IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 1 December 2023, accepted 29 December 2023, date of publication 4 January 2024,
date of current version 18 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3349970

RESEARCH ARTICLE

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 (rizinski@bu.edu)

This work was supported in part by the European Cooperation in Science and Technology (COST) through the COST Action CA18209-NexusLinguarum "European Network for Web-Centred Linguistic Data Science."

ABSTRACT Lexicon-based sentiment analysis in finance leverages specialized, manually annotated lexicons created by human experts to extract sentiment from financial texts effectively. 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. Our experiments show that XLex outperforms LM when applied to general financial texts, resulting in enhanced word coverage and an overall increase in classification accuracy by 0.431. Furthermore, by employing XLex to extend LM, we create a combined dictionary, XLex+LM, which achieves an even higher accuracy improvement of 0.450. 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

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

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], [4], [5].

Sentiment analysis can play a crucial role in analyzing, interpreting, and extracting insights from big financial data. Sentiment analysis has become increasingly important in the field of finance and fintech, where it has 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], [7], [8], [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 monitor 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], [11], [12], [13], [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], [16], [17], [18], [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], [21], [22].

Lexicon-based sentiment analysis is a commonly used approach that relies on pre-defined sets of words known as lexicons [23], [24], [25]. Lexicons are manually annotated by experts in the field, and sentiment scores are assigned 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. Generic lexicons are also 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], [28], [29], [30] and deep learning (DL) techniques [25], [31], [32], [33], [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. To create the LM lexicon, its authors employed Release 4.0 of the *2of12inf* dictionary as a basis and extended it using 10-X fillings.[1] The LM authors do not extract all words from the fillings; they rather use only those words that appear frequently (with a frequency count of 50 or more).[2] This means that LM will not have a recall even for a large number of the sentences present in the 10-X fillings. The approach of using lexicons annotated by experts has its limitations as 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 improved 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.

Our research focuses on sentiment analysis in the field of finance, driven by the recognition of the Loughran-McDonald

[1]Detailed documentation on the LM development can be found in https://sraf.nd.edu/loughranmcdonald-master-dictionary/

[2]It is worth mentioning that the XLex methodology that we develop in this paper is not bound to the frequency count.

lexicon as a standard baseline for sentiment analysis in this domain. As indicated in a 2016 paper by the LM authors [40], the LM dictionary has been used in various studies to measure the sentiment in newspaper articles and columns such as [41], [42], and [43], among others. This well-established lexicon provides a solid foundation for our study, allowing us to pose a well-defined research question. Furthermore, this work is part of a broader research project conducted by our team, which explores the application of advanced NLP techniques in finance-related contexts. Consequently, our primary emphasis lies on sentiment analysis within the field of 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 accuracy and 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 [44]. 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 extracting words and 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 [40], [45], [46]. Reference [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 the Loughran-McDonald (LM) lexicon, to more accurately capture sentiments in financial texts. Other dictionaries used in finance include General Inquirer (GI) [47], 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, [48] 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 [48] relied on calculating pointwise mutual information (PMI) between words and emoticons, [49] uses a simple neural network to train lexicons that improve the accuracy of predicting emoticons in tweets.

The study in [50] 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 data labeling process is not only challenging and costly but also suffers from the drawback of producing

limited lexicon coverage. Therefore, as its main contribution, [50] 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 [50] 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 [51]. Adapting lexicons that depend on word context is studied in [52]; this work captures the context of words appearing in tweet messages and uses it to update their prior sentiment accordingly. The methodology in [52] showed improvement in lexicon performance due to the sentiment adaptation to the underlying context. Earlier works explored various directions such as lexicon generation from a massive collection of web resources [53], automatic lexicon expansion for domain-oriented sentiment analysis [54], construction of polarity-tagged corpus from HTML documents [55], etc.

Inducing domain-specific sentiment lexicons from small seed words and domain-specific corpora is studied in [56], 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 [57]. This paper emphasizes that the capability to infer embedding models automatically leads to higher vocabulary coverage. The experiments in [57] 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 has given rise to a vast amount of research on the explainability of ML models. 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) [58]. When combined with CNNs, SHAP effectively explains the importance of local features while also taking advantage of CNN's potential to reduce the high feature dimensionality of NLP tasks. CNNs are known to outperform other ML algorithms for text classification, which implies that the SHAP-based analysis of CNNs in [58] 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 [59] and financial data of bankrupt companies [60]. A comprehensive study has been performed in [61] 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 to improve the visualizations of the generated explanations [62].

Extracting sentiment from news text, social media, and blogs has gained increasing interest in economics and finance. The study in [63] 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 [64]. A hybrid lexicon that combines corpus-based and dictionary-based methods with statistical and semantic measures is proposed in [65], 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 [66] 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 Support Vector Machine (SVM) and showed that it could improve the accuracy of sentiment analysis tasks.

While acknowledging the advantages of deep learning methods, the results in [67] showed that lexicon-based methods are preferred for use 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

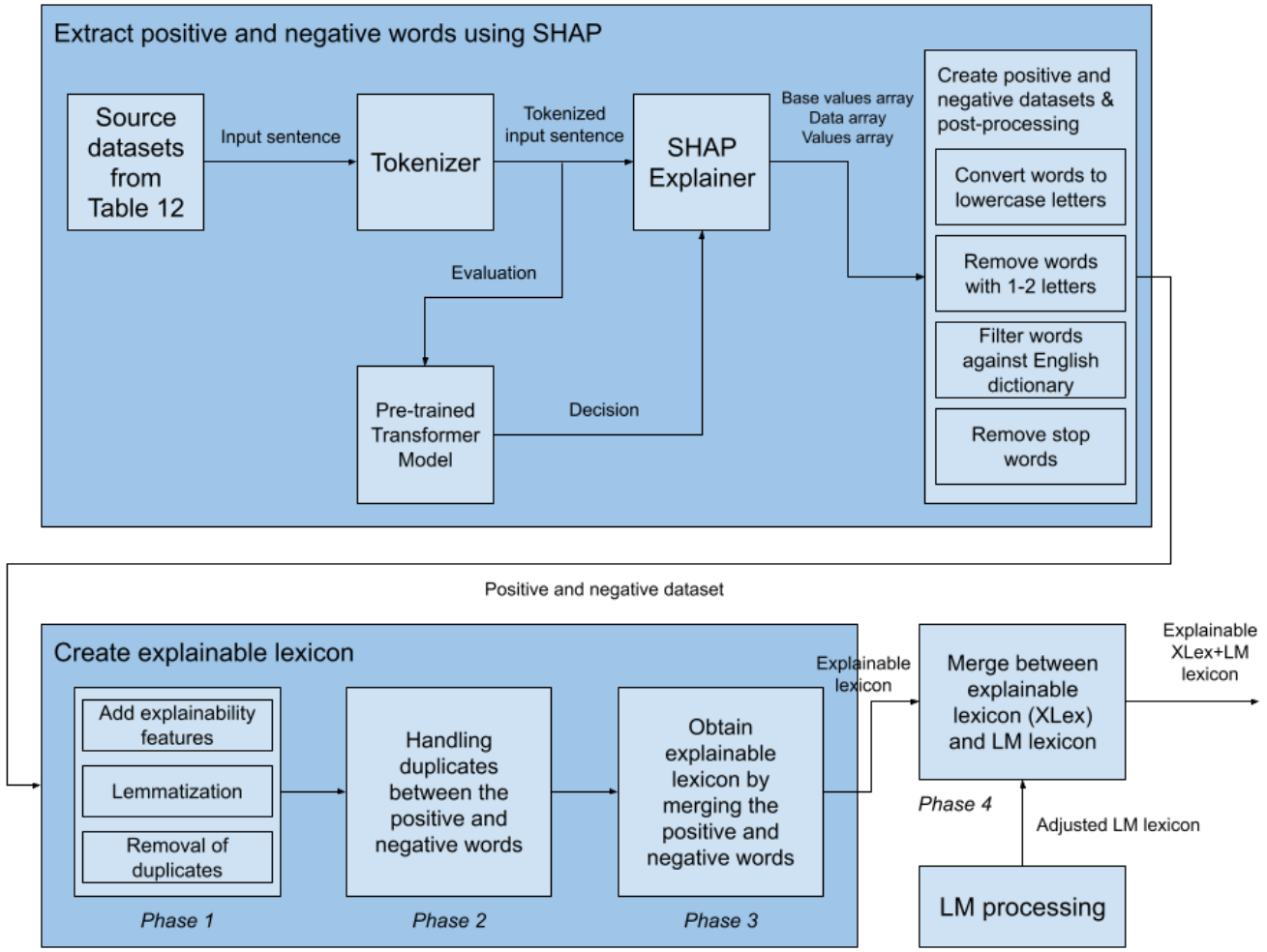

FIGURE 1. Architecture of the data processing pipeline for generating the explainable lexicon (XLex). The upper section of the figure, labeled as "Extract positive and negative words using SHAP", illustrates the word extraction process using SHAP, followed by post-processing steps to generate separate positive and negative word datasets from the chosen source datasets. The lower section of the figure, referred to as "Create explainable lexicon", encompasses adding explainability features, handling duplicates, and merging the positive and negative datasets to form the comprehensive explainable lexicon XLex. The pipeline concludes by merging XLex with the Loughran-McDonald (LM) lexicon, resulting in the combined XLex+LM lexicon.

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 WORD EXTRACTION METHODOLOGY

The construction of a lexicon for sentiment analysis comprises several consecutive stages, each involving suitable text processing.[3] To facilitate sentiment analysis, the lexicon must incorporate words from both positive and negative polarities. In this section, we explain the steps involved in generating the positive and negative sentiment sets, which will be merged to form an explainable lexicon.

[3]The source code and datasets related to the XLex methodology, including all conducted experiments, are accessible on GitHub at the following link: https://github.com/hristijanpeshov/SHAP-Explainable-Lexicon-Model

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 subsections of the paper.

A. A TRANSFORMER-BASED MODEL FOR SENTIMENT ANALYSIS

To develop an explainable lexicon, we begin by using a transformer-based model tailored for sentiment analysis. Specifically, FinBERT is a notable model in this domain, designed explicitly for analyzing financial texts [68]. However, FinBERT is fine-tuned on a closed dataset comprising 10,000 sentences from analyst reports sourced from Thompson Reuters's proprietary Investext database. To ensure a controlled environment over the fine-tuning process, we take charge of it by using a publicly available financial sentiment dataset and one of the available base transformer models, specifically RoBERTa. According to a survey [37], the RoBERTa model demonstrates exceptional performance

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

in various finance-related sentiment classification tasks, achieving an accuracy of 94%. Therefore, we decided to use RoBERTa as our starting model.

To provide a comprehensive analysis, we also included results obtained using the FinBERT model. While both models produce comparable results (as shown in Table 14), the key advantage of using the fine-tuned RoBERTa model lies in the customized approach and rigorous control we have over the data and fine-tuning process, thereby boosting our flexibility to perform experiments.

The datasets utilized for learning the explainable dictionaries are presented in Table 12, where they are labeled as ''Source'' datasets. The results of the XLex model are then evaluated using the datasets labeled as ''Evaluation''.

To construct the sentiment dictionaries, we use SHAP to interpret the output of the pre-trained transformer model. This approach aids us in identifying individual words and classifying them as either positive or negative in sentiment. This approach is discussed in detail in Subsection III-B.

The RoBERTa-Large model is originally trained in a self-supervised manner on an extensive corpus of English text.[4] The RoBERTa-Large model comprises 24 layers, 1024 hidden units, 16 attention heads and is based on a total of 355 million parameters. Subsequently, we fine-tune the foundational RoBERTa model using the approach outlined in [37]. The fine-tuning is conducted on a merged dataset comprising the Financial PhraseBank [69] and SemEval-2017-Task5 datasets [70]. This fine-tuning procedure ensures that the resulting model is specialized for the domain of financial sentiment analysis. This fine-tuned model will also be referred to as the RoBERTa-based model for convenience in the following sections.

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 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.

[4]For a comprehensive explanation of the pretraining methodology employed for RoBERTa-Large, the reader is directed to the official Hugging Face documentation: https://huggingface.co/roberta-large

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

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

As illustrated in Table 1, 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 [37]. The sentences in the dataset are shuffled and divided into 80% training set and 20% testing set. The training and testing 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 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 [71]. Its approach uses Shapley values from game theory to explain the output of ML models [72]. Game theory is characterized by two elements: a game and players. In SHAP, 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 pos-

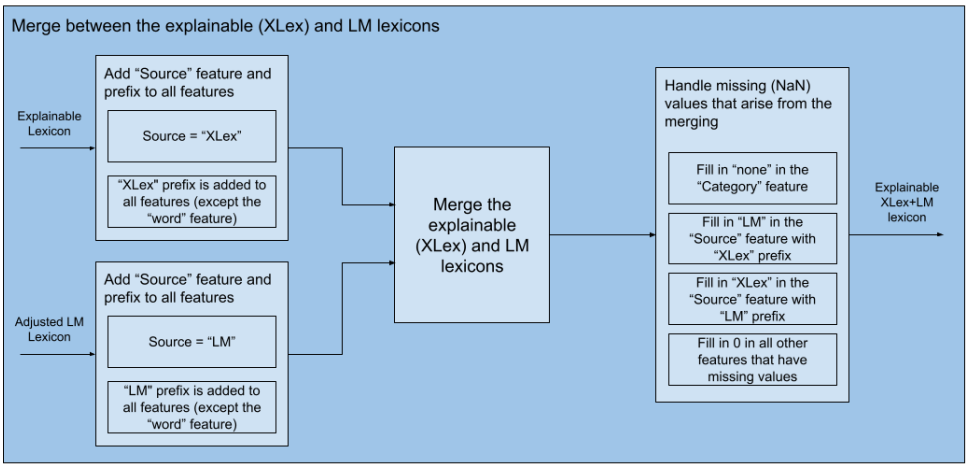

FIGURE 2. The explainable lexicon (XLex) and the LM lexicon are merged to form the combined XLex+LM lexicon. Before the merging process, the "Source" feature is introduced to both XLex and LM, and all features (excluding the "word" feature) are appropriately prefixed to enable identification of XLex features as well as LM features within the combined lexicon. Handling of missing values takes place subsequent to the merging.

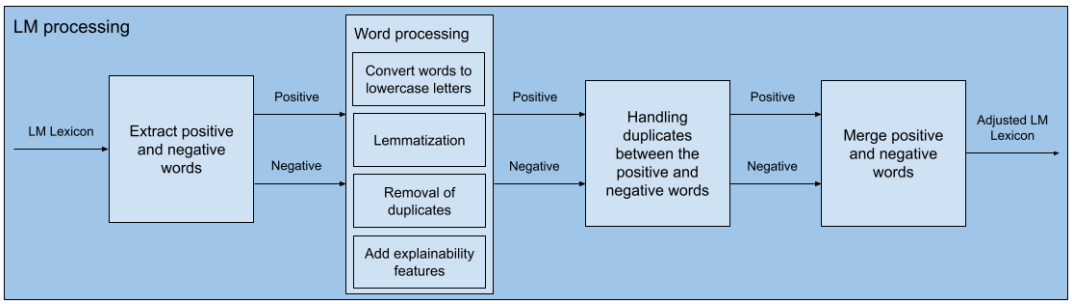

FIGURE 3. The LM lexicon undergoes a preparatory adjustment process to enable its seamless integration with the explainable XLex lexicon, resulting in the formation of the combined XLex+LM lexicon. This adjustment process includes the extraction of positive and negative words, subsequent word processing, handling of duplicates, and the final step of merging the positive and negative word sets.

sible to aggregate them, allowing us to calculate global values for the overall contribution of the features across all samples.

To evaluate the features' contributions to the model prediction for a given sample, SHAP creates a copy of the model for each combination of the input features. Each of these models is the same, the only difference is the combination of features passed to the model. In one of these combinations, 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 (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 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

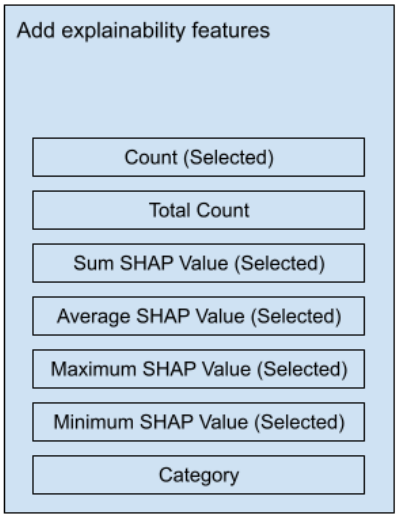

FIGURE 4. A list of explainability features based on SHAP added in the explainable and LM lexicons. For the LM lexicon, all features except "Category" are assigned the value of 1 as their default value.

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. Using these 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. As a result, these words are not accurately represented since they are divided into parts based on more common entries found in the tokenizer's 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 LEXICON CREATION METHODOLOGY

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),

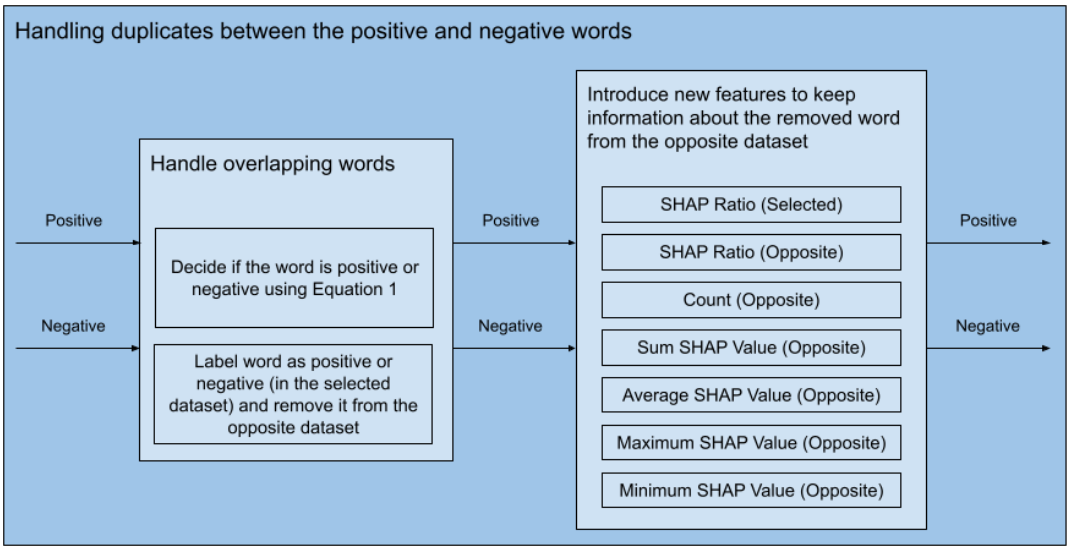

FIGURE 5. The process of dealing with duplicate entries between the positive and negative words for each of the explainable (XLex) and LM lexicons. In the case of the LM lexicon, features designated as "Opposite" are assigned a default value of 0. The "Total Count" feature can be obtained by deriving it from the values of "Count (Selected)" and "Count (Opposite)".

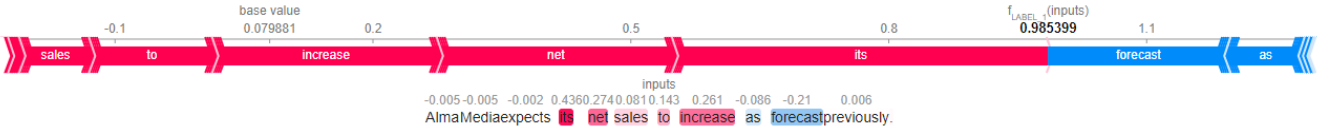

FIGURE 6. An example of using SHAP for evaluating the word contributions to the sentiment of a sentence.

while the other contains all words with negative sentiment 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. The evaluation results 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 for each feature (column) 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,

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

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 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 ($5.33 \div 18 = 0.3$), 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.

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:

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

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, the

TABLE 5. Features of the words in the lexicons.

Feature name	Feature notation	Feature description
Word feature		
Word	$word$	A 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)	$SHAP^{opp}_{sum}$	Sum SHAP value of the word in the opposite sentiment
Average SHAP Value (Opposite)	$SHAP^{opp}_{avg}$	Average SHAP value of the word in the opposite sentiment
Maximum SHAP Value (Opposite)	$SHAP^{opp}_{max}$	Maximum SHAP value of the word in the opposite sentiment
Minimum SHAP Value (Opposite)	$SHAP^{opp}_{min}$	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 $SHAP^{opp}_{avg}$
SHAP Ratio (Opposite)	$SHAP^{opp}_{ratio}$	Ratio between $SHAP^{opp}_{avg}$ and the sum of $SHAP_{avg}$ and $SHAP^{opp}_{avg}$
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

"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). As can be seen, this word has a sum SHAP value in the positive and negative dataset of $SHAP^{pos}_{sum} = 0.39$ and $SHAP^{neg}_{sum} = 0.023$, respectively. Given that $SHAP^{pos}_{sum} > SHAP^{neg}_{sum}$, 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: $count_{opp} = 7$, $SHAP^{opp}_{sum} = 0.023$, $SHAP^{opp}_{avg} = 0.0033$, $SHAP^{opp}_{max} = 0.009$, $SHAP^{opp}_{min} = 0.0001$. The SHAP ratio of the word in the two datasets is calculated as $SHAP_{ratio} = \frac{0.026}{0.026+0.0033} = 0.887$; $SHAP^{opp}_{ratio} = \frac{0.0033}{0.026+0.0033} = 0.113$. 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 $SHAP^{opp}_{avg}$ 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

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

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.[5] 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 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 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.

[5]To ensure the diagrams fit within the page limits, only a subset of the dataset features are depicted in this and subsequent lexicon-related diagrams.

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

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

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 present 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 (XLex), 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,

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

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 19-20 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 21. 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 22. Words of this type do not appear in both lexicons and 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 22. 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 22 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.

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

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

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.

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 a tokenizer. We employ two types of tokenizers in our methodology. For XLex, we use the corresponding RoBERTa/FinBERT tokenizers, and for LM, we use NLTK, which is a standard rule-based tokenizer. After applying the tokenizers, 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, $1 \leq i \leq n$:

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

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:

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

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:

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

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:

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

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:

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

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:

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

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, F1 score, and MCC. We also generate a classification report and confusion matrix. The results regarding the accuracy metric are presented in Table 14, while the F1 and MCC scores are given in Appendix B. The confusion matrix and classification report are presented in Figure 7 and Table 18, respectively. The confusion matrix and classification report are generated for the XLex+LM model achieving the highest accuracy across the experiments performed. As shown later in the paper, the XLex+LM model achieves its highest accuracy of 84.3% when constructed with the *nasdaq* dataset as its source dataset and evaluated on the *financial_phrase_bank* dataset.

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. In addition, the evaluation datasets do not include any sentences that are present in the source datasets.

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. The label "Evaluation" in the "Purpose" column in Table 12 denotes that the corresponding dataset is used to evaluate the generated explainable lexicons. Details about generating the explainable lexicons from these datasets are shown as follows.

The datasets utilized in this study were primarily obtained from Kaggle, with the exception of the SemEval-2017-Task5 dataset, which was accessed from the official page of the SemEval competition. For extracting words with SHAP, we conducted a thorough search on Kaggle to find suitable

TABLE 11. Descriptions of the datasets that are used in the evaluation of XLex methodology.

Dataset	Description
sentfin	Sentfin: Dataset of financial news with entity-sentiment annotations
fiqa_labeled_df	Fiqa: Aspect-based dataset of financial sentences
sem_eval	Financially relevant news headlines annotated for fine-grained sentiment
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 evaluation of XLex methodology. The number of positive, negative, and neutral sentences, as well as the purpose of the datasets (used as a source or for evaluation), are shown.

Label	Total number of sentences	Positive	Negative	Neutral	Purpose
fiqa_labeled_df	201	139	62	0	Evaluation
sem_eval	353	270	83	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

financial-related datasets comprising textual statements, sentences, or news headlines for sentiment analysis. Our selection criteria included datasets studied in the literature or containing relevant data about companies listed on the stock market, ensuring diverse sources for extracting positive and negative words for building the explainable lexicon. As for the evaluation datasets, we not only considered financial textual data but also ensured that they were appropriately annotated by financial experts. This selection process was implemented to ensure the validity and robustness of these datasets for evaluation purposes.

B. GENERATED LEXICONS

We generate three different explainable lexicons. The words in the lexicons are generated from three different sources. The datasets serving as 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).

In Table 12, the Sentfin dataset is denoted as a "Source" dataset because it is used only for the purposes of word extraction. The Sentfin dataset comprises headlines and corresponding sentiment labels for the financial entities mentioned in the headlines. Each financial entity in a headline is assigned a sentiment label. However, the dataset lacks sentiment labels specifically for the headlines themselves, which renders it unsuitable for evaluation purposes. Nonetheless, we utilize this dataset as a "Source" since the sentiment labels for the headlines are not required for our word extraction and classification process.

We also want to evaluate the effectiveness of the methodology to label the words with the appropriate sentiments automatically. Summary data of the explainable lexicons is shown in Table 13, which also gives information about the

TABLE 13. Statistics of the lexicons on which sentiment analysis is performed. The lexicons are obtained using the RoBERTa transformer model.

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

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 two parameters that can be fine-tuned: decision coefficients and features that will be used to make sentiment decisions. We perform a grid search to find the optimal values of the model parameters that maximize the accuracy, F1 and MCC.

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 revealed that $SHAP_{avg}$ has the dominant impact on accuracy, F1, and MCC, and we, therefore, selected it as the primary decision feature. Then we performed a second grid search by using both the standard (without normalization) and normalized versions of the explainable lexicons in order to find the optimal values of the coefficients c_{xlp}, c_{xlo}, c_{lmp} and c_{lmo}. We chose 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 three coefficients (we exclude the c_{lmo} coefficient as obsolete because there is no word shared between

positive and negative words in the LM dictionary; thus, there are no ''opposite'' words in the LM dictionary). Thus, the total number of permutations is $5^3 = 125$. These permutations are combined with each of the three explainable lexicons (using both standard and normalized versions of the lexicons), and each of the four evaluation datasets. The only exceptions are 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 2750 models that are created for the purpose of grid search (125 permutations $\times$ 2 explainable lexicons $\times$ 2 lexicon versions $\times$ 4 evaluation datasets + 125 permutations $\times$ 1 explainable lexicon $\times$ 2 lexicon versions $\times$ 3 evaluation datasets). For each of the models in the grid search set of models, we are using the $SHAP_{avg}$ feature as the decision maker.

After obtaining the results, our primary goal is to identify the combination of coefficients that yields the highest aggregated average across the accuracy, F1, and MCC metrics. To calculate this aggregated average, we first determine the average values for accuracy, F1, and MCC scores across all experiments. We compute these average values for each combination of coefficients, considering both the explainable lexicons XLex and XLex+LM obtained from the three distinct source datasets (*nasdaq*, *fpb*, and *sentfin*). This results in a total of $3 \times 2 \times 3 = 18$ average values for each coefficient combination. The aggregated average for a specific combination of coefficients is obtained by summing these 18 average values, allowing us to represent each coefficient combination using a single consolidated parameter.

Through the grid search procedure, we discovered that the coefficients $(c_{xlp}, c_{xlo}, c_{lmp}, c_{lmo}) = (0.3, 0.1, 0.1, 0.5)$ form the combination that achieves the highest aggregated average, thus defining the optimal model parameters. It is worth noting that the choice of the value for the c_{lmo} coefficient does not impact the grid search procedure, as there are no ''opposite'' words in the LM dictionary. Hence, we can safely assume that $c_{lmo} = 0.5$ without affecting the outcome of the grid search.

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 datasets in Table 12 (these are the datasets containing the label ''Evaluation'' in the ''Purpose'' column). Only the model that 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 about accuracy are shown in Table 14a. Additional classification metrics, such as the F1 and MCC scores, are given in Tables 23-24 in Appendix B.

D. DISCUSSION

Table 14a reveals that the model achieves overall best accuracy 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 0.843 (column 6 in Table 14a); 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 dataset), the obtained accuracy evaluates to 0.837 (column 5 in Table 14a). For the same experiment, the accuracy evaluates to 0.303 (column 4 in Table 14a) if only the LM lexicon is taken as a basis for sentiment classification. The reason for this result is the insufficient word coverage of the LM lexicon since it does not contain the words that make up a large part of the sentences of the evaluation dataset. Therefore, those expressions remain 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.

As evidenced by the data presented in Table 14, it can be observed that XLex consistently surpasses LM in all experiments, resulting in an overall increase of 0.431 in terms of classification accuracy. This improvement remains evident when we extend LM with XLex. The combined XLex+LM dictionary leads to an overall 0.450 increase in accuracy over LM. Moreover, as observed by Tables 23-24 in Appendix B, it is noteworthy to highlight that the XLex+LM model consistently outperforms the LM model in terms of both F1 and MCC scores across all conducted experiments. Notably, the explainable lexicon XLex alone exhibits improvements over LM, leading to a 0.155 increase in F1 and a 0.090 enhancement in MCC. Even higher are the results achieved by the combined lexicon, XLex+LM, which demonstrates an increase of 0.226 in F1 and a 0.190 rise in MCC compared to LM. The enhancements in performance are determined by computing the difference between the average of the respective metric (accuracy, F1, or MCC) for XLex (XLex+LM) and the same metric averaged for LM. The metric's average is derived by averaging its values across all experiments. Table 16 consolidates the performance enhancements of XLex and XLex+LM over LM in terms of accuracy, F1, and MCC.

To test XLex methodology's effectiveness in the worst-case scenario, we conducted an evaluation only on portions of the datasets where we have a recall from LM (i.e., filtering

TABLE 14. Accuracy obtained using the XLex methodology based on the RoBERTa and FinBERT transformer models. The columns "LM", "XLex", and "XLex+LM" show the accuracy of the models on the corresponding dataset. The columns "LM on LM", "XLex on LM", "(XLex+LM) on LM" show the accuracy on portions of the datasets that have a recall from LM (i.e., all instances where the LM either provided no answer or were unable to make a decision are removed from the datasets). The approach with the highest accuracy among the three approaches is represented in bold.

(a) RoBERTa-based results

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	financial_phrase_bank	0.303	0.836	**0.843**	0.753	0.784	**0.801**
nasdaq	Yes	fiqa_labeled_df	0.313	0.721	**0.731**	0.808	0.795	**0.821**
nasdaq	Yes	fpb_fiqa	0.296	0.761	**0.776**	0.748	0.726	**0.766**
nasdaq	Yes	sem_eval	0.275	0.745	**0.765**	0.752	0.721	**0.775**
nasdaq	No	financial_phrase_bank	0.303	0.837	**0.843**	0.753	0.781	**0.795**
nasdaq	No	fiqa_labeled_df	0.313	0.697	**0.706**	0.808	0.795	**0.821**
nasdaq	No	fpb_fiqa	0.296	0.75	**0.768**	0.748	0.72	**0.764**
nasdaq	No	sem_eval	0.275	0.745	**0.756**	0.752	0.729	**0.76**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.303	**0.808**	**0.808**	0.753	**0.801**	**0.801**
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.313	0.697	**0.711**	0.808	0.808	**0.846**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.296	0.733	**0.746**	0.748	0.748	**0.779**
fiqa_fpb_sentfin_neutral	Yes	sem_eval	0.275	0.725	**0.756**	0.752	0.69	**0.775**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.303	0.81	**0.811**	0.753	0.798	**0.801**
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.313	0.672	**0.692**	0.808	0.795	**0.846**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.296	0.733	**0.746**	0.748	0.744	**0.777**
fiqa_fpb_sentfin_neutral	No	sem_eval	0.275	0.722	**0.759**	0.752	0.674	**0.775**
financial_phrase_bank	Yes	fiqa_labeled_df	0.313	0.632	**0.682**	0.808	0.731	**0.859**
financial_phrase_bank	Yes	fpb_fiqa	0.296	0.676	**0.707**	0.748	0.687	**0.764**
financial_phrase_bank	Yes	sem_eval	0.275	0.671	**0.697**	0.752	0.698	**0.767**
financial_phrase_bank	No	fiqa_labeled_df	0.313	0.662	**0.692**	0.808	0.744	**0.821**
financial_phrase_bank	No	fpb_fiqa	0.296	0.691	**0.716**	0.748	0.7	**0.762**
financial_phrase_bank	No	sem_eval	0.275	0.671	**0.694**	0.752	0.705	**0.767**
Average accuracy			0.296	0.727	**0.746**	0.766	0.744	**0.793**

(b) FinBERT-based results

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	financial_phrase_bank	0.303	0.768	**0.779**	0.753	0.736	**0.761**
nasdaq	Yes	fiqa_labeled_df	0.313	0.761	**0.771**	0.808	0.833	**0.859**
nasdaq	Yes	fpb_fiqa	0.296	0.747	**0.752**	0.748	0.754	**0.767**
nasdaq	Yes	sem_eval	0.275	0.725	**0.728**	0.752	0.775	**0.783**
nasdaq	No	financial_phrase_bank	0.303	0.694	**0.722**	0.753	0.694	**0.764**
nasdaq	No	fiqa_labeled_df	0.313	**0.761**	**0.761**	0.808	**0.846**	**0.846**
nasdaq	No	fpb_fiqa	0.296	0.709	**0.724**	0.748	0.73	**0.767**
nasdaq	No	sem_eval	0.275	0.714	**0.722**	0.752	0.767	**0.791**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.303	0.831	**0.836**	0.753	0.756	**0.77**
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.313	0.791	**0.811**	0.808	0.795	**0.846**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.296	0.78	**0.796**	0.748	0.736	**0.777**
fiqa_fpb_sentfin_neutral	Yes	sem_eval	0.275	0.756	**0.785**	0.752	0.721	**0.798**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.303	0.816	**0.823**	0.753	0.753	**0.77**
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.313	0.786	**0.801**	0.808	0.795	**0.833**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.296	0.774	**0.791**	0.748	0.736	**0.779**
fiqa_fpb_sentfin_neutral	No	sem_eval	0.275	0.756	**0.785**	0.752	0.729	**0.806**
financial_phrase_bank	Yes	fiqa_labeled_df	0.313	0.751	**0.771**	0.808	0.782	**0.833**
financial_phrase_bank	Yes	fpb_fiqa	0.296	0.792	**0.799**	0.748	0.749	**0.766**
financial_phrase_bank	Yes	sem_eval	0.275	0.734	**0.776**	0.752	0.698	**0.814**
financial_phrase_bank	No	fiqa_labeled_df	0.313	0.741	**0.751**	0.808	0.795	**0.821**
financial_phrase_bank	No	fpb_fiqa	0.296	0.778	**0.788**	0.748	0.743	**0.767**
financial_phrase_bank	No	sem_eval	0.275	0.739	**0.776**	0.752	0.705	**0.806**
Average accuracy			0.296	0.759	**0.775**	0.766	0.756	**0.797**

out all instances where the LM either provided no answer or were unable to make a decision). This creates a dataset with a strong bias in favor of LM, ultimately resulting in higher classification accuracy than the case when using LM on the whole dataset. For example, LM did not provide answers for 522 out of 885 sentences in the *financial_phrase_bank* dataset. This means that the evaluation, in this case, was conducted on only 363 sentences, effectively reducing the

TABLE 15. Number of sentences on which the models based on a given lexicon did not give answers. The results are obtained based on the lexicons generated using the RoBERTa-based transformer model.

Evaluation dataset	Number of sentences	Lexicons			
		fiqa_fpb_sentfin	nasdaq	financial_phrase_bank	Loughran-McDonald
fiqa_labeled_df	201	3	2	6	130
sem_eval	353	1	0	2	224
financial_phrase_bank	885	0	0	/	522
fpb_fiqa	1542	4	3	8	937

original dataset by 59% (Table 15 illustrates the reduction in the size of the evaluation datasets).

On this heavily constrained dataset towards LM, first, we tested the accuracy of the LM dictionary, and the results are shown in the ''LM on LM'' column in Table 14a. Then, we applied the XLex on this LM-contained dataset, and we obtained slightly worse results (column ''XLex on LM''). The accuracy decreased by only 1% in the case of the FinBERT-based model and by 2.2% for the RoBERTa-based model. This experiment indicates that despite the fact that XLex is trained on a general dataset and is automatically created, it produces comparable results to the state-of-the-art expert-annotated dictionary in this worst-case scenario.

Furthermore, we wanted to explore if XLex could be used to extend LM in this worst-case scenario, so we evaluated the performance of the combined dictionary XLex+LM on the LM-constrained datasets. Our findings reveal that the combined dictionary always leads to improvement of the results (column ''(XLex+LM) on LM'' in Table 14a). The average accuracy increased by 3% for the FinBERT-based model and 2.65% for the RoBERTa-based model. These results show that the proposed methodology can also be effectively used as an automated dictionary enhancement methodology that can help in extending the expert annotated dictionaries.

To gain a deeper understanding of the decision-making processes of XLex and LM, we present a detailed analysis of several text instances. We will use the test setup involving the standard XLex-based model (without normalization) in combination with *nasdaq* and *financial_phrase_bank* used as the source and evaluation dataset. As expected, our observations revealed that LM's errors stem from its insufficient word coverage. For instance, in the sentences *''Finnish engineering and technology company Metso Oyj said on May 27, 2008, it completed the acquisition of paper machinery technology from Japanese engineering company Mitsubishi Heavy Industries (MHI) for an undisclosed sum''* and *''Nokia also noted the average selling price of handsets declined during the period, though its mobile phone profit margin rose to more than 22 percent from 13 percent in the year-ago quarter''*, LM incorrectly places emphasis solely on the words ''undisclosed'' and ''decline'', respectively. LM classifies both words with a negative sentiment, disregarding the rest of the words in the respective sentences, which leads to inaccurate predictions. In contrast, XLex exhibits a larger vocabulary coverage and can assign sentiment scores to other relevant words in these sentences, resulting in accurate predictions. We also analyzed

TABLE 16. Average improvements of XLex and XLex+LM over LM in terms of accuracy, F1, and MCC. The values represent the differences in average metric scores (accuracy, F1, or MCC) between XLex (XLex+LM) and LM, both calculated by averaging the values of the corresponding metrics across all experiments.

	Improvement over LM		
Model	Acc	F1	MCC
XLex (RoBERTa-based)	0.431	0.155	0.090
XLex+LM (RoBERTa-based)	0.450	0.226	0.190
XLex (FinBERT-based)	0.463	0.227	0.183
XLex+LM (FinBERT-based)	0.479	0.250	0.253

cases where XLex made errors while LM produced correct predictions, such as the sentences *''Finnish airline Finnair is starting the temporary layoffs of cabin crews in February 2010''* and *''The financial impact is estimated to be an annual improvement of EUR 2.0 m in the division's results, as of fiscal year 2008''*. In these cases, LM correctly classified the words ''layoffs'' and ''improvement'', respectively, which had a dominant impact on the sentiment classification for the two sentences. While XLex formed its decision score on more words, its classification ultimately resulted in an inaccurate prediction. However, for these cases, the combined lexicon XLex+LM resulted in accurate predictions.

It is worth mentioning that we explored two distinct approaches in our initial experimental setup, one utilizing FinBERT and the other employing RoBERTa. FinBERT is a pre-trained large language model specifically designed for financial text analysis [68]. It is based on the BERT architecture and trained on a large corpus of financial text data to better understand and analyze financial language and documents. It has been reported that FinBERT performs well in financial applications compared to general-purpose language models. In our analysis, the FinBERT model demonstrated similar performance to RoBERTa with a slight accuracy improvement when utilizing XLex and XLex+LM. We want to note that we have made a deliberate choice to base our main analysis on RoBERTa, as FinBERT is fine-tuned on a closed proprietary dataset. In contrast, our model (using RoBERTa) is fine-tuned on publicly available datasets, enabling us to exercise precise control over the fine-tuning process while still achieving satisfactory results. All the details regarding the RoBERTa and FinBERT-based models are shown in Table 14, Table 16, Table 17 and Appendix B.[6] The results for FinBERT were obtained

[6]The source code and results about the comparison between RoBERTa and FinBERT can be found at: https://github.com/hristijanpeshov/SHAP-Explainable-Lexicon-Model/tree/master/notebooks

TABLE 17. Comparison of the XLex-based model with the RoBERTa and FinBERT transformer models in terms of model speed and size. The XLex-based model utilizes various source datasets and undergoes evaluation across different evaluation datasets. The execution speed of the models is assessed in a CPU environment available within the free tier of Google Colab to ensure a fair comparison under identical conditions. The CPU environment uses an Intel Xeon CPU with one physical and two logical cores running at 2.20GHz, equipped with 12GB of RAM.

Model	Source dataset	Is normalized?	Num. of words in source dataset	Eval dataset	Num. of sentences in eval dataset	Model size	Processing time (CPU) in seconds
XLex-based model	nasdaq	No	7094	fpb_fiqa	1542	363 KB	11.48
	nasdaq	No	7094	fiqa_labeled_df	201	363 KB	1.44
	nasdaq	No	7094	financial_phrase_bank	885	363 KB	7.47
	nasdaq	No	7094	sem_eval	353	363 KB	1.56
	nasdaq	Yes	7094	fpb_fiqa	1542	363 KB	11.68
	nasdaq	Yes	7094	fiqa_labeled_df	201	363 KB	1.44
	nasdaq	Yes	7094	financial_phrase_bank	885	363 KB	7.48
	nasdaq	Yes	7094	sem_eval	353	363 KB	1.59
	financial_phrase_bank	No	4344	fpb_fiqa	1542	202 KB	11.71
	financial_phrase_bank	No	4344	fiqa_labeled_df	201	202 KB	1.21
	financial_phrase_bank	No	4344	sem_eval	353	202 KB	1.46
	financial_phrase_bank	Yes	4344	fpb_fiqa	1542	202 KB	11.61
	financial_phrase_bank	Yes	4344	fiqa_labeled_df	201	202 KB	1.44
	financial_phrase_bank	Yes	4344	sem_eval	353	202 KB	1.44
	fiqa_fpb_sentfin_neutral	No	4868	fpb_fiqa	1542	233 KB	11.7
	fiqa_fpb_sentfin_neutral	No	4868	fiqa_labeled_df	201	233 KB	1.29
	fiqa_fpb_sentfin_neutral	No	4868	financial_phrase_bank	885	233 KB	7.53
	fiqa_fpb_sentfin_neutral	No	4868	sem_eval	353	233 KB	1.6
	fiqa_fpb_sentfin_neutral	Yes	4868	fpb_fiqa	1542	233 KB	11.74
	fiqa_fpb_sentfin_neutral	Yes	4868	fiqa_labeled_df	201	233 KB	1.14
	fiqa_fpb_sentfin_neutral	Yes	4868	financial_phrase_bank	885	233 KB	7.51
	fiqa_fpb_sentfin_neutral	Yes	4868	sem_eval	353	233 KB	1.6
RoBERTa transformer model	—	—	—	fpb_fiqa	1542	1.32 GB	960.05
	—	—	—	fiqa_labeled_df	201	1.32 GB	111.5
	—	—	—	financial_phrase_bank	885	1.32 GB	600.12
	—	—	—	sem_eval	353	1.32 GB	183.45
FinBERT transformer model	—	—	—	fpb_fiqa	1542	417.8 MB	232.26
	—	—	—	fiqa_labeled_df	201	417.8 MB	27.95
	—	—	—	financial_phrase_bank	885	417.8 MB	143.99
	—	—	—	sem_eval	353	417.8 MB	38.02

following the identical grid search procedure as that applied to RoBERTa.

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 doing at all in a system, it is worth doing it fast [73].

Table 17 shows a comparative analysis of model sizes and sentiment classification execution times for three models: the XLex-based model, featured in Section V, the fine-tuned RoBERTa transformer model, presented in Section III, and the FinBERT model. The analysis involves conducting experiments using the XLex-based model with lexicons learned from various source datasets, assessing its performance in sentiment classification across different evaluation datasets. We explore both normalized and non-normalized versions of the lexicons. In contrast, the RoBERTa and FinBERT-based models do not rely on source datasets by design, as they are pre-trained models that can be readily used for sentiment classification. Consequently, the corresponding entries in columns 2-4 of Table 17 remain empty.

The execution speed of the models is evaluated across the evaluation datasets using a central processing unit (CPU) in Google Colab to ensure a fair comparison under identical conditions. The execution time of each model is determined by calculating the average of the times recorded from 10 experimental runs. For the experiments, we used Google Colab's free tier, which provides an Intel Xeon CPU with one physical and two logical cores running at 2.20GHz paired with 12GB of RAM. As can be seen in Table 17, the results reveal a substantial difference in the execution time of the models. The XLex-based model leads to a significantly smaller execution time compared to the RoBERTa and FinBERT transformer models by a factor of about 87 and 21, respectively. The factor is determined by dividing the average CPU speed of the RoBERTa (FinBERT) model by that of the XLex-based model (the averages are calculated considering the respective experiments for each of the two models). 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.

Similar to other neural network models, RoBERTa can leverage parallel architectures to enhance its processing speed. Employing RoBERTa on a GPU, as opposed to a CPU, yields a substantial reduction in execution time, achieving an overall speedup factor of approximately 26. The GPU tests were performed on the NVIDIA Tesla T4 GPU computing environment available within Google Colab's free tier. Although the XLex-based model can be parallelized, its current implementation lacks the necessary capabilities for parallel processing. As a result, evaluating its performance in a GPU environment would lead to an unfair comparison. Consequently, Table 17 presents results only for the CPU comparison. The parallelization of the XLex-based model goes beyond the scope of this paper; however, it can serve as a potential avenue for future work.

In the GPU environment, we perform the word extraction process using SHAP and transformer models. The word extraction process with the RoBERTa and FinBERT models

TABLE 18. Classification report showcasing the performance of the XLex-based model (using RoBERTa) based on the combined XLex+LM lexicon. The model is constructed using the lexicon created with the *nasdaq* dataset as its source dataset. The model evaluation is performed on the *financial_phrase_bank* dataset. This model achieves the highest accuracy among all XLex+LM models tested, achieving an accuracy rate of 84.3%.

	precision	recall	f1-score	support
Negative sentences	0.43	0.72	0.54	111
Positive sentences	0.96	0.86	0.91	774
accuracy			0.84	885
macro avg	0.69	0.79	0.72	885
weighted avg	0.89	0.84	0.86	885

on 9202 sentences from the *nasdaq* cf dataset took approximately 23 and 10 hours, respectively. The extended time required for word extraction is primarily attributed to the slow performance and time-consuming nature of SHAP's operations.

The XLex methodology provides flexibility in selecting the underlying transformer model, allowing for the easy integration of any preferred model. Thus, we were also interested in assessing the speed performance of the FinBERT model. We conducted tests in the same CPU environment. Table 17 shows that the FinBERT model leads to better execution speeds than the RoBERTa model. However, the FinBERT model exhibits a smaller size compared to RoBERTa. Similarly to RoBERTa, the FinBERT model also underwent testing on a GPU, resulting in a nearly 12-fold improvement in execution speed compared to running it on a CPU. As expected, the XLex-based model outperforms the FinBERT model in terms of speed.

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. 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 17. The difference in size is considerable. As can be seen, the lexicon-based model is about three orders of magnitude smaller than the RoBERTa transformer model. The XLex-based model exhibits an identical size for each specific source dataset, with the XLex-based model registering sizes of 363KB, 202KB, and 233KB for the *nasdaq*, *financial_phrase_bank* (*fpb*), and *fiqa_fpb_sentfin_neutral* source datasets, respectively. Additionally, the RoBERTa-based model maintains a size of 1.32GB, regardless of the evaluation dataset used. The size of the FinBERT model is 417.8MB. Although there are approaches to make transformer-based models smaller [74], transformers are not suitable for certain use cases, such as environments with limited computational resources or embedded devices. On the other hand, Table 17 shows that

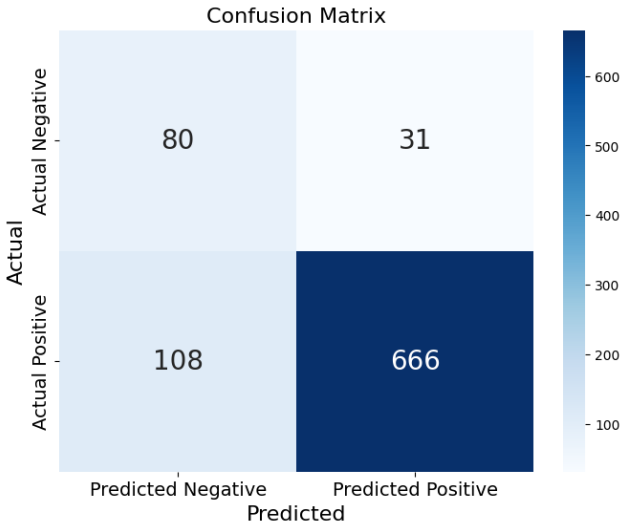

FIGURE 7. Confusion matrix showcasing the performance of the XLex-based model (using RoBERTa) based on the combined XLex+LM lexicon. The model is constructed using the lexicon created with the *nasdaq* dataset as its source dataset. The model evaluation is performed on the *financial_phrase_bank* dataset. This model achieves the highest accuracy among all XLex+LM models tested, achieving an accuracy rate of 84.3%.

lexicon-based models have a size that is suitable for such applications.

Another 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 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. 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 is 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 is 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 present a novel XLex methodology that leverages 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 employ a multi-faceted validation strategy that integrates multiple datasets. Specifically, we learn the lexicon on one dataset and subsequently test its effectiveness on various other datasets. We have conducted 22 separate experiments, and in all of them, the proposed XLex methodology leads to increased performance compared to LM. Specifically, we evaluated the XLex methodology in two separate instances: one employing a fine-tuned RoBERTa-based model and the other utilizing a pre-trained FinBERT model. The results yielded are largely comparable while also emphasizing the robustness of the XLex methodology and its effectiveness in working with various transformer models.

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, particularly 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.

Our study highlights the performance improvements of XLex, achieved even with a simple optimization algorithm based on a grid search procedure characterized by a relatively low granularity. One avenue for future work is to improve the grid search by increasing granularity or incorporating more advanced techniques, such as Bayesian optimization. This would enable a more precise discovery of optimal model parameters, potentially resulting in additional gains in accuracy, F1, and MCC. Furthermore, the parallelization of the XLex-based model could serve as a viable direction for future research given its potential to further increase the computational efficiency of XLex.

Additionally, 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, offering opportunities for future research to explore its application across other domains beyond finance. Adopting the XLex methodology for different domains 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

APPENDIX B

TABLE B.1. F1 score obtained across different experiments using the XLex methodology based on the RoBERTa transformer model. The highest values are represented in bold.

Source of the lexicon	Normalized	Evaluation set	F1 on whole dataset			F1 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	financial_phrase_bank	0.287	0.451	**0.714**	0.688	0.448	**0.731**
nasdaq	Yes	fiqa_labeled_df	0.297	0.457	**0.473**	0.789	0.739	**0.802**
nasdaq	Yes	fpb_fiqa	0.295	0.434	**0.466**	0.726	0.437	**0.739**
nasdaq	Yes	sem_eval	0.305	0.664	**0.721**	0.752	0.704	**0.775**
nasdaq	No	financial_phrase_bank	0.287	0.457	**0.72**	0.688	0.446	**0.726**
nasdaq	No	fiqa_labeled_df	0.297	0.437	**0.454**	0.789	0.739	**0.802**
nasdaq	No	fpb_fiqa	0.295	0.426	**0.459**	0.726	0.433	**0.738**
nasdaq	No	sem_eval	0.305	0.672	**0.713**	0.752	0.717	**0.759**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.287	0.451	**0.689**	0.688	0.474	**0.731**
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.297	0.434	**0.456**	0.789	0.752	**0.827**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.295	0.424	**0.448**	0.726	0.458	**0.751**
fiqa_fpb_sentfin_neutral	Yes	sem_eval	0.305	0.434	**0.476**	0.752	0.676	**0.775**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.287	0.448	**0.694**	0.688	0.463	**0.731**
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.297	0.416	**0.439**	0.789	0.747	**0.827**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.295	0.425	**0.448**	0.726	0.457	**0.748**
fiqa_fpb_sentfin_neutral	No	sem_eval	0.305	0.434	**0.48**	0.752	0.66	**0.775**
financial_phrase_bank	Yes	fiqa_labeled_df	0.297	0.409	**0.447**	0.789	0.688	**0.84**
financial_phrase_bank	Yes	fpb_fiqa	0.295	0.4	**0.43**	0.726	0.428	**0.739**
financial_phrase_bank	Yes	sem_eval	0.305	0.416	**0.445**	0.752	0.462	**0.767**
financial_phrase_bank	No	fiqa_labeled_df	0.297	0.43	**0.456**	0.789	0.692	**0.802**
financial_phrase_bank	No	fpb_fiqa	0.295	0.413	**0.439**	0.726	0.436	**0.738**
financial_phrase_bank	No	sem_eval	0.305	0.416	**0.444**	0.752	0.467	**0.767**

TABLE B.2. MCC obtained across different experiments using the XLex methodology based on the RoBERTa transformer model. The highest values are represented in bold.

Source of the lexicon	Normalized	Evaluation set	MCC on whole dataset			MCC 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	financial_phrase_bank	0.192	0.362	**0.453**	0.489	0.364	**0.539**
nasdaq	Yes	fiqa_labeled_df	0.216	0.372	**0.432**	0.594	0.492	**0.615**
nasdaq	Yes	fpb_fiqa	0.209	0.306	**0.418**	0.519	0.312	**0.524**
nasdaq	Yes	sem_eval	0.265	0.332	**0.477**	0.567	0.417	**0.601**
nasdaq	No	financial_phrase_bank	0.192	0.383	**0.471**	0.489	0.359	**0.531**
nasdaq	No	fiqa_labeled_df	0.216	0.313	**0.374**	0.594	0.492	**0.615**
nasdaq	No	fpb_fiqa	0.209	0.281	**0.399**	0.519	0.301	**0.525**
nasdaq	No	sem_eval	0.265	0.349	**0.465**	0.567	0.437	**0.578**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.192	0.392	**0.434**	0.489	0.46	**0.539**
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.216	0.303	**0.371**	0.594	0.523	**0.659**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.209	0.286	**0.376**	0.519	0.375	**0.54**
fiqa_fpb_sentfin_neutral	Yes	sem_eval	0.265	0.308	**0.466**	0.567	0.355	**0.593**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.192	0.376	**0.443**	0.489	0.412	**0.539**
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.216	0.249	**0.32**	0.594	0.499	**0.659**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.209	0.288	**0.373**	0.519	0.373	**0.531**
fiqa_fpb_sentfin_neutral	No	sem_eval	0.265	0.31	**0.481**	0.567	0.322	**0.593**
financial_phrase_bank	Yes	fiqa_labeled_df	0.216	0.25	**0.363**	0.594	0.375	**0.683**
financial_phrase_bank	Yes	fpb_fiqa	0.209	0.245	**0.351**	0.519	0.303	**0.528**
financial_phrase_bank	Yes	sem_eval	0.265	0.296	**0.422**	0.567	0.386	**0.581**
financial_phrase_bank	No	fiqa_labeled_df	0.216	0.306	**0.398**	0.594	0.385	**0.615**
financial_phrase_bank	No	fpb_fiqa	0.209	0.284	**0.383**	0.519	0.325	**0.529**
financial_phrase_bank	No	sem_eval	0.265	0.296	**0.418**	0.567	0.4	**0.581**

TABLE B.3. F1 score obtained across different experiments using the XLex methodology based on the FinBERT transformer model. The highest values are represented in bold.

Source of the lexicon	Normalized	Evaluation set	F1 on whole dataset			F1 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	financial_phrase_bank	0.287	0.421	**0.432**	0.688	0.441	**0.694**
nasