons (XLex) using pre-trained transformer models and explainable ML tools. The results demonstrate that the proposed methodology leads to the creation of new lexicons that outperform the current state-of-the-art sentiment lexicons in finance.

We compare the newly created explainable lexicon with the LM lexicon (known to outperform general-purpose lexicons in financial contexts) on financial datasets to assess the overall potential and performance of the methodology. Our study demonstrates that generated lexicons can improve the coverage of lexicons annotated by domain experts, potentially leading to faster and more automated data processing pipelines tailored to productive NLP applications while reducing the manual work needed by domain experts. Additionally, we show that our methodology has a generic architecture and can be applied in other areas beyond financial applications.

Dictionary-based sentiment models have their own advantages and disadvantages. To use a dictionary-based sentiment model, the text to be analyzed is first preprocessed to remove stop words, punctuation, and other non-alphanumeric characters. Then, each word in the preprocessed text is matched against the words in the sentiment dictionary and assigned a sentiment score based on its associated sentiment value. The sentiment scores for each word in the text are then aggregated to obtain an overall sentiment score for the text. This approach is relatively simple and straightforward, as it does not require any training or complex modeling. The sentiment dictionary is fixed and does not change during analysis, making it easy to use and implement. Another advantage of dictionary-based sentiment models is their interpretability. Since the sentiment scores assigned to each word in the dictionary are pre-defined, it is easy to understand why a particular text was classified as positive, negative, or neutral. This can be useful for analyzing the sentiment of text in various applications, such as customer feedback

analysis, social media monitoring, and market research. With its inherent interpretability, utilizing lexicons for sentiment analysis can also aid in examining the relationship between the polarity of news articles and the movements of stock prices [40]. In addition, dictionary-based sentiment models have low computational requirements, making them suitable for the real-time analysis of high-volume text sources like social media streams. They can be implemented on low-powered devices, such as mobile phones, which is useful for applications that require quick sentiment analysis results. Despite their benefits, dictionary-based models also exhibit limitations. They may fail to capture the nuances and complexities of natural languages, such as sarcasm and irony, and may exhibit biases towards certain words or sentiment values. Additionally, these models might be ineffective for analyzing text in multiple languages or domains with specialized terminology.

The paper is organized as follows. In Section II, we make a review of the relevant literature. Section III describes the methodology and data processing pipeline for generating explainable lexicons using transformers and SHAP explainability. In Section IV, we explain in detail the constituent phases of the pipeline to create an explainable lexicon based on SHAP that is used to expand the standard LM lexicon. We use this explainable lexicon in Section V to create a new model for sentiment classification. We demonstrate the effectiveness of our approach in Section VI, where we show it outperforms the LM lexicon. Specifically, we provide a discussion assessing the performance of the model in sentiment classification tasks on financial datasets. We use the last Section VII to give concluding remarks and suggest directions for future research.

II. RELATED WORK

The process of lexicon-based sentiment analysis has traditionally focused on creating lexicons by manually labeling the sentiment of the words included in the lexicons. While such lexicons are of high quality, they require laborious curation and domain expertise [23]. Thus, lexicons created for one domain use specialized vocabulary and may not be suitable or directly applicable to other domains. As the polarity of words may vary across disciplines, domain dependence in sentiment analysis has been emphasized by researchers in the field [41]–[43]. [26] showed that word lists curated for other domains misclassify common words in financial texts. For example, the word "liability" is considered neutral in finance, but it usually conveys a negative polarity in general-purpose applications, making the reuse difficult in specialized lexicons. In their seminal study [26], the authors created an expertly annotated lexicon, called Loughran-McDonald (LM) lexicon, to more accurately capture sentiments in financial texts. Other dictionaries used in finance include General Inquirer (GI) [44], Harvard IV-4 (HIV4), and Diction, but their performance is known to be inferior compared to the LM lexicon in sentiment classification tasks in finance.

Given these drawbacks, statistical methods have been proposed for automatic lexicon learning. For example, [45] showed that emoticons or hashtags in tweet messages can be used to avoid manual lexicon annotation and to significantly improve lexicon coverage while effectively leveraging the abundance of training data. While [45] relied on calculating pointwise mutual information (PMI) between words and emoticons, [46] uses a simple neural network to train lexicons that improve the accuracy of predicting emoticons in tweets.

The study in [47] takes a different approach that proves to be beneficial; it recognizes that supervised solutions can be expensive due to the need to perform burdensome labeling of data. The process of data labeling can be not only difficult but also expensive while still having the disadvantage of producing a limited lexicon coverage. Therefore, as its main contribution, [47] proved that semantic relationships between words can be effectively used for lexicon expansion, contrary to what has been widely assumed in the semantic analysis literature. Their method uses word embeddings to expand lexicons in the following way: it adds new words whose sentiment values are inferred from "close" word vectors that are already present in the lexicon. Surprisingly, the experimental analysis in [47] showed that the unsupervised method proposed by the authors is as competitive as state-of-the-art supervised solutions such as transformers (BERT) without having to rely on any training (labeled) data.

Automatic lexicon building has been studied in several papers in the literature. For instance, certain approaches have shown that taking negation into account improves the performance of financial sentiment lexicons on various sentiment classification tasks [48]. Adapting lexicons that depend on word context is studied in [49]; this work captures the context of words as they appear in tweet messages and uses it to update their prior sentiment accordingly. The methodology in [49] showed improvement in lexicon performance due to the sentiment adaptation to the underlying context. Earlier works explored various directions such as automatic lexicon expansion for domain-oriented sentiment analysis [50], lexicon generation from a massive collection of web resources [51], construction of polarity-tagged corpus from HTML documents [52], etc.

Inducing domain-specific sentiment lexicons from small seed words and domain-specific corpora is studied in [53], where it is shown that this approach outperforms methods that rely on hand-curated resources. The approach is validated by showing that it accurately captures the sentiment mood of important economic topics of interest, such as data from the Beige Book of the U.S. Federal Reserve Board (FED) and data from the Economic Bulletin of the European Central Bank (ECB). Combining word embeddings with semantic similarity metrics between words and lexicon vocabulary is shown to better extract subjective sentiment information from lexicons [54]. This paper emphasizes that the capability to automatically infer embedding models leads to higher vocabulary coverage. The experiments in [54] also demonstrate that lexicon words largely determine the performance of the resulting sentiment analysis, meaning

that similar lexicons (i.e., with similar vocabulary) result in similar performance.

The comparable performance among lexicons containing similar vocabulary is one of our main reasons to explore the potential of transformers to automatically learn and expand known lexicons in an explainable way. The power of NLP transformers to accurately extract sentiment from financial texts is presented in [37], where the authors perform a comprehensive analysis with more than one hundred experiments to prove the capabilities of transformers, and in particular, how their word embeddings outperform lexicon-based knowledge extraction approaches or statistical methods.

Due to the complexity of machine learning (ML) techniques, especially deep learning models, the outputs of the models are hard to visualize, explain and interpret. In recent years, this gives rise to a vast literature on ML model explainability. A state-of-the-art technique for explainability is considered SHAP (SHapley Additive exPlanations), which uses Shapley values from game theory to explain the output of ML models [39].

The potential of SHAP is explored in different use cases. SHAP has been recently proven beneficial for diagnosing the explainability of text classification models based on Convolutional Neural Networks (CNNs) [55]. When combined with CNNs, SHAP is effective in explaining local feature importance while also taking advantage of CNN's potential to reduce the high feature dimensionality of NLP tasks. CNN is known to outperform other ML algorithms for text classification, which implies that the SHAP-based analysis of CNN in [55] can be potentially carried out to explain any text classification tasks. The increased interest in SHAP has also been extended to the financial domain, where SHAP values are used for topics such as interpreting financial time series [56] and financial data of bankrupt companies [57]. A comprehensive study has been performed in [58] to evaluate SHAP in the context of ethically responsible ML in finance. The SHAP method has been adapted for explaining SOTA transformer language models such as BERT with the goal of improving the visualizations of the generated explanations [59].

Extracting sentiment from news text, social media, and blogs has gained increasing interest in economics and finance. The study in [60] proposes a fine-grained aspect-based sentiment analysis to identify sentiment associated with specific topics of interest in each sentence of a document. Business news texts are used to compile a comprehensive domain-specific lexicon in [61]. A hybrid lexicon that combines corpus-based and dictionary-based methods with statistical and semantic measures is proposed in [62], showing that sentiments extracted from a large dataset of financial tweets exhibit a correlation with market trends.

Sentiment analysis of news articles using lexicons has been performed on the BBC news dataset in [24]. The work outlines the two main lexicon approaches to sentiment analysis, namely dictionary-based and corpus-based methods, but it does not involve machine learning techniques. The study in [63] recognized that focusing entirely on machine learning by ignoring the knowledge encoded in sentiment lexicons may not be optimal. Thus, the authors presented a method that incorporates domain-specific lexicons as prior knowledge into algorithms such as SVM and showed that it could improve the accuracy of sentiment analysis tasks.

While acknowledging the advantages of deep learning methods, the results in [64] showed that lexicon-based methods are to be preferred for uses cases with low-resource languages or limited computational resources at the expense of slightly lower performance. The authors performed a comparative study between the BERT Base Italian XXL language model and the NooJ-based lexical system with Sentix and SentIta lexicons, thereby validating the idea of using lexicons in use cases with scarce datasets. The paper used SHAP to perform qualitative analysis between the two approaches, but SHAP was not used to improve the coverage of existing lexicons. To the best of our knowledge, SHAP has still not been explored for the purpose of automatic lexicon generation.

III. THE XLex METHODOLOGY

The construction of a lexicon for sentiment analysis comprises several consecutive stages, each involving suitable word processing. To facilitate sentiment analysis, the lexicon must incorporate words from both positive and negative polarities. In this section, we delineate the steps involved in generating the positive and negative sentiment sets, which will subsequently be merged to form an explainable lexicon.

The architecture of the data processing pipeline is depicted in Figure 1. The individual components of the pipeline are elaborated in detail in the following sections of the paper.

The development of a lexicon which we will use to perform sentiment analysis, takes place in several successive stages, each involving appropriate word processing. To facilitate sentiment analysis, the lexicon must incorporate words from both positive and negative polarities. In this section, we will explain each of the steps taken to create the two sentiment sets (positive and negative), which will then be combined into an explainable lexicon.

The architecture of the data processing pipeline is given in Figure 1. Each of the constituent elements of the pipeline will be explained in detail in subsequent parts of the paper.

A. AN INITIAL MODEL FOR SENTIMENT ANALYSIS

To create the explainable lexicon, we use the RoBERTa-based pre-trained transformer model studied in [65]. The model is selected as it shows superior performance in various finance-related sentiment classification experiments, achieving an accuracy of 94%. Given its performance capabilities, we use the model to classify the sentiment of input sentences taken from financial-related textual datasets. The datasets are given in Table 12 where they are denoted as "Source" datasets. The results of the model are then evaluated and explained using SHAP. Employing SHAP to explain the model's decisions enables the extraction of words belonging

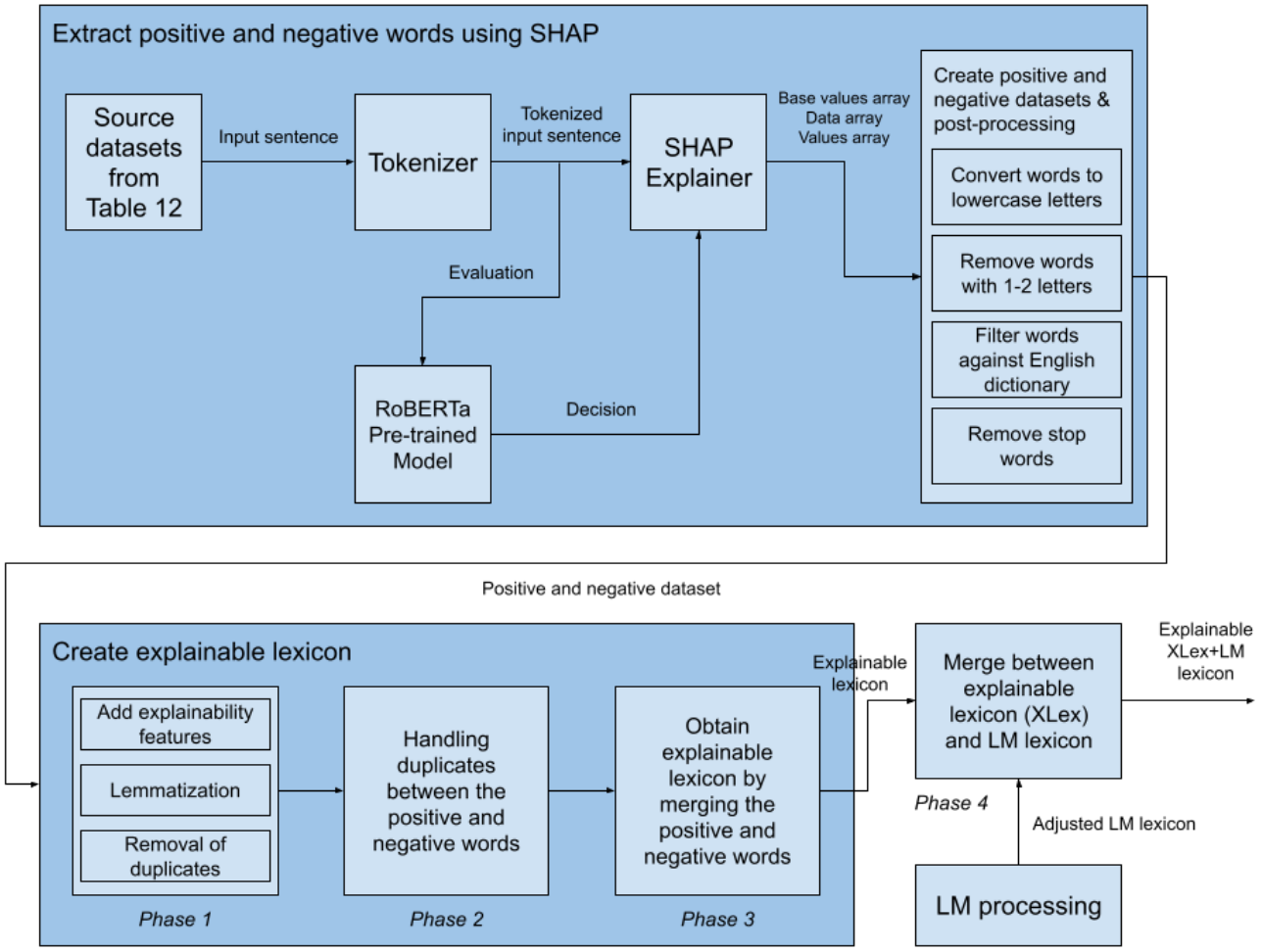

FIGURE 1: Architecture of the data processing pipeline used to obtain the explainable lexicon (XLex). Details on the merge between explainable and LM lexicons and the LM processing are given in Figures 2 and 3. Adding explainability features and handling duplicates between the positive and negative words are shown on Figures 4 and 5 respectively.

to specific sentiment groups, i.e., either positive or negative. This approach is discussed in detail in Subsection III-B.

This initial RoBERTa-based model is fine-tuned on a dataset that is a combination of two datasets: Financial PhraseBank [66] and SemEval-2017-Task5 [67]. These two constituent datasets are composed of financial headlines extracted from two different sources. The sentences in the Financial PhraseBank corpus are selected using random sampling from English news on all listed companies in the OMX Helsinki stock index. The sampling is performed to ensure that the selected sentences represent both small and large companies, different industries as well as different news sources. The dataset contains 4846 sentences annotated with three labels, i.e., three polarities: positive, negative, and neutral. On the other hand, SemEval-2017-Task5 is the dataset used for the "Fine-Grained Sentiment Analysis" problem posed by Task 5 of the SemEval 2017 competition. It consists of approximately 1200 news headlines related to large companies operating worldwide. The headlines are extracted from various internet sources, including Yahoo Finance. The sentiment score of each sentence in the dataset is labeled with a real number ranging from -1 to 1. A summary of the statistics of the two datasets is given in Table 1.

As illustrated in Table 1, the sentences are not evenly distributed across the different types of polarity. There is an imbalance between the number of positive and negative sentences in both datasets. The number of neutral sentences also differs drastically when compared to the number of positive or negative sentences. To address the problem, balancing is performed by extracting 1093 positive and 1093 negative sentences, which are then merged into one dataset. This dataset is used for training and evaluation of the model that we take from [65]. The sentences in the dataset are shuffled and divided into 80% training set and 20% test set. The training and test sets contain 1748 and 438 sentences, respectively. Both the training and test sets are balanced, i.e., they contain the same number of positive and negative sentences. The statistics of the resulting dataset are shown in

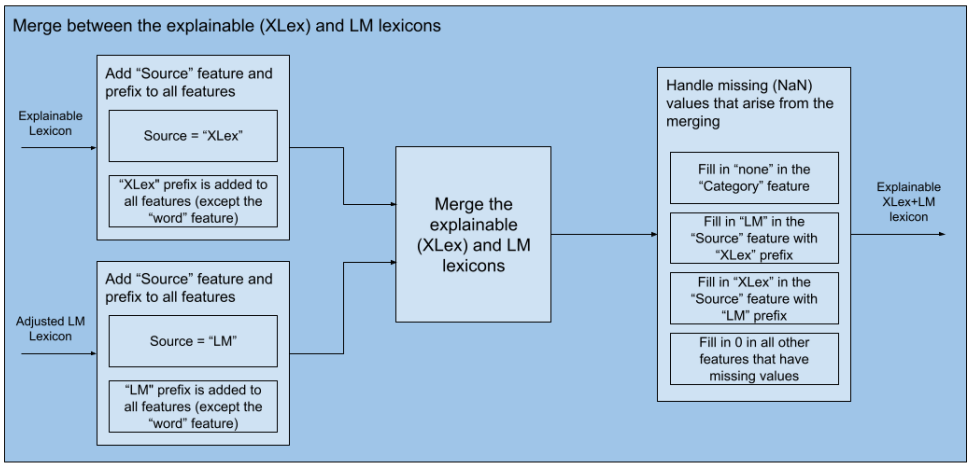

FIGURE 2: Merge between the explainable and LM lexicons.

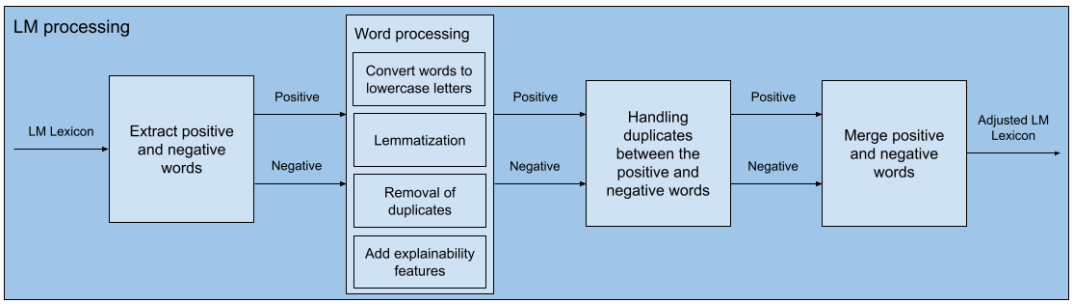

FIGURE 3: LM processing.

Table 2.

B. EXTRACTING WORDS AND THEIR ANALYSIS WITH SHAP

The first step in creating the lexicon involves extracting words from financial sentences and labeling them as positive or negative. For this purpose, we use the previously introduced model together with a tokenizer. The model classifies the sentiment of the input sentences, while the tokenizer deals with tokenization, i.e. dividing the sentences into component words. The model and tokenizer are then passed to the SHAP explainer, which generates explanations for the model decisions.

SHAP is considered a state-of-the-art technique for ML model explainability [68]. Its approach uses Shapley values from game theory to explain the output of ML models [69]. Game theory is characterized by two elements: a game and players. From the perspective of the SHAP explainer, the game consists of reproducing the results of the model being explained (in our case, that is, the NLP model for sentiment analysis), while the players are the features (the financial statement, i.e., its constituent words) that are passed as input to the model. SHAP evaluates the contribution of each feature to the model predictions and assigns each feature an importance value, called a SHAP value. SHAP values are calculated for each feature across all samples of the dataset to assess the contribution of individual features to the model's output [39]. It is important to note that SHAP explains the predictions locally, meaning that the contributions of the features (words) on the model prediction are related to a specific sample in the dataset. A different sample can yield other values for the features' contributions. However, due to the additive nature of SHAP values, it is also possible to aggregate them, allowing us to calculate global values for the overall contribution of the features across all samples.

To evaluate the features' contributions on the model prediction for a given sample, SHAP creates a copy of the model for each combination of the input features. Each of these

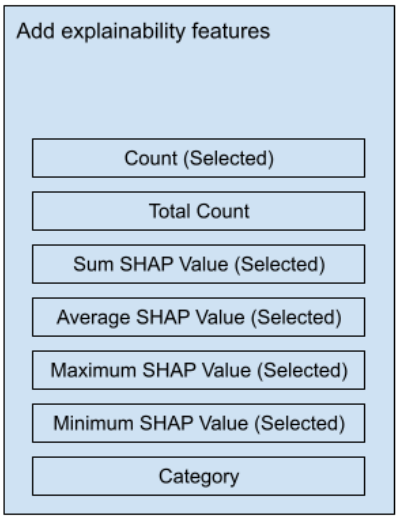

FIGURE 4: Add explainability features based on SHAP in the explainable and LM lexicons. For the LM lexicon, all features except "Category" are assigned the value of 1 as their default value.

TABLE 1: Polarity distribution of sentences in the Financial PhraseBank and SemEval-2017-Task5 datasets.

Sentiment	Dataset	
	Financial PhraseBank	SemEval2017-Task5
Neutral	2879	38
Negative	604	451
Positive	1363	654
Total	4846	1143

TABLE 2: Statistics of the train and test sets used for fine-tuning the initial RoBERTa-based model.

Polarity	Financial PhraseBank & SemEval2017-Task5		
	Train set	Test set	Total
Negative	874	219	1093
Positive	874	219	1093
Total	1748	438	2186

models is the same, the only difference is the combination of features passed to the model. One of these combinations is the one where none of the features is passed to the model. In that case, the model results in a mean value for the prediction; the value is obtained by averaging the labels of the dataset on which the model was trained. This value is called a *base value*. The base value is the value that would be predicted if no features are known for the model's current output [39]. In this way, by adding a certain feature to the input, the SHAP explainer can record the changes to the predicted value and can measure the contribution of that feature. Each of the features can increase or decrease the predicted value. Finally, to obtain the value predicted by the model being explained (when all features are present), SHAP aggregates the contribution values (which can be positive or negative) for each feature and superposes the result to the base value (the prediction with no input features provided). Using this process, SHAP explains the contribution (importance) of each feature in a given sample. In other words, SHAP measures the difference in the predicted value caused by the presence or by absence of a feature. The additive nature of the aggregation is where the name SHAP comes from, namely Shapley Additive exPlanations.

The input parameter passed to the SHAP explainer is a sentence. The NLP model evaluates the sentiment of the sentence by making a sentiment classification decision, while SHAP provides an explanation for the decision. The explanation of the SHAP explainer returns three arrays: *base values array*, *data array*, and *values array*. The base values array contains two numeric values: a base value for the positive class and a base value for the negative class. These base values represent the values that would be predicted for a particular sentence if no input features are known. In this case, it is the mean value of the labels for each of the classes obtained across all instances (samples) on which the model was trained. The data array contains the tokens (the constituent words), which are obtained by applying the tokenizer to the input sentence. The elements of the values array represent the weights, that is, the contribution of each of the words (tokens) in the calculation of the sentiment of the sentence. The weights in the values array are real numbers ranging from -1 to 1. The weights represent the importance of a particular word (token) and its contribution to the final value predicted by the sentiment classification model. The data array and the values array have the same number of elements.

As mentioned earlier, the weights are additive which allows them to be superposed. By adding the weights to the base value, the explainer arrives at the value predicted by the sentiment model. Visually, this superposition is represented using diagrams where the calculated weights "push" the base value to the "right" or to the "left", causing the model to increase or decrease its predicted value. By doing so, it is possible to explain how the model arrived at a given decision and how different parts of a sentence contributed to the model's output. Specifically, in terms of sentiment analysis with SHAP, this helps understand why a given NLP model classified a sentence as positive or negative and how each of the constituent words of the sentence contributed to that classification decision.

A visual example is given in Figure 6. The figure shows that positive importance values, marked red, "push" the base value to the "right" (increasing the model's predicted value), while negative weights, marked blue, "push" the base value to the "left" (reducing the model's predicted value). The example is visualized from the perspective of the positive class, meaning that each "push" to the "right" increases the probability of predicting a positive sentiment for the given sentence. Each "push" to the "left" decreases this probability, that is, it increases the probability of predicting a negative sentiment for the sentence. This view makes it possible to understand how the sentiment classification model arrived at the predicted decision and how different parts of the sentence contribute to the model's outcome. Using these

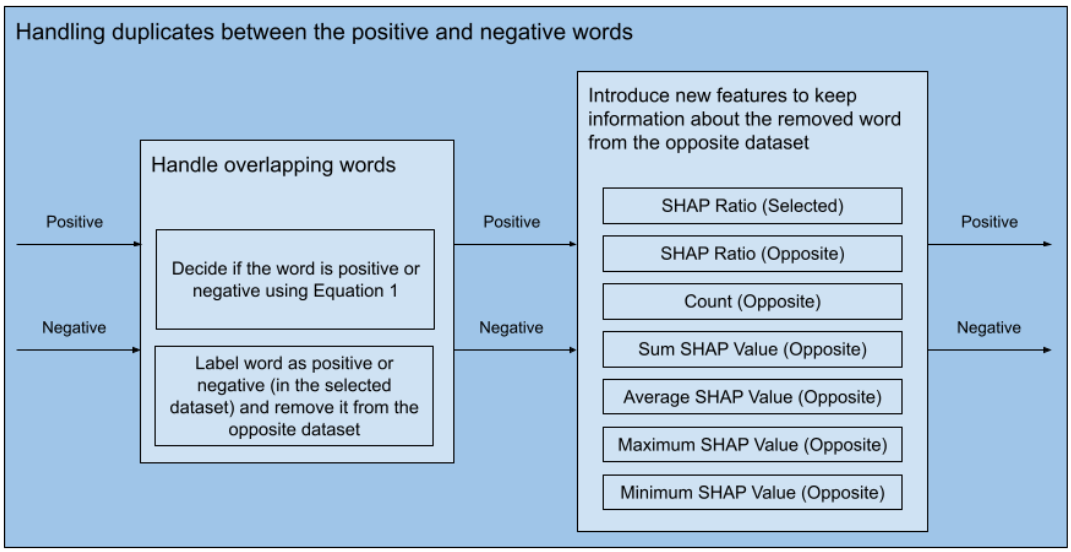

FIGURE 5: Handling duplicates between the positive and negative words for the explainable and LM lexicons. For the LM lexicon, all features marked as "Opposite" are assigned the value of 0 as their default value. The feature "Total Count" can be derived from "Count (Selected)" and "Count (Opposite)".

preliminaries about the SHAP explainer, we will next create two sets containing positive and negative words as explained in Subsection III-C.

C. CREATING A POSITIVE AND NEGATIVE DATASET AND THEIR POSTPROCESSING

The sentiment classification in the previous subsection is performed on datasets containing financial sentences. These datasets are denoted as source datasets in Table 12. Using SHAP, each of the words in a sentence is marked as positive or negative in the given context. The decision to label a particular word as positive or negative depends on whether it contributes with a positive or negative weight to the final decision of the model. As a result, two new datasets are generated. One dataset contains all words across all sentences that contribute to the positive sentiment of each sentence (we refer to the words and dataset as "positive" words and "positive" dataset, respectively), while the other dataset contains all words across all sentences that contribute to the negative sentiment ("negative" words, "negative" set). In addition to the words themselves, these datasets store a few additional parameters for each word, such as the mean value of the weights (importance values) obtained by the SHAP explainer for all of the word appearances, the sum of these values as well as their maximum and minimum values. The datasets also store the total number (count) of appearances of each of the words. All numerical entries in the datasets are represented by their absolute value.

After creating the positive and negative datasets, we perform post-processing to filter the extracted words. The goal is to keep only the words that are valid and have meaning. The word post-processing process is explained as follows.

The post-processing begins by transforming all words into lowercase letters. Then, all entries consisting of one or two letters are removed since they are of little sentiment utility to the datasets. These entries are typically fragments of words that are obtained due to the limitations of the tokenizer. The RoBERTa tokenizer is limited by the size and coverage of the vocabulary that is used to train the tokenizer. This leads to incorrect or imprecise tokenization of certain words that are either not sufficiently represented in the training vocabulary or are not represented at all. Consequently, these words are not represented accurately as they are rather divided into parts based on more common entries that the tokenizer discovers in its vocabulary. Thus, entries with one or two letters are deemed unnecessary and are removed due to their insufficient contribution to the sentiment analysis.

To obtain valid and useful words, we apply another filter to the datasets. Using a dictionary of English words, we remove all words that are not contained in the English dictionary. This is done to address the limitation of the tokenizer and also to provide a dataset containing only valid words. The last step in the post-processing removes auxiliary words that do not carry meaning in the sentence, such as adverbs, prepositions, pronouns, and conjunctions (stop words).

These preliminary steps and the data stored for each word are necessary to develop an explainable lexicon which will be shown in Section IV.

IV. XLex DEVELOPMENT PROCESS

In the previous section, we demonstrated the use of a transformer model for sentiment analysis in combination with SHAP to process finance-related sentences, which resulted in the creation of two datasets. One dataset contains all words with positive sentiment in a given context (positive dataset), while the other contains all words with negative sentiment

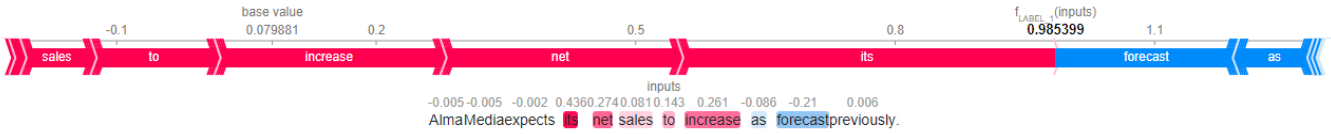

FIGURE 6: An example of using SHAP for evaluating the word contributions to the sentiment of a sentence

in a given context (negative dataset). These two datasets are used to create an explainable lexicon as will be shown later on. We will evaluate the performance of the explainable lexicon employing the model proposed in the subsequent Section V. In addition, the explainable lexicon will be merged with the Loughran-McDonald (LM) lexicon due to its popularity in lexicon-based sentiment analysis tasks in finance. The purpose of merging the two lexicons is twofold. First, we assess the sentiment classification performance of the resulting (merged) lexicon. Secondly, we investigate the potential for improving the manually annotated LM lexicon by expanding it with words obtained in an explainable way. The evaluation of the combined lexicon will again be done with the model proposed in Section V. The results related to the evaluation of the lexicons are presented in Section VI.

This section explains in detail the methodology for generating the explainable lexicon as well as the process of merging it with the LM lexicon. The methodology encompasses four phases, as shown in Figure 1.

A. PHASE 1: LEMMATIZATION AND REMOVAL OF DUPLICATE WORDS WITHIN THE POSITIVE AND NEGATIVE DATASETS

In this phase, we lemmatize the words in the positive and negative datasets obtained in Section III. As a result of the lemmatization, each of the words is replaced by its lemma, that is, by its basic form. The goal is to bring different forms of a certain word to their common lemma, thereby avoiding different interpretations of the same word. However, this causes duplicate words to appear in the datasets. After their lemmatization, different forms of a word (which until that moment are uniquely represented) can have the same lemma. The purpose of Phase 1 is to make the datasets consistent by removing duplicate words. Avoiding duplicates will also result in a single source of information related to a particular word.

Each of the duplicates may result in different values for the number of appearances, the average SHAP value, the sum SHAP value, as well as for the maximum and minimum SHAP values. Thus, the goal is to merge the duplicates so that each word is characterized by a single (unique) value for each of these features. The removal of duplicates is performed separately in each of the two datasets (the positive and the negative set). As will be shown, there are words that are labeled both as positive and negative, i.e., words that are present in both datasets. Dealing with these duplicates between the two datasets is done in Phase 2.

To calculate the unique values for the features of a particular word, it is necessary to aggregate the values of the features across all duplicates. The method for aggregating the duplicates by each of the features (columns) is shown in Table 3. The aggregation function represents how the values of all duplicates are combined (aggregated) for a certain feature. From the table, it can be seen that the feature indicating the number of appearances of the word is obtained as a summation of the number of appearances for each of the duplicates. The reason for this is that each of the duplicates represents the same word after lemmatization, so the number of occurrences of that word will be represented as the sum of all the occurrences of the duplicates. The same approach is applied to calculate the total (sum) SHAP value. After lemmatization, all duplicates of a word have the same form, so the sum SHAP value of all occurrences is the sum of the values of this feature across all duplicates. The maximum value is represented by an aggregation function that takes the maximum along this column for all duplicates. The maximum SHAP value of all duplicates is a suitable representative of the maximum SHAP value of that word. The minimum SHAP value is handled similarly. To obtain the minimum SHAP value for the word, it is necessary to aggregate it with a function that calculates the minimum SHAP value from all duplicates. As can be seen from Table 3, no aggregation is performed for the feature “Average SHAP value” because it is obtained by dividing the sum SHAP value by the number of occurrences of the word.

Table 4 illustrates an example of merging duplicate words in the positive dataset and getting unique values for the features. The example presents three different words that result in the same lemma after performing lemmatization, demonstrating how to duplicate words are handled. In order to have only one instance of the word “acquire” in the dataset, it is necessary to merge these three instances into one. This is done as per the definition of the aggregation functions given in Table 3. In the example, the sum is calculated based on the number of occurrences of all duplicates (9 + 4 + 5 = 18), which results in a total of 18 occurrences of the word “acquire”. The sum of the sum SHAP values across all duplicates (3.05+1.4+0.88 = 5.33) represents a sum SHAP value of 5.33 for the word. To get the average SHAP value, it is necessary to divide the sum SHAP value by the number of occurrences (5.33 ÷ 18 = 0.3), which gives an average SHAP value of 0.3 for the word “acquire”. The maximum SHAP value for the word is 0.6, while the minimum SHAP value is 0.02.

TABLE 3: Aggregation functions to handle duplicates across the numerical features of the sentiment dataset. No aggregation is performed for the average SHAP value as it is obtained by dividing the sum SHAP value by the total number of word appearances. Another feature that is not aggregated is Category since it is a categorical variable.

	Features				
	Number of apperances	Sum SHAP value	Average SHAP value	Maximum SHAP value	Minimum SHAP value
Aggregation function	Sum	Sum	N/A	Max	Min

TABLE 4: An example for aggregating duplicates in the positive dataset. Duplicates are handled similarly in the negative dataset.

Duplicates		Features				
Original word	Lemma	Number of appearances	Total SHAP value	Average SHAP value	Maximum SHAP value	Minimum SHAP value
acquire	acquire	9	3.05	0.34	0.6	0.05
acquired	acquire	4	1.4	0.35	0.5	0.23
acquiring	acquire	5	0.88	0.18	0.43	0.02
Single instance after aggregation (acquire)		18	5.33	0.3	0.6	0.02

The method demonstrated in this phase is applied to each of the two datasets separately. The example above shows how this process is performed for one word in the positive dataset, but the same procedure is used for all other duplicate words in that dataset, as well as for all duplicate words in the negative dataset. This is indicated with the elements "Lemmatization" and "Removal of duplicates" in Figure 1.

B. PHASE 2: HANDLING OF DUPLICATE WORDS BETWEEN DATASETS

A particular word can be present in both the positive and negative datasets, leading to word overlaps between the datasets. Given our goal to generate a lexicon as a combination of the two datasets, each word should be represented by a single instance in the resulting lexicon. To overcome the overlaps, we use the following approach. If an overlapping word has a higher sum SHAP value in the positive dataset ($SHAP^{pos}_{sum}$) when compared to the negative one ($SHAP^{neg}_{sum}$), then the word is labeled as positive. Similarly, if $SHAP^{neg}_{sum}$ is higher than or equal to $SHAP^{pos}_{sum}$, then the word is labeled as negative. The decision criteria are shown in Equation 1:

$$selected\,dataset = \begin{cases} \text{positive}, & SHAP^{pos}_{sum} > SHAP^{neg}_{sum} \\ \text{negative}, & otherwise \end{cases} \tag{1}$$

If a certain word is labeled as positive or negative (in the selected dataset), it is removed from the opposite dataset. To keep the information about the word removed from the opposite dataset, new columns are introduced in the datasets. The new columns are given in Table 5 under "Features added in Phase 2". A complete representation of the words in the two datasets, including their features from both polarities, is achieved by adding these columns. Using Equation 1 as a decision criterion and keeping information about the word removed from the opposite dataset is shown in Figure 5.

Table 5 shows the features added in Phase 2 in addition to the existing features of the datasets. The table also consolidates brief explanations for each of the features. It should be noted that the label "opposite" represents the set that was not selected during the decision, in accordance with Equation 1. Thus, if Equation 1 decides that an overlapping word belongs to the positive dataset, in that case, the "opposite" dataset is the negative dataset. This word is removed from the negative dataset, and all its values from the negative dataset are placed in the positive dataset, in the corresponding columns marked as "opposite". Similarly, if the decision criteria decide that the word belongs to the negative dataset, in that case "opposite" represents the positive dataset. This word is removed from the positive dataset, and all its values from the positive dataset are placed in the negative set in the corresponding columns marked "opposite".

If a word is decided to belong to the positive set, then the SHAP ratio ($SHAP_{ratio}$) is calculated as the ratio between the average SHAP value of the word from the positive dataset and the sum of the average SHAP values of the word from the positive and negative datasets. This is shown in Equation 2:

$$SHAP_{ratio} = \frac{SHAP^{pos}_{avg}}{SHAP^{pos}_{avg} + SHAP^{neg}_{avg}} \tag{2}$$

The opposite value of the SHAP ratio is expressed as the ratio between the average SHAP value of the word from the opposite dataset (in this case, it is the negative dataset) and the sum of the average SHAP values of the word from the positive and negative sets. This is shown in Equation 3.

$$SHAP^{opp}_{ratio} = \frac{SHAP^{neg}_{avg}}{SHAP^{pos}_{avg} + SHAP^{neg}_{avg}} = 1 - SHAP_{ratio} \tag{3}$$

Similar steps are taken if the word is decided to belong to the negative dataset. The only difference is that $SHAP_{ratio}$ is calculated based on the average SHAP value of the negative dataset ($SHAP^{neg}_{avg}$), while $SHAP^{opp}_{ratio}$ is calculated based on the average SHAP value of the positive set ($SHAP^{pos}_{avg}$).

Table 6 shows an illustrative example with the word "option" that appears in both datasets (positive and negative).

TABLE 5: Features of the words in the lexicons.

Feature name	Feature notation	Feature description
Word feature		
Word	$word$	Word that appears in the lexicon
Initial features		
Count (Selected)	$count$	Number of appearances of the word
Sum SHAP Value (Selected)	$SHAP_{sum}$	Sum SHAP value of the word
Average SHAP Value (Selected)	$SHAP_{avg}$	Average SHAP value of the word
Maximum SHAP Value (Selected)	$SHAP_{max}$	Maximum SHAP value of the word
Minimum SHAP Value (Selected)	$SHAP_{min}$	Minimum SHAP value of the word
Features added in Phase 2		
Total Count	$count_{total}$	Total number of appearances of the word (in the two sentiments)
Count (Opposite)	$count_{opp}$	Total number of appearances in the opposite sentiment
Sum SHAP Value (Opposite)	$SHAP_{sum}^{opp}$	Sum SHAP value of the word in the opposite sentiment
Average SHAP Value (Opposite)	$SHAP_{avg}^{opp}$	Average SHAP value of the word in the opposite sentiment
Maximum SHAP Value (Opposite)	$SHAP_{max}^{opp}$	Maximum SHAP value of the word in the opposite sentiment
Minimum SHAP Value (Opposite)	$SHAP_{min}^{opp}$	Minimum SHAP value of the word in the opposite sentiment
SHAP Ratio (Selected)	$SHAP_{ratio}$	Ratio between $SHAP_{avg}$ and the sum of $SHAP_{avg}$ and $SHAP_{avg}^{opp}$
SHAP Ratio (Opposite)	$SHAP_{ratio}^{opp}$	Ratio between $SHAP_{avg}^{opp}$ and the sum of $SHAP_{avg}$ and $SHAP_{avg}^{opp}$
Features added in Phase 3		
Category	$category$	Category of the word (positive or negative)
Features added in Phase 4		
Source	src	Source lexicon from which the word originates from

As can be seen, this word has a sum SHAP value in the positive and negative dataset of $SHAP_{sum}^{pos} = 0.39$ and $SHAP_{sum}^{neg} = 0.023$, respectively. Given that $SHAP_{sum}^{pos} > SHAP_{sum}^{neg}$, it is decided that the word belongs to the positive dataset and is removed from the negative dataset. Before removing the word from the negative dataset, the values of its features from the negative dataset are added to the positive dataset in the corresponding columns labeled as "opposite". The values added in the "opposite" columns are given as follows: $count_{opp} = 7$, $SHAP_{sum}^{opp} = 0.023$, $SHAP_{avg}^{opp} = 0.0033$, $SHAP_{max}^{opp} = 0.009$, $SHAP_{min}^{opp} = 0.0001$. The SHAP ratio of the word in the two datasets is calculated as $SHAP_{ratio} = \frac{0.026}{0.026+0.0033} = 0.887$; $SHAP_{ratio}^{opp} = \frac{0.0033}{0.026+0.0033} = 0.113$. All other features of the word in the selected (positive) dataset remain unchanged.

If a word appears only in one of the datasets, a zero value is assigned to each of the features labeled as "opposite" because that word does not appear in the opposite sentiment. Also, according to Equation 2, $SHAP_{ratio}$ evaluates to 1 since $SHAP_{avg}^{opp}$ for the corresponding word is 0.

C. PHASE 3: MERGING THE POSITIVE AND NEGATIVE DATASETS

In this phase, the two datasets, positive and negative, are merged into a single dataset. The feature "Category" is important for the merging. Possible values for this feature are "positive" and "negative", depending on whether the word is in the positive or negative dataset. All words from the positive dataset have "positive" as the value for this feature, while all words from the negative dataset have the value "negative". The purpose of the "Category" feature is to delineate positive words from negative words in the resulting explainable lexicon. Using this feature, the two datasets are merged by simply adding all the data points (i.e., all words together with all their features) from the negative set to the positive one. An excerpt of the explainable lexicon after the merge is shown in Table 7. [1] This finalizes the creation of the explainable lexicon containing words that are automatically extracted with the help of transformers and SHAP. A distinctive property of this lexicon is the usage of SHAP values, especially $SHAP_{avg}$, which will be later on used to perform sentiment analysis. As will be shown by the results in Section VI, $SHAP_{avg}$ is a good indicator of the sentiment of a particular word. In addition to this feature, $SHAP_{ratio}$ and $count$ are also introduced as parameters that will be used in sentiment analysis when determining the polarity of a particular sentence. This is explained in detail in Section V.

Our aim is to use the explainable lexicon to improve and extend the LM lexicon. To compare their performance, these two lexicons are combined into a final lexicon which for the remainder of the paper will be interchangeably referred to as the combined lexicon or XLex+LM lexicon. We will use the combined lexicon to perform sentiment analysis on financial sentences, thereby evaluating the possible improvement of

[1] For this and the remaining diagrams related to lexicons, only part of the dataset features are shown so that the diagrams can fit within the page limits.

TABLE 6: An example with the word "option" and its features in the dataset.

Word	option				
Dataset	Features (columns)				
	$count$	$SHAP_{sum}$	$SHAP_{avg}$	$SHAP_{max}$	$SHAP_{min}$
Positive	15	0.39	0.026	0.2	0.07
Negative	7	0.023	0.0033	0.009	0.0001

TABLE 7: An excerpt of selected features of the explainable lexicon.

Word	Count (Selected)	Total	Count (Opposite)	Category	Average SHAP Value (Selected)	Average SHAP Value (Opposite)	Sum SHAP Value (Selected)	Sum SHAP Value (Opposite)
new	426	577	151	positive	0.090629	0.021480	38.608165	3.243527
amp	311	568	257	negative	0.032435	0.037868	10.087175	9.732118
world	267	513	246	positive	0.046991	0.027468	12.546591	6.757057
year	384	461	77	negative	0.050818	0.032912	19.514242	2.534230
china	213	426	213	positive	0.042231	0.032894	8.995204	7.006360
group	297	398	101	positive	0.079659	0.026396	23.658838	2.666006
company	291	387	96	positive	0.076470	0.030276	22.252625	2.906498
energy	327	357	30	positive	0.109173	0.019223	35.699520	0.576700
bank	202	351	149	positive	0.069276	0.036797	13.993703	5.482755
power	312	328	16	positive	0.136059	0.019770	42.450354	0.316315
state	258	301	43	negative	0.039640	0.011132	10.227108	0.478668
million	168	285	117	negative	0.048877	0.050262	8.211418	5.880613
use	147	284	137	positive	0.042997	0.025001	6.320554	3.425115
country	240	280	40	negative	0.037619	0.023715	9.028585	0.948618
market	177	273	96	positive	0.048187	0.043074	8.529055	4.135104
trump	231	269	38	negative	0.086255	0.022430	19.924996	0.852340
time	192	264	72	negative	0.045261	0.026059	8.690184	1.876225
apple	224	255	31	positive	0.112486	0.022463	25.196845	0.696366
chinese	139	248	109	negative	0.035228	0.026792	4.896641	2.920374
global	173	240	67	positive	0.078297	0.040759	13.545395	2.730843
billion	168	233	65	negative	0.055175	0.048924	9.269427	3.180079
ban	149	228	79	negative	0.066532	0.032665	9.913209	2.580523
day	180	223	43	negative	0.064980	0.026559	11.696429	1.142023

the combined lexicon over the plain vanilla LM lexicon. The results obtained by analyzing the combined lexicon will be shown and discussed in Section VI. In the next and last phase, Phase 4, we explain the process of combining the explainable and LM lexicons into the combined lexicon.

D. PHASE 4: MERGING WITH THE LOUGHRAN-MCDONALD DICTIONARY

In this last phase, we combine the explainable lexicon with the LM lexicon. However, before this can be done, it is necessary that the words in the LM lexicon undergo similar processing as in the case of the explainable lexicon so that the LM words obtain the same set of features. The processing of words of the LM lexicon is given in Figure 3 and is explained as follows.

1) Processing of the LM Lexicon

While the Loughran-McDonald lexicon consists of seven sentiment datasets, only its positive and negative components (datasets) are of interest to the combined (XLex+LM) lexicon. Similarly to the datasets used to create the explainable lexicon, the words from the LM datasets are first transformed into lowercase letters and then lemmatized. Duplicate words are obtained due to lemmatization. These datasets consist only of words without any other additional features (columns), so there is no need to aggregate the duplicates for a particular word. Instead, all duplicates are removed, leaving only one instance of the word in the datasets. To be able to combine this lexicon with the explainable lexicon, it is necessary to ensure they have the same features. Thus, all features from the explainable lexicon (shown in Table 5) are added to the LM datasets.

As a first step, the initial features and the features introduced in Phase 2 are added to each of the LM datasets. These newly added features (except for those labeled as *"opposite"*) are assigned a value of 1 as their main (default) value. Since these words do not contain values for the corresponding features, it is necessary to assign them a specific value. The value 1 is chosen as the main value to indicate if the word is contained in the given dataset. While 1 is a high value to be assigned to $SHAP_{avg}$, this default value assignment is compensated with the model coefficients that are introduced in Section V. On the other hand, those features labeled as opposite are assigned a value of 0 since there are no words from one dataset that overlap with the other dataset. This assignment of values is a consequence of the fact that the words originating from the LM datasets are not obtained in an explainable way using SHAP; thus, they do not have the characteristics shown in Table 5.

In addition, the feature "category" is added to all the words from the LM datasets. For the words from the positive and negative LM dataset, this column is filled with the value

"positive" or "negative" respectively. As was the case with the datasets from the explainable lexicon, the purpose of the "category" feature in the LM datasets is to be able to identify the origin of a given word in the merged LM lexicon, i.e., whether the word originates from the positive or negative LM dataset. After this, the two LM datasets are merged into a single consolidated dataset by simply adding the data points (i.e., the words together with all their features) from the negative to the positive LM dataset. This concludes the processing of the LM lexicon. In the next subsection, the LM lexicon will be merged with the explainable lexicon to arrive at the combined (XLex+LM) lexicon. A visual overview of the LM lexicon after merging the positive and negative LM datasets, along with some of the added features, is shown in Table 8.

2) Obtaining the XLex+LM Lexicon by Merging the XLex and LM Lexicons

As a final step, we merge the explainable lexicon (XLex) created in Phase 3 with the LM lexicon. We make two changes in the lexicons before merging them. We introduce a new feature (column) called *src* ("source") as shown in Table 5. Since two different lexicons will be merged into one, the purpose of this feature is to indicate the origin of a certain word in the merged lexicon, i.e., whether the word originates from the explainable or LM lexicon. The feature is filled in with the value "XLex" and "LM" if the word originates from the explainable and LM lexicon, respectively. The src feature allows flexibility in selecting the lexicon that is used by the sentiment analysis model in the evaluation process. Thus, it is possible to select the explainable lexicon, the LM lexicon, or the combined (XLex+LM) lexicon.

We also add a prefix to all features in the lexicons (i.e., all features indicated in Table 5). The only exception is the column that contains the word itself ("word" column) since that column is used to merge the two lexicons. Adding the prefix is done with the same purpose, namely to have the flexibility to select a lexicon for the sentiment analysis model. Selecting a certain lexicon means taking into account only its words and features in the sentiment analysis and not the words and features of the other lexicon. Before merging the lexicons, they have the same names for the features, so to distinguish these features in the combined lexicon, it is necessary to name them differently. We add prefixes "XLex" and "LM" to denote the columns from the explainable and LM lexicon, respectively. This is shown in Tables A.1-A.2 in the Appendix A. The prefix "XLex" stands for "eXplainable Lexicon" and indicates that the lexicon is created using explainability tools. The prefix "LM" is an abbreviation for the Loughran-McDonald lexicon, indicating that these features are related to the LM lexicon. With these two changes, it is possible to completely extract the explainable or LM lexicon from the combined lexicon.

After merging the lexicons, all words will appear with one instance in the combined dataset, including words that appear in both lexicons. The features of a given word in the combined lexicon will contain the feature values of both the explainable and LM lexicons for that word. This is shown in Table A.3. If a particular word does not appear in both lexicons, it will also be represented by a single instance in the combined lexicon, but its instance will be populated only with the features of that word from the lexicon in which it exists, not the features from the other lexicon. This is shown in Table A.4. Words of this type do not appear in both lexicons, therefore, for the lexicon in which they do not appear, there is no value that can be assigned to them. This is the reason why after merging the lexicons, there are words with missing feature values for certain columns, as indicated with "NaN" (missing value) in Table A.4. For different columns, the missing values are handled differently. The feature "XLex Category" is filled with the value "none" because that word does not appear in the explainable lexicon. Similarly, the feature "LM Category" is filled in with the value "none" because the word is not present in the LM lexicon. If the column "XLex Source" has the value "NaN", then it means that the word is from the LM lexicon, so the column "LM Source" has the value "LM". To indicate that the word does not appear in the explainable lexicon, the "NaN" value of the "XLex Source" column is replaced by the "LM" value. Similarly, if the "LM Source" column has the value "NaN", then it means that the word is from the explainable lexicon, so the "XLex Source" column has the value "XLex". To indicate that the word is not contained in the LM lexicon, the value "NaN" of the column "LM Source" is replaced by the value "XLex". All other columns that contain "NaN" values (for the corresponding lexicon in which the given word does not appear) are assigned the value of 0. If a word does not appear in a given lexicon, the value for all its features is 0. Figure 2 summarizes the handling of missing ("NaN") values that arise due to the merging of the two lexicons.

After handling the invalid feature values, an excerpt of the lexicon's content is shown in Table 9. Comparing Table A.4 and Table 9 can reveal the effect of replacing invalid values for certain columns. A normalized version of the combined lexicon is then created. To obtain the normalized lexicon, the values of each of the numerical features are modified according to Equation 4:

$$v_{norm} = \frac{v(f)}{max(f)} \tag{4}$$

where $v(f)$ represents the value of a feature for a given word, while $max(f)$ is the maximum value of that feature across all words. This step concludes the creation of the combined XLex+LM lexicon, which can now be used as a basis for performing sentiment analysis.

In the next section, we define a model for sentiment analysis based on the combined lexicon.

TABLE 8: An excerpt of selected features of the LM lexicon.

Word	Count (Selected)	Total	Count (Opposite)	Category	Average SHAP Value (Selected)	Average SHAP Value (Opposite)	Sum SHAP Value (Selected)	Sum SHAP Value (Opposite)
surpasses	1	1	0	positive	1	0	1	0
transparency	1	1	0	positive	1	0	1	0
tremendous	1	1	0	positive	1	0	1	0
tremendously	1	1	0	positive	1	0	1	0
unmatched	1	1	0	positive	1	0	1	0
unparalleled	1	1	0	positive	1	0	1	0
unsurpassed	1	1	0	positive	1	0	1	0
upturn	1	1	0	positive	1	0	1	0
valuable	1	1	0	positive	1	0	1	0
versatile	1	1	0	positive	1	0	1	0
versatility	1	1	0	positive	1	0	1	0
vibrancy	1	1	0	positive	1	0	1	0
vibrant	1	1	0	positive	1	0	1	0
win	1	1	0	positive	1	0	1	0
winner	1	1	0	positive	1	0	1	0
worthy	1	1	0	positive	1	0	1	0
abandon	1	1	0	negative	1	0	1	0
abandonment	1	1	0	negative	1	0	1	0
abdicate	1	1	0	negative	1	0	1	0
abdicates	1	1	0	negative	1	0	1	0
abdication	1	1	0	negative	1	0	1	0
aberrant	1	1	0	negative	1	0	1	0
aberration	1	1	0	negative	1	0	1	0

TABLE 9: The combined lexicon after handling invalid values.

Word	XLex Count (Selected)	XLex Total	XLex Count (Opposite)	XLex Average SHAP Value (Selected)	XLex Source	LM Average SHAP Value (Selected)	LM Category	LM Sum SHAP Value (Opposite)	LM Source	LM Max SHAP Value (Opposite)
abide	1.0	1.0	0.0	0.003838	XLex	0	none	0	XLex	0
abo	1.0	1.0	0.0	0.000976	XLex	0	none	0	XLex	0
aboard	2.0	2.0	0.0	0.122640	XLex	0	none	0	XLex	0
abolition	1.0	1.0	0.0	0.006430	XLex	0	none	0	XLex	0
abroad	3.0	3.0	0.0	0.039073	XLex	0	none	0	XLex	0
writeoff	0.0	0.0	0.0	0.000000	LM	1	negative	0	LM	0
writeoffs	0.0	0.0	0.0	0.000000	LM	1	negative	0	LM	0
wrongful	0.0	0.0	0.0	0.000000	LM	1	negative	0	LM	0
wrongfully	0.0	0.0	0.0	0.000000	LM	1	negative	0	LM	0
wrongly	0.0	0.0	0.0	0.000000	LM	1	negative	0	LM	0

V. MODEL FOR SENTIMENT ANALYSIS BASED ON EXPLAINABLE LEXICONS

In this section, we develop a model for sentiment analysis. The model is designed to make lexicon-based decisions, namely using the combined XLex+LM lexicon. To perform sentiment classification, the model can also use the explainable or LM lexicon as input since both can be extracted from the combined lexicon. To determine the sentiment of sentences, it is necessary to pass the following input parameters to the model: the combined lexicon, the lexicon's features that will be used to make decisions about the sentiment of the sentences, as well as the source of the words, that is, which of the lexicons will be used in the analysis (explainable, LM or combined lexicon). There are three features used for decision-making purposes: $SHAP_{avg}$, $SHAP_{ratio}$, and $count$. Each of these characteristics can make the decision individually, but they can also be used together in any combination. The details of how these decision features are used together are explained later in this section.

After defining the model and its input parameters, we use the model to perform sentiment classification of financial sentences. The datasets used in the process of sentiment classification, and the corresponding results are outlined in Section VI. We use the evaluation method of our model. The input parameters passed to the method are the sentences to be evaluated, the actual labels (sentiment) of those sentences, as well as 4 or 2 coefficients, depending on whether the combined lexicon or any of the constituent lexicons is used individually. The purpose of these coefficients is to control how much each of the lexicons will contribute to the decision as well as how much importance will be given to the selected category relative to the opposite category. The coefficients will be discussed in more detail in this section, where the sentiment calculation is expressed mathematically.

We now explain how to calculate the sentiment of a certain sentence using the sentiment analysis model, relying on the combined lexicon. The explanation applies to one sentence, but the same process is applied to every sentence in the dataset. To determine the sentiment of a particular sentence, it is first split into its component words using the RoBERTa tokenizer from Section III. Then, every word is transformed into lowercase letters and lemmatized. All words in the

combined lexicon are lemmatized, so in order to follow an identical approach, we also lemmatize the words from the evaluation sentences. Each of the sentences is represented as a set of words w_i, $1 \leq i \leq n$:

$$sentence = \{w_1, w_2, ..., w_n\} \tag{5}$$

We calculate the sentiment value of every word w_i in a given sentence. Before calculating this sentiment value, it is necessary to calculate a cumulative value for each of the lexicons selected in the sentiment analysis (explainable and LM) and for each of the word categories (positive and negative). The term "cumulative value" refers to the sum of the values of the word's features. For a specific word and for a specific lexicon, the cumulative value for the positive category is calculated as per Equation 6:

$$v_c{}^{pos}(w_i) = \sum_{i=1}^{n} v^{pos}(x_i) \tag{6}$$

where x_i, $1 \leq i \leq n$, are the features selected to make the decision in the sentiment analysis. These features are the same for each of the selected lexicons and for each of the word categories. $v^{pos}(x_i)$ is the value of the feature x_i of the positive word category. The sum of all these features represents the cumulative value of a given word in the selected lexicon as per the positive category. As previously mentioned, these decision-making features can be at most three ($SHAP_{avg}$, $SHAP_{ratio}$, and $count$) and at least one. At the same time, it is possible to use any other combination of them. Similarly to the positive word category, the cumulative value of a given word in the selected lexicon with respect to the negative category is calculated as per Equation 7:

$$v_c{}^{neg}(w_i) = (-1) \sum_{i=1}^{n} v^{neg}(x_i) \tag{7}$$

where x_i, $1 \leq i \leq n$, are the selected features used to make sentiment decisions while $v^{neg}(x_i)$ is the value of the feature x_i in the negative word category. The sum across all the selected features represents the cumulative value of a given word in the negative category for the selected lexicon. As can be seen in Equation 7, the sum is multiplied by -1, ensuring that the cumulative value for the negative word category is always negative. Initially, all feature values are positive in each of the two lexicons as well as in the combined lexicon, i.e., given by their absolute values. Thus, it is necessary to multiply the cumulative value by -1 for the negative category. As will be pointed out later in this subsection, this facilitates the calculation of the sentiment value of the analyzed word. It should be noted that if a certain word does not exist in one of the categories or in one of the lexicons, the cumulative value evaluates to 0 according to Equations 6-7.

The sentiment value of a given word can be calculated after determining the cumulative value for each lexicon and each category. If the combined XLex+LM lexicon is chosen for performing the sentiment analysis, the sentiment value of the word is obtained using Equation 8:

$$\begin{aligned} v_{sent}(w_i) = c_{xlp} * v_c{}^{xl}(w_i) + c_{xlo} * v_c{}^{xl,opp}(w_i) + \\ + c_{lmp} * v_c{}^{lm}(w_i) + c_{lmo} * v_c{}^{lm,opp}(w_i) \end{aligned} \tag{8}$$

The variables used in Equation 8 are summarized and explained in Table 10. The coefficients (parameters) in Equation 8 are introduced to control the contribution of each lexicon and each category on the sentiment classification decision. As will be explained in Section VI, the parameters can be fine-tuned to investigate which values lead to improved sentiment classification performance.

If only the explainable lexicon is passed to the sentiment analysis model when determining the sentiment of the sentences, the sentiment value of a given word is calculated using Equation 9:

$$v_{sent}(w_i) = c_{xlp} * v_c{}^{xl}(w_i) + c_{xlo} * v_c{}^{xl,opp}(w_i) \tag{9}$$

As can be seen, only two coefficients are used in this equation instead of four since only one of the lexicons is selected.

On the other hand, if only the LM lexicon is selected as an input to the sentiment analysis model, the sentiment value of a given word is calculated by Equation 10:

$$v_{sent}(w_i) = c_{lmp} * v_c{}^{lm}(w_i) + c_{lmo} * v_c{}^{lm,opp}(w_i) \tag{10}$$

Equation 10 also has only two parameters instead of four since only one of the lexicons is selected.

After calculating the sentiment value of every word in a sentence using the above equations, the sentiment value of the sentence is evaluated as the sum of the sentiment value of each of the constituent words. This is given in Equation 11:

$$v_{sent}(sentence) = \sum_{i=1}^{n} v_{sent}(w_i) \tag{11}$$

where $sentence$ is represented as a set of words (Equation 5). In this way, the sentiment value of a certain sentence is calculated. Next, we determine the polarity of a sentence, i.e., whether it is positive, negative, or neutral. To calculate the sentiment polarity s_{pol} of a sentence from its sentiment value, we check whether the sentiment value is positive, negative, or equal to 0 as follows:

$$s_{pol}(sentence) = \begin{cases} positive & : v_{sent}(sentence) > 0 \\ negative & : v_{sent}(sentence) < 0 \\ neutral & : otherwise \end{cases} \tag{12}$$

The sentiment model uses Equation 12 to calculate the sentiment of each sentence that is subject to sentiment analysis. After calculating the sentiment of each sentence, we evaluate the sentiment classification performance of the model. The evaluation is performed using the predicted and actual sentiments of each of the sentences. For this purpose, we use standard classification metrics such as accuracy, precision, recall, F1 score, and MCC. We also generate a classification report and confusion matrix.

TABLE 10: Explanations of the variables used in the equations for calculating the sentiment value of a given word (Equations 8-10).

Variable	Description
Primary category	The category selected as the primary category in Phase 2 (positive or negative)
Opposite category	The category that was not selected as the primary category in Phase 2 (positive or negative)
c_{xlp}	Coefficient of influence on the cumulative value of the primary category in relation to the explainable lexicon
${v_c}^{xl}$	Cumulative value of the primary category in relation to the explainable lexicon
c_{xlo}	Coefficient of influence on the cumulative value of the opposite category in relation to the explainable lexicon
${v_c}^{xl,opp}$	Cumulative value of the opposite category in relation to the explainable lexicon
c_{lmp}	Coefficient of influence on the cumulative value of the primary category in relation to the LM lexicon
${v_c}^{lm}$	Cumulative value of the primary category in relation to the LM lexicon
c_{lmo}	Coefficient of influence on the cumulative value of the opposite category in relation to the LM lexicon
${v_c}^{lm,opp}$	Cumulative value of the opposite category in relation to the LM lexicon

Equations 11-12 show why it is necessary to involve multiplication by -1 in Equation 7. The sentiment value of a certain sentence is the sum of the sentiment values of the constituent words of that sentence. The sentiment of the sentence depends on whether its sentiment value is positive or negative. Thus, it is important to ensure that a positive word leads to a positive sentiment value while a negative word leads to a negative sentiment value. This is achieved by the equations for calculating the cumulative value (Equations 6-7).

The methodology explained in this section completes the entire process – from automatic word extraction, word classification, and postprocessing to creating an explainable lexicon with SHAP, combining it with the manually annotated LM lexicon, and finally creating a model that will classify the sentiment of finance-related sentences. The results obtained by applying this model to different datasets of financial sentences are shown in the next section.

VI. RESULTS AND DISCUSSION

We present results obtained by the model introduced in the previous section using the combined XLex+LM lexicon.

A. USED DATASETS

Tables 11-12 present the datasets used to build the explainable lexicons as well as the datasets on which these lexicons are evaluated. Table 11 summarizes these datasets by giving their descriptions, while Table 12 contains summary statistics about the datasets. Each of the datasets consists of financial sentences, where each sentence is labeled with its sentiment polarity. It should be noted that these datasets do not contain the sentences that were used to train the initial model given in Section III with the goal of avoiding bias in the experiments[2].

[2]The code and datasets for the experiments can be found at: https://github.com/hristijanpeshov/SHAP-Explainable-Lexicon-Model

The label "Source" in the "Purpose" column in Table 12 denotes that the corresponding dataset is used to extract words with SHAP and to generate an explainable lexicon. Details about generating the explainable lexicons from these datasets are shown as follows.

B. SUMMARIZATION OF THE GENERATED LEXICONS

We generate three different explainable lexicons. The words in the lexicons are generated from three different sources. The datasets that are the sources of the lexicons are marked as "Source" in the "Purpose" column of Table 12. Each of the lexicons is created using the method described in Sections III-IV. The purpose of using different sources is to verify the ability of the method presented in this paper to successfully generate explainable lexicons under different conditions (given that different sources exhibit varied data). We also want to evaluate the effectiveness of the methodology to automatically label the words with the appropriate sentiments. Summary data of the explainable lexicons is shown in Table 13, which also gives information about the LM lexicon.

Each of the explainable lexicons from Table 13 is combined with the LM lexicon, and the resulting lexicons are used in the process of evaluating the model performance. The results of the analysis are shown in the next subsection.

C. RESULTS FROM THE SENTIMENT ANALYSIS

The model takes (uses) two parameters that can be fine-tuned: decision coefficients and the features that will be used to make sentiment decisions. We perform a grid search to find the optimal values of the model parameters that result in the highest accuracy when performing sentiment analysis.

Although it is possible to use all three decision features ($SHAP_{avg}$, $SHAP_{ratio}$, and $count$), we conducted a grid search to identify the most effective combination. Our results

TABLE 11: Descriptions of the datasets that are used in the process of obtaining results from the sentiment analysis.

Dataset	Description
sentfin	Sentfin: Aspect-based dataset of financial news
fiqa_labeled_df	Fiqa: Aspect-based dataset of financial sentences
dev_df	Test set for the sentiment analysis from Table 2
financial_phrase_bank	Financial PhraseBank: Manually annotated financial sentences about companies listed on the OMX Helsinki stock index
fpb_fiqa	Financial PhraseBank + Fiqa
nasdaq	Financial news about companies listed on the NASDAQ index
fiqa_fpb_sentfin_neutral	All neutral sentences from Fiqa, Financial PhraseBank and Sentfin

TABLE 12: Statistics of the datasets used in the process of generating results from the sentiment analysis giving the number of positive, negative, and neutral sentences as well as indicating the purpose of the datasets (used as a source or for evaluation).

Label	Total number of sentences	Positive	Negative	Neutral	Purpose
fiqa_labeled_df	882	602	280	0	Evaluation
dev_df	398	200	198	0	Evaluation
fpb_fiqa	1542	1236	306	0	Evaluation
financial_phrase_bank	885	774	111	0	Evaluation
financial_phrase_bank	2960	89	0	2871	Source
nasdaq	9202	3067	5903	232	Source
fiqa_fpb_sentfin_neutral	6086	0	0	6086	Source

TABLE 13: Statistics of the lexicons on which sentiment analysis is performed.

Lexicon	Total number of words	Positive	Negative
fiqa_fpb_sentfin_neutral	3313	1635	1678
nasdaq	5751	2537	3214
financial_phrase_bank	2729	1342	1387
Loughran–McDonald	1731	246	1485

revealed that $SHAP_{avg}$ has the dominant impact on accuracy, and we, therefore, selected it as the primary decision feature. Then we performed a second grid search by using only the standard (without normalization) explainable lexicons in order to find the optimal values of the coefficients c_{xlp} and c_{xlo}. We choose $0.1, 0.3, 0.5, 0.7$, and 0.9 as possible values for these coefficients to distinguish between different levels of impact (0.1 denotes weak impact, while 0.9 denotes strong impact). By applying permutations with repetition, all permutations of the values for the coefficients are obtained. There are five possible values that can be assigned to two coefficients, thus, the total number of permutations is $5^2 = 25$. These permutations are combined with each of the three explainable lexicons and each of the four evaluation datasets. The only exception is the financial phrase bank explainable lexicon and the financial phrase bank evaluation dataset. We do not evaluate this combination to avoid biased results. As a result, we arrive at a total of 275 models that are created for the purpose of grid search (25 permutations $\times$ 2 explainable lexicons $\times$ 4 evaluation datasets $+$ 25 permutations $\times$ 1 explainable lexicon $\times$ 3 evaluation datasets). For each of the models in the grid search set of models, we are using the $SHAP_{avg}$ as the decision maker.

To select the optimal values for the coefficients c_{xlp} and c_{xlo}, we extract the top half (13 out of 25) results in terms of accuracy score for each of the explainable lexicons and each of the evaluation datasets. Following that, we proceed by measuring the number of occurrences of each coefficient pair. The results show that the following (c_{xlp}, c_{xlo}) pairs $(0.7, 0.5)$, $(0.9, 0.7)$, $(0.9, 0.5)$, $(0.1, 0.1)$, $(0.7, 0.7)$ and $(0.5, 0.5)$ are the most frequent (each pair appears with 11 occurrences) and are the ones that lead to the highest accuracy. Since all these pairs have the same number of occurrences, either of them can be used in the subsequent experiments. We select the pair $(c_{xlp}, c_{xlo}) = (0.7, 0.5)$.

To find the optimal values for the coefficients c_{lmp} and c_{lmo}, we need to repeat this procedure, but now utilizing the LM lexicon instead of the explainable lexicons. We also use the standard (without normalization) version of the LM lexicon. For each evaluation dataset, the results show that the obtained accuracy is the same irrespective of the coefficients' values. Instead of searching for the most favorable values for the c_{lmp} and c_{lmo} coefficients using only the LM lexicon, we search for the coefficient values using the combined lexicon. As the combined lexicon requires all four coefficients, we are fixing the c_{xlp} and c_{xlo} with the values determined in the previous grid search, and then we determine the optimal values of the c_{lmp} and c_{lmo} coefficients. Again, the possible values for the coefficients are 0.1, 0.3, 0.5, 0.7, and 0.9, and the decision maker is $SHAP_{avg}$. Similarly to the previous grid search, we created 275 models, but this time utilizing the standard (without normalization) combined XLex+LM lexicons instead of just the explainable lexicons. Same as previously, we extract the top half results (13 out of 25) in terms of accuracy score for each of the explainable lexicons and each of the evaluation datasets and measure the number

TABLE 14: Results obtained using the models for sentiment analysis. The columns "LM", "XLex", "XLex+LM" suggest that the model determined the sentiment of the sentences using only the LM lexicon, only the explainable lexicon or the combined lexicon, respectively. Similarly, the columns "LM on LM", "XLex on LM", "(XLex+LM) on LM" indicate that the model determined the sentiment using only the LM lexicon, only the explainable lexicon or the combined lexicon, respectively, on those expressions from the evaluation set for which the model from the LM column had an answer. Within each of the two methods (the evaluation on the whole dataset or the evaluation on the part of the dataset), the approach with the highest accuracy among the three approaches is represented in bold.

Source of the lexicon	Normalized	Evaluation set	Accuracy on whole dataset			Accuracy only on the part of the dataset for which the LM-based model has an answer		
			LM	XLex	XLex+ LM	LM on LM	XLex on LM	(XLex+LM) on LM
nasdaq	Yes	dev_df	0.369	0.621	**0.704**	0.808	0.648	**0.83**
nasdaq	Yes	financial_phrase_bank	0.31	0.764	**0.78**	0.755	0.749	**0.788**
nasdaq	Yes	fiqa_labeled_df	0.27	0.615	**0.642**	0.698	0.633	**0.704**
nasdaq	Yes	fpb_fiqa	0.285	0.699	**0.717**	0.727	0.699	**0.744**
nasdaq	No	dev_df	0.369	0.704	**0.764**	0.808	0.692	**0.824**
nasdaq	No	financial_phrase_bank	0.31	**0.846**	**0.846**	0.755	**0.791**	**0.791**
nasdaq	No	fiqa_labeled_df	0.27	0.673	**0.681**	0.698	0.68	**0.701**
nasdaq	No	fpb_fiqa	0.285	0.765	**0.768**	0.727	0.737	**0.744**
fiqa_fpb_sentfin_neutral	Yes	dev_df	0.369	0.698	**0.744**	0.808	0.731	**0.83**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.31	**0.782**	0.779	0.755	**0.785**	0.777
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.27	0.65	**0.662**	0.698	0.683	**0.716**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.285	0.715	**0.718**	0.727	0.736	**0.744**
fiqa_fpb_sentfin_neutral	No	dev_df	0.369	0.711	**0.761**	0.808	0.72	**0.83**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.31	**0.809**	0.802	0.755	**0.81**	0.793
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.27	0.668	**0.675**	0.698	0.692	**0.71**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.285	0.735	**0.735**	0.727	0.749	**0.75**
financial_phrase_bank	Yes	dev_df	0.369	0.661	**0.714**	0.808	0.698	**0.813**
financial_phrase_bank	Yes	fiqa_labeled_df	0.27	0.554	**0.595**	0.698	0.595	**0.701**
financial_phrase_bank	Yes	fpb_fiqa	0.285	0.646	**0.675**	0.727	0.656	**0.731**
financial_phrase_bank	No	dev_df	0.369	0.709	**0.761**	0.808	0.709	**0.824**
financial_phrase_bank	No	fiqa_labeled_df	0.27	0.626	**0.642**	0.698	0.672	**0.713**
financial_phrase_bank	No	fpb_fiqa	0.285	0.695	**0.708**	0.727	0.709	**0.74**

of occurrences of the coefficient pairs. The results show that the value pairs leading to the highest accuracy and which are most frequent for the (c_{lmp}, c_{lmo}) coefficients are $(0.3, 0.1)$, $(0.3, 0.3)$, $(0.3, 0.5)$, $(0.3, 0.7)$, $(0.3, 0.9)$, all with 11 occurrences. Since all of the pairs have the same number of occurrences, we can choose any of the pairs. We select the pair $(c_{lmp}, c_{lmo}) = (0.3, 0.5)$.

Once the optimal model parameters are selected, we proceed by generating the results. For this purpose, we use the combined lexicons from Table 13 that are also available in their normalized form. Each of these lexicons serves as a basis for performing sentiment analysis using the model proposed in Section V. Each of the created models is evaluated on all evaluation sets in Table 12 (these are the sets containing the label "Evaluation" in the "Purpose" column). Only the model which uses the Financial PhraseBank dataset as a source for the combined lexicon is not evaluated on that same dataset in order to avoid model bias. The results are shown in Table 14.

D. DISCUSSION

Table 14 reveals that the model achieves overall best results in sentiment analysis when the combined lexicon (XLex+LM) is used as a basis for the analysis. The same applies when the explainable lexicon (XLex) is used as a basis. The highest accuracy of the model based on the combined (XLex+LM) lexicon is 0.846; if sentiment classification is performed using only the explainable lexicon under the same conditions (i.e., the same source of the lexicon and the same evaluation set), the obtained accuracy also evaluates to 0.846. The accuracy evaluates to 0.31 if only the LM lexicon is taken as a basis for sentiment classification. The reason for this result is due to the insufficient word coverage of the LM lexicon since the LM lexicon does not contain the words that make up a large part of the sentences of the evaluation set. Therefore, those expressions are unanswered. As can be seen from Table 15, the sentiment analysis performed with the LM lexicon leads to a large number of unanswered sentences for each of the datasets. The percentage of unanswered sentences is about 60% for each dataset. These unanswered expressions are considered wrongly answered, leading to very low accuracy of the LM lexicon. On the other hand, Table 15 shows that there are almost no unanswered sentences when using explainable lexicons combined with the LM lexicon. Hence, the results show that explainable lexicons are advantageous over manually annotated lexicons as they achieve larger vocabulary coverage and higher accuracy in sentiment analysis. Explainable lexicons are able to achieve larger vocabulary

coverage because they can automatically extract words and classify them using explainable ML models. Consequently, the combined lexicon also leads to a larger vocabulary coverage.

To compare the explainable and LM lexicons in terms of accuracy, we proceed as follows. First, the model performs sentiment analysis using only the LM lexicon under certain conditions (with a certain source of words in the lexicon and a given evaluation set). After the evaluation process is finished, we take all expressions for which the model created this way gave an answer, i.e., if $v_{sent}(sentence) \neq 0$ (Equation 11), thereby marking the sentence as positive or negative according to Equation 12. Sentiment analysis is performed on these sentences with the same model to determine the accuracy of the model on them (LM on LM) and not on sentences from the whole dataset. Then, we create a model based on the explainable lexicon, we perform sentiment analysis on these same expressions (XLex on LM), and we measure the model's accuracy. In the end, the procedure is repeated once again; we create a model based on the combined lexicon, and sentiment analysis is performed only on the sentences from the dataset for which the model based on the LM lexicon had an answer ((XLex+LM) on LM).

The results of these calculations are shown in Table 14. It can be observed that in about half of the experiments, the "XLex on LM" approach has higher accuracy than the "LM on LM" approach or slightly smaller accuracy. This indicates that performing sentiment analysis based on the automatically generated explainable lexicon achieves comparable results as the lexicon manually annotated by financial experts while significantly increasing the word coverage. Thus, the approach deserves to be further explored with the goal of surpassing the results of the manually annotated lexicon. However, as stated in the introduction section, our aim is to explore whether explainable lexicons can extend standard manually annotated lexicons such as the LM lexicon. Therefore, the focus of our analysis is on the approach labeled as "(XLex+LM) on LM". As can be seen, the approach "(XLex+LM) on LM" results in higher accuracy than "LM on LM" across all experiments with the sentiment analysis model and under different conditions (different source set and evaluation set). In other words, it indicates that when the model based on the combined lexicon is applied to the sentences for which the LM lexicon gave an answer, the accuracy is always higher than when the model based on the LM lexicon is applied to the same sentences. This shows that standard, manually annotated lexicons can be improved by augmenting them with automatically extracted words obtained in an explainable way. Furthermore, this addition of words allows them to be used in the process of sentiment analysis because, as shown in Table 15, the LM lexicon is not able to perform a complete sentiment analysis (due to its weak coverage) unless words are added to it. The combined lexicon can lead to models that can be used to perform sentiment analysis while achieving higher accuracy. Adding words in an automatic, explainable way can avoid the expensive and time-consuming manual word extraction and labeling process that requires domain experts to review and analyze all words and then label them accordingly.

Besides the achieved high accuracy and increased vocabulary coverage, the proposed explainable lexicons also lead to two additional benefits: speed and size. The speed for processing sentences is an important factor in real-time production systems. If NLP processing is worth to be done at all in a system, it is worth doing it fast [70]. Table 16 shows a comparison of the times needed to perform sentiment classification by the RoBERTa transformer model, presented in Section III, and model from Section V. The model from Section V uses the combined XLex+LM lexicon where the words of the explainable lexicon come from the $nasdaq$ source. The execution speed of both models is evaluated on a central processing unit (CPU) over the evaluation version of the $financial\_phrase\_bank$ dataset given in Table 12. Generally, even though the transformer models are trained on a graphical processing unit (GPU), allowing fast execution, they are used in environments where no GPU is available [71]. This leads to slow model execution, which could become even slower as the model gets larger. Such behavior can also be observed in Table 16, where the results reveal a substantial difference in the execution time of the two models. The combined XLex+LM lexicon leads to a significantly smaller execution time (by a factor of 20) compared to the transformer model. This makes the lexicon-based model suitable for tasks that need to be performed quickly and in real-time and still lead to reasonably accurate predictions.

The second important aspect of the lexicon-based model is the size. The model size is an important factor to consider when deciding which model to be used in production systems. Transformer models are trained on large datasets and are often larger than the free disk space available on resource-constrained devices. Given that they can be too large for usage in certain production systems, transformer models typically serve to monitor smaller production models [70]. For the lexicon-based model proposed in Section V, the model size is actually represented by the size of the lexicon. The size comparison between the RoBERTa transformer model and the lexicon-based model is shown in Table 16. The difference in size is considerable. As can be seen, the lexicon-based model is about three orders of magnitude smaller than the transformer model. Although there are approaches to make transformer-based models smaller [71], transformers are not suitable for certain use cases, such as environments with limited computational resources or embedded devices. On the other hand, Table 16 shows that lexicon-based models have a size that is suitable for such applications.

The third and probably the most important advantage of lexicon-based approaches is their interpretability. Lexicon-based sentiment models are generally more interpretable than transformer-based sentiment models because they rely on a pre-defined set of rules that are easy to understand and interpret. In a lexicon-based sentiment model, each word is assigned a sentiment score based on its associated sentiment

TABLE 15: Number of sentences on which the models based on a given lexicon did not give answers.

Evaluation dataset	Number of sentences	Lexicons			
		fiqa_fpb_sentfin	nasdaq	financial_phrase_bank	Loughran-McDonald
fiqa_labeled_df	882	5	4	9	541
dev_df	398	0	0	0	216
financial_phrase_bank	885	0	0	/	522
fpb_fiqa	1542	4	3	8	937

TABLE 16: Comparison between the RoBERTa transformer model and the lexicon-based model in terms of model speed and size. The comparison is evaluated on a dataset of 885 sentences.

Model	Comparison metric	
	Speed	Size
RoBERTa transformer model	555 seconds	1.32 GB
XLex-based model (with nasdaq lexicon)	27 seconds	363 KB

value in the sentiment dictionary, and the overall sentiment of the text is calculated based on the sum or average of the sentiment scores of the words in the text. This makes it easy to understand why a particular text was classified as positive, negative, or neutral, as the sentiment scores assigned to each word in the text are transparent and interpretable. Moreover, the sentiment dictionary can be customized for specific domains or use cases, allowing for more accurate and relevant sentiment analysis. In contrast, transformer-based sentiment models are based on more complex deep learning architectures that are more difficult to interpret. Transformer models use large neural networks to learn the context and meaning of words in a sentence and assign a sentiment score to the sentence based on this understanding. While transformer-based sentiment models can achieve higher accuracy than dictionary-based models, the sentiment scores assigned to each word or phrase in the sentence are not as transparent or interpretable as they are generated by a complex neural network that learns its own set of rules based on the training data. This lack of interpretability can be a limitation for applications where it is important to understand why a particular text was classified as positive, negative, or neutral. However, transformer-based sentiment models can be useful for tasks that require a more nuanced understanding of sentiment, as they can capture the complex relationships between words and the context in which they are used. However, it's important to note that while explainable AI (XAI) methods like SHAP can provide some level of interpretability for transformer-based models, they may not always provide a complete understanding of how the model works due to the black-box nature of its inner workings. Additionally, the interpretability of XAI methods is often limited by the complexity of the model and may not be able to fully capture the nuances of natural language. Therefore, it's important to use XAI methods in conjunction with other approaches to ensure accurate and reliable sentiment analysis.

Due to their inherent advantages, explainable lexicons could potentially be used to replace standard lexicons that are nowadays still established in various domains. For example, the proposed lexicon in this paper could be used to replace the LM lexicon in the domain of finance. However, it is essential to use domain experts before nominating an explainable lexicon as the new standard for a specific application or domain. The domain experts can give an expert opinion when validating the sentiment scores of its constituent words. The involvement of domain experts in the lexicon review process could be a possible direction of future research as it could improve transparency and objectivity. Only then we can have a lexicon that is not only superior in terms of speed, size, and interpretability but also validated by human experts. This is especially important in critical applications where the quality of the results directly affects people's lives or safety. Examples of such applications may include not only finance but also knowledge extraction in medicine, legal document analysis, and risk assessment. By having an expert review validate the lexicon, the dictionary becomes more accurate and reliable, enhancing its usefulness and value to the users. It also ensures that the lexicon is consistent with the standard conventions of the language while meeting the needs and expectations of the intended audience.

VII. CONCLUSION

In this paper, we have explored the use of NLP transformer models and SHAP explainability to automatically enhance the vocabulary coverage of the Loughran-McDonald (LM) lexicon in sentiment analysis scenarios for financial applications. Our results demonstrate that standard domain-specific lexicons, such as the LM lexicon, can be expanded in an explainable way with new words without the need for laborious annotation involvement of human experts, a process that is both expensive and time-consuming.

To ensure the robustness of our findings, we provide a comprehensive validation methodology combining several different datasets by learning the dictionary on one dataset and testing it on others. We have conducted 22 separate experiments, and in all of them, the proposed methodology leads to increased performance.

The use of generated (XLex) or combined lexicons (XLex+LM) leads to significant improvements in sentiment analysis results compared to using the manually annotated lexicon alone. This improvement is demonstrated by higher accuracy and larger vocabulary coverage, directly addressing the limitations of standard, manually annotated lexicons.

Overall, the proposed XLex methodology holds great promise in advancing the field of sentiment analysis, par-

ticularly in applications where interpretability is of utmost importance. Unlike transformer models that rely on complex inner workings of neural networks, lexicon models depend on pre-defined rules, making it easy to interpret why a particular text is classified as positive, negative, or neutral. The enhanced interpretability provided by explainable lexicons makes them especially well-suited for critical applications where the quality of the results directly affects people's lives or safety. Examples of such applications include finance, medicine, legal document analysis, and risk assessment. In these areas, the transparency and explainability of the analysis process are essential for building trust and ensuring the responsible use of AI technologies.

As a future work, it would be beneficial to investigate the integration of explainable lexicons with other NLP techniques, to further enhance the performance and applicability of sentiment analysis. It is also essential to evaluate the robustness of explainable lexicons against various challenges, such as changes in language use, evolving domains, and the presence of adversarial examples.

The proposed methodology is general and adaptable, providing opportunities for future work to focus on applying the methodology in other fields beyond finance. By adopting the XLex methodology for different domains, it has the potential to significantly impact various industries, enhancing the accuracy and interpretability of sentiment analysis results while reducing the time and cost associated with manual lexicon development.

.

APPENDIX A

TABLE A.1: Features of the explainable lexicon after adding the XLex prefix.

Word	XLex Count (Selected)	XLex Total	XLex Count (Opposite)	XLex Category	XLex Sum SHAP Value (Selected)	XLex Average SHAP Value (Selected)	XLex Max SHAP Value (Selected)	XLex Min SHAP Value (Selected)	XLex Ratio (Selected)	XLex Sum SHAP Value (Opposite)	XLex Average SHAP Value (Opposite)
abet	1	1	0	positive	0.025090	0.025090	0.025090	0.025090	1.000000	0.000000	0.000000
abide	1	1	0	positive	0.003838	0.003838	0.003838	0.003838	1.000000	0.000000	0.000000
abo	1	1	0	positive	0.000976	0.000976	0.000976	0.000976	1.000000	0.000000	0.000000
aboard	2	2	0	positive	0.245279	0.122640	0.210718	0.034561	1.000000	0.000000	0.000000
abolition	1	1	0	positive	0.006430	0.006430	0.006430	0.006430	1.000000	0.000000	0.000000
ken	4	7	3	negative	0.114558	0.028640	0.086224	0.003522	0.504271	0.084463	0.028154
peter	9	15	6	negative	0.146561	0.016285	0.068352	0.000112	0.477515	0.106909	0.017818
uri	1	2	1	negative	0.016012	0.016012	0.016012	0.016012	0.565765	0.012289	0.012289
military	76	98	22	negative	2.958678	0.038930	0.169050	0.001473	0.773876	0.250255	0.011375
depth	1	2	1	positive	0.018833	0.018833	0.018833	0.018833	0.997290	0.000051	0.000051

TABLE A.2: Features of the LM lexicon after adding the LM prefix.

Word	LM Count (Selected)	LM Total	LM Count (Opposite)	LM Category	LM Sum SHAP Value (Selected)	LM Average SHAP Value (Selected)	LM Max SHAP Value (Selected)	LM Min SHAP Value (Selected)	LM Ratio (Selected)	LM Sum SHAP Value (Opposite)	LM Average SHAP Value (Opposite)
abet	1	1	0	negative	1	1	1	1	1	0	0
accomplish	1	1	0	positive	1	1	1	1	1	0	0
advance	1	1	0	positive	1	1	1	1	1	0	0
advantage	1	1	0	positive	1	1	1	1	1	0	0
advantageous	1	1	0	positive	1	1	1	1	1	0	0
writeoff	1	1	0	negative	1	1	1	1	1	0	0
writeoffs	1	1	0	negative	1	1	1	1	1	0	0
wrongful	1	1	0	negative	1	1	1	1	1	0	0
wrongfully	1	1	0	negative	1	1	1	1	1	0	0
wrongly	1	1	0	negative	1	1	1	1	1	0	0

TABLE A.3: Values of selected features of the combined lexicon for words that appear in both the explainable and LM lexicon.

Word	XLex Count (Selected)	XLex Total	XLex Count (Opposite)	XLex Average SHAP Value (Selected)	XLex Source	LM Average SHAP Value (Selected)	LM Category	LM Sum SHAP Value (Opposite)	LM Source	LM Max SHAP Value (Opposite)
abet	1.0	1.0	0.0	0.025090	XLex	1	negative	0	LM	0
accomplish	1.0	1.0	0.0	0.078244	XLex	1	positive	0	LM	0
advance	9.0	9.0	0.0	0.113294	XLex	1	positive	0	LM	0
advantage	7.0	7.0	0.0	0.431898	XLex	1	positive	0	LM	0
advantageous	1.0	1.0	0.0	0.441719	XLex	1	positive	0	LM	0
good	65.0	68.0	3.0	0.149197	XLex	1	positive	0	LM	0
pose	10.0	20.0	10.0	0.027118	XLex	1	negative	0	LM	0
gain	14.0	16.0	2.0	0.157810	XLex	1	positive	0	LM	0
evasion	2.0	3.0	1.0	0.032876	XLex	1	negative	0	LM	0
defeat	10.0	12.0	2.0	0.077834	XLex	1	negative	0	LM	0

TABLE A.4: Values of selected features of the combined lexicon for words that appear in either the explainable or the LM lexicon.

Word	XLex Count (Selected)	XLex Total	XLex Count (Opposite)	XLex Average SHAP Value (Selected)	XLex Source	LM Average SHAP Value (Selected)	LM Category	LM Sum SHAP Value (Opposite)	LM Source	LM Max SHAP Value (Opposite)
abide	1.0	1.0	0.0	0.003838	XLex	NaN	NaN	NaN	NaN	NaN
abo	1.0	1.0	0.0	0.000976	XLex	NaN	NaN	NaN	NaN	NaN
aboard	2.0	2.0	0.0	0.122640	XLex	NaN	NaN	NaN	NaN	NaN
abolition	1.0	1.0	0.0	0.006430	XLex	NaN	NaN	NaN	NaN	NaN
abroad	3.0	3.0	0.0	0.039073	XLex	NaN	NaN	NaN	NaN	NaN
writeoff	NaN	NaN	NaN	NaN	NaN	1	negative	0	LM	0
writeoffs	NaN	NaN	NaN	NaN	NaN	1	negative	0	LM	0
wrongful	NaN	NaN	NaN	NaN	NaN	1	negative	0	LM	0
wrongfully	NaN	NaN	NaN	NaN	NaN	1	negative	0	LM	0
wrongly	NaN	NaN	NaN	NaN	NaN	1	negative	0	LM	0

REFERENCES

[1] M. Hasan, J. Popp, J. Oláh *et al.*, "Current landscape and influence of big data on finance," *Journal of Big Data*, vol. 7, no. 1, pp. 1–17, 2020.

[2] I. Goldstein, C. S. Spatt, and M. Ye, "Big data in finance," *The Review of Financial Studies*, vol. 34, no. 7, pp. 3213–3225, 2021.

[3] N. Mohamed and J. Al-Jaroodi, "Real-time big data analytics: Applications and challenges," in *2014 international conference on high performance computing & simulation (HPCS)*. IEEE, 2014, pp. 305–310.

[4] V. Ravi and S. Kamaruddin, "Big data analytics enabled smart financial services: opportunities and challenges," in *Big Data Analytics: 5th International Conference, BDA 2017, Hyderabad, India, December 12-15, 2017, Proceedings 5*. Springer, 2017, pp. 15–39.

[5] M. Cao, R. Chychyla, and T. Stewart, "Big data analytics in financial statement audits," *Accounting Horizons*, vol. 29, no. 2, pp. 423–429, 2015.

[6] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Predictive sentiment analysis of tweets: A stock market application," in *Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data: Third International Workshop, HCI-KDD 2013, Held at SouthCHI 2013, Maribor, Slovenia, July 1-3, 2013. Proceedings*. Springer, 2013, pp. 77–88.

[7] A. Derakhshan and H. Beigy, "Sentiment analysis on stock social media for stock price movement prediction," *Engineering Applications of Artificial Intelligence*, vol. 85, pp. 569–578, 2019.

[8] R. Ren, D. D. Wu, and T. Liu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," *IEEE Systems Journal*, vol. 13, no. 1, pp. 760–770, 2018.

[9] R. Yang, L. Yu, Y. Zhao, H. Yu, G. Xu, Y. Wu, and Z. Liu, "Big data analytics for financial market volatility forecast based on support vector machine," *International Journal of Information Management*, vol. 50, pp. 452–462, 2020.

[10] F.-T. Tsai, H.-M. Lu, and M.-W. Hung, "The effects of news sentiment and coverage on credit rating analysis," 2010.

[11] S. Gül, Ö. Kabak, and I. Topcu, "A multiple criteria credit rating approach utilizing social media data," *Data & Knowledge Engineering*, vol. 116, pp. 80–99, 2018.

[12] H.-M. Lu, F.-T. Tsai, H. Chen, M.-W. Hung, and S.-H. Li, "Credit rating change modeling using news and financial ratios," *ACM Transactions on Management Information Systems (TMIS)*, vol. 3, no. 3, pp. 1–30, 2012.

[13] D. Zhang, W. Xu, Y. Zhu, and X. Zhang, "Can sentiment analysis help mimic decision-making process of loan granting? a novel credit risk evaluation approach using gmkl model," in *2015 48th Hawaii International Conference on System Sciences*. IEEE, 2015, pp. 949–958.

[14] B. Yoon, Y. Jeong, and S. Kim, "Detecting a risk signal in stock investment through opinion mining and graph-based semi-supervised learning," *IEEE Access*, vol. 8, pp. 161 943–161 957, 2020.

[15] J. R. McColl-Kennedy, M. Zaki, K. N. Lemon, F. Urmetzer, and A. Neely, "Gaining customer experience insights that matter," *Journal of service research*, vol. 22, no. 1, pp. 8–26, 2019.

[16] L. Ziora, "The sentiment analysis as a tool of business analytics in contemporary organizations," *Studia Ekonomiczne*, vol. 281, pp. 234–241, 2016.

[17] H. Mili, I. Benzarti, M.-J. Meurs, A. Obaid, J. Gonzalez-Huerta, N. Haj-Salem, and A. Boubaker, "Context aware customer experience management: A development framework based on ontologies and computational intelligence," *Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence*, pp. 273–311, 2016.

[18] X. Tian, J. S. He, and M. Han, "Data-driven approaches in fintech: a survey," *Information Discovery and Delivery*, 2021.

[19] C.-C. Chen, H.-H. Huang, and H.-H. Chen, "Fintech applications," in *From Opinion Mining to Financial Argument Mining*. Springer, 2021, pp. 73–87.

[20] H. Liu, I. Chatterjee, M. Zhou, X. S. Lu, and A. Abusorrah, "Aspect-based sentiment analysis: A survey of deep learning methods," *IEEE Transactions on Computational Social Systems*, vol. 7, no. 6, pp. 1358–1375, 2020.

[21] F. Benedetto and A. Tedeschi, "Big data sentiment analysis for brand monitoring in social media streams by cloud computing," *Sentiment Analysis and Ontology Engineering: An Environment of Computational Intelligence*, pp. 341–377, 2016.

[22] D. Alessia, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, tools and applications for sentiment analysis implementation," *International Journal of Computer Applications*, vol. 125, no. 3, 2015.

[23] M. Taboada, J. Brooke, M. Tofiloski, K. Voll, and M. Stede, "Lexicon-based methods for sentiment analysis," *Computational linguistics*, vol. 37, no. 2, pp. 267–307, 2011.

[24] S. Taj, B. B. Shaikh, and A. F. Meghji, "Sentiment analysis of news articles: a lexicon based approach," in *2019 2nd international conference on computing, mathematics and engineering technologies (iCoMET)*. IEEE, 2019, pp. 1–5.

[25] M. Wankhade, A. C. S. Rao, and C. Kulkarni, "A survey on sentiment analysis methods, applications, and challenges," *Artificial Intelligence Review*, vol. 55, no. 7, pp. 5731–5780, 2022.

[26] T. Loughran and B. McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-ks," *The Journal of finance*, vol. 66, no. 1, pp. 35–65, 2011.

[27] E. Boiy and M.-F. Moens, "A machine learning approach to sentiment analysis in multilingual web texts," *Information retrieval*, vol. 12, pp. 526–558, 2009.

[28] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis," in *Proceedings of the 2012 ACM research in applied computation symposium*, 2012, pp. 1–7.

[29] S. Malviya, A. K. Tiwari, R. Srivastava, and V. Tiwari, "Machine learning techniques for sentiment analysis: A review," *SAMRIDDHI: A Journal of Physical Sciences, Engineering and Technology*, vol. 12, no. 02, pp. 72–78, 2020.

[30] M. Neethu and R. Rajasree, "Sentiment analysis in twitter using machine learning techniques," in *2013 fourth international conference on computing, communications and networking technologies (ICCCNT)*. IEEE, 2013, pp. 1–5.

[31] L. Zhang, S. Wang, and B. Liu, "Deep learning for sentiment analysis: A survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1253, 2018.

[32] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: a review," *Artificial Intelligence Review*, vol. 53, no. 6, pp. 4335–4385, 2020.

[33] N. C. Dang, M. N. Moreno-García, and F. De la Prieta, "Sentiment analysis based on deep learning: A comparative study," *Electronics*, vol. 9, no. 3, p. 483, 2020.

[34] D. Tang, B. Qin, and T. Liu, "Deep learning for sentiment analysis: successful approaches and future challenges," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 5, no. 6, pp. 292–303, 2015.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[37] K. Mishev, A. Gjorgjevikj, I. Vodenska, L. T. Chitkushev, and D. Trajanov, "Evaluation of sentiment analysis in finance: from lexicons to transformers," *IEEE Access*, vol. 8, pp. 131 662–131 682, 2020.

[38] X. S. Huang, F. Perez, J. Ba, and M. Volkovs, "Improving transformer optimization through better initialization," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4475–4483.

[39] S. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *arXiv preprint arXiv:1705.07874*, 2017.

[40] L. Dodevska, V. Petreski, K. Mishev, A. Gjorgjevikj, I. Vodenska, L. Chitkushev, and D. Trajanov, "Predicting companies stock price direction by using sentiment analysis of news articles," in *Proceedings of the 15th Annual International Conference on Computer Science and Education in Computer Science*, 2019, pp. 37–42.

[41] T. Loughran and B. McDonald, "Measuring readability in financial disclosures," *the Journal of Finance*, vol. 69, no. 4, pp. 1643–1671, 2014.

[42] ——, "Textual analysis in accounting and finance: A survey," *Journal of Accounting Research*, vol. 54, no. 4, pp. 1187–1230, 2016.

[43] S. Krishnamoorthy, "Sentiment analysis of financial news articles using performance indicators," *Knowledge and Information Systems*, vol. 56, no. 2, pp. 373–394, 2018.

[44] P. J. Stone, D. C. Dunphy, and M. S. Smith, "The general inquirer: A computer approach to content analysis." 1966.

[45] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets," *arXiv preprint arXiv:1308.6242*, 2013.

[46] D. T. Vo and Y. Zhang, "Don't count, predict! an automatic approach to learning sentiment lexicons for short text," in *Proceedings of the 54th*

Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 219–224.
[47] F. Viegas, M. S. Alvim, S. Canuto, T. Rosa, M. A. Gonçalves, and L. Rocha, "Exploiting semantic relationships for unsupervised expansion of sentiment lexicons," *Information Systems*, vol. 94, p. 101606, 2020.
[48] T. Bos and F. Frasincar, "Automatically building financial sentiment lexicons while accounting for negation," *Cognitive Computation*, pp. 1–19, 2021.
[49] H. Saif, Y. He, M. Fernandez, and H. Alani, "Adapting sentiment lexicons using contextual semantics for sentiment analysis of twitter," in *European Semantic Web Conference*. Springer, 2014, pp. 54–63.
[50] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proceedings of the 2006 conference on empirical methods in natural language processing*, 2006, pp. 355–363.
[51] N. Kaji and M. Kitsuregawa, "Building lexicon for sentiment analysis from massive collection of html documents," in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007, pp. 1075–1083.
[52] ——, "Automatic construction of polarity-tagged corpus from html documents," in *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, 2006, pp. 452–459.
[53] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proceedings of the conference on empirical methods in natural language processing. conference on empirical methods in natural language processing*, vol. 2016. NIH Public Access, 2016, p. 595.
[54] O. Araque, G. Zhu, and C. A. Iglesias, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowledge-Based Systems*, vol. 165, pp. 346–359, 2019.
[55] W. Zhao, T. Joshi, V. N. Nair, and A. Sudjianto, "Shap values for explaining cnn-based text classification models," *arXiv preprint arXiv:2008.11825*, 2020.
[56] K. E. Mokhtari, B. P. Higdon, and A. Başar, "Interpreting financial time series with shap values," in *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, 2019, pp. 166–172.
[57] X. Xiaomao, Z. Xudong, and W. Yuanfang, "A comparison of feature selection methodology for solving classification problems in finance," in *Journal of Physics: Conference Series*, vol. 1284, no. 1. IOP Publishing, 2019, p. 012026.
[58] M. Rizinski, H. Peshov, K. Mishev, L. T. Chitkushev, I. Vodenska, and D. Trajanov, "Ethically responsible machine learning in fintech," *IEEE Access*, vol. 10, pp. 97 531–97 554, 2022.
[59] E. Kokalj, B. Škrlj, N. Lavrač, S. Pollak, and M. Robnik-Šikonja, "Bert meets shapley: Extending shap explanations to transformer-based classifiers," in *Proceedings of the EACL Hackashop on News Media Content Analysis and Automated Report Generation*, 2021, pp. 16–21.
[60] S. Consoli, L. Barbaglia, and S. Manzan, "Fine-grained, aspect-based sentiment analysis on economic and financial lexicon," *Knowledge-Based Systems*, vol. 247, p. 108781, 2022.
[61] A. Moreno-Ortiz, J. Fernández-Cruz, and C. P. C. Hernández, "Design and evaluation of sentiecon: A fine-grained economic/financial sentiment lexicon from a corpus of business news," in *Proceedings of The 12th Language Resources and Evaluation Conference*, 2020, pp. 5065–5072.
[62] M. Yekrangi and N. Abdolvand, "Financial markets sentiment analysis: Developing a specialized lexicon," *Journal of Intelligent Information Systems*, vol. 57, pp. 127–146, 2021.
[63] J. Fang and B. Chen, "Incorporating lexicon knowledge into svm learning to improve sentiment classification," in *Proceedings of the workshop on sentiment analysis where AI meets psychology (SAAIP 2011)*, 2011, pp. 94–100.
[64] R. Catelli, S. Pelosi, and M. Esposito, "Lexicon-based vs. bert-based sentiment analysis: A comparative study in italian," *Electronics*, vol. 11, no. 3, p. 374, 2022.
[65] K. Mishev, A. Gjorgjevikj, R. Stojanov, I. Mishkovski, I. Vodenska, L. Chitkushev, and D. Trajanov, "Performance evaluation of word and sentence embeddings for finance headlines sentiment analysis," in *International Conference on ICT Innovations*. Springer, 2019, pp. 161–172.
[66] P. Malo, A. Sinha, P. Takala, O. Ahlgren, and I. Lappalainen, "Learning the roles of directional expressions and domain concepts in financial news analysis," in *2013 IEEE 13th International Conference on Data Mining Workshops*. IEEE, 2013, pp. 945–954.
[67] K. Cortis, A. Freitas, T. Daudert, M. Huerlimann, M. Zarrouk, S. Handschuh, and B. Davis, "Semeval-2017 task 5: Fine-grained sentiment analysis on financial microblogs and news." Association for Computational Linguistics (ACL), 2017.
[68] S. Mazzanti, "Shap values explained exactly how you wished someone explained to you," https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30, 2020, [Online; accessed 05-July-2021].
[69] S. Lundberg, "SHapley Additive exPlanations," https://github.com/slundberg/shap, 2018, [Online; accessed 29-Jan-2023].
[70] M. Honnibal and I. Montani, "spaCy meets Transformers: Fine-tune BERT, XLNet and GPT-2," https://explosion.ai/blog/spacy-transformers, 2019, [Online; accessed 10-February-2023].
[71] N. Lathia, "When is a neural net too big for production?" https://neal-lathia.medium.com/when-is-a-neural-net-too-big-for-production-4315452193ef, 2019, [Online; accessed 10-March-2023].

MARYAN RIZINSKI received the B.S. and M.S. degrees in electrical engineering and information technologies from University Ss. Cyril and Methodius in Skopje where he is a Ph.D. candidate in computer science. He is currently an engineering manager at Bosch, with over ten years of industry experience leading globally-distributed software engineering teams. His expertise spans multiple aspects of the software project lifecycle management, from planning, requirement gathering, and analysis, estimations to driving delivery, rollout, and troubleshooting for international customers. Throughout his professional career, he has managed the implementation of Internet of Things (IoT) and fiber-optics infrastructure projects and has been mentoring and consulting startup IT companies. He is also a lecturer of computer science at Boston University's Metropolitan College where he is teaching and facilitating networking and data science classes. His doctoral research focuses on novel approaches for using machine learning (ML) and natural language processing (NLP) in the financial industry and other related areas. His research aims to enable more accurate decision-making and address fundamental problems of improving the explainability of deep-learning models and addressing ML-related ethical challenges in finance applications. His past research interests focused on computer networking, wireless communications, and new Internet and IoT architectures.

HRISTIJAN PESHOV received the Bachelor of Science degree in software engineering and information systems from the Faculty of Computer Science and Engineering, Saints Cyril and Methodius University in Skopje, in 2022. He also works as a Software Engineer. His research interests include data science, machine learning, explainable AI, natural language processing, and network analysis.

KOSTADIN MISHEV received the bachelor's degree in informatics and computer engineering and the master's degree in computer networks and e-technologies degree from Saints Cyril and Methodius University, Skopje, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree. He is also a Teaching and a Research Assistant with the Faculty of Computer Science and Engineering, Saints Cyril and Methodius University. His research interests include Data Science, Natural Language Processing, Semantic Web, Web technologies, and Computer Networks.

MILOS JOVANOVIK is an Associate Professor at the Faculty of Computer Science and Engineering, at the Ss. Cyril and Methodius University in Skopje. He's also a Senior R&D Knowledge Graphs Engineer at OpenLink Software, London, UK. He obtained his Ph.D. in 2016 at the Ss. Cyril and Methodius University in Skopje, in the field of Computer Science and Engineering, with a doctoral thesis in the domain of Linked Data. He has published over 50 scientific papers and has participated in over 30 research projects on international and national levels. His main research interests include Knowledge Graphs, Linked Data, Open Data, and Data Science.

DIMITAR TRAJANOV (Member, IEEE) received a Ph.D. degree in computer science. He is Full professor at the Faculty of Computer Science and Engineering - ss. Cyril and Methodius University – Skopje and Visiting Research Professor at Boston University. From March 2011 until September 2015, he was the founding Dean of the Faculty of Computer Science and Engineering, and in his tenure, the faculty became the largest technical Faculty in Macedonia. Dimitar Trajanov is the leader of the Regional Social Innovation Hub, established in 2013 as a cooperation between UNDP and the Faculty of Computer Science and Engineering. Dimitar Trajanov is the author of more than 180 journal and conference papers and seven books. He has been involved in more than 70 research and industry projects, of which in more than 40 projects as a project leader. His research interests include Data Science, Machine Learning, NLP, FinTech, Semantic Web, Open Data, Social Innovation, e-commerce, Technology for Development, and Climate Change.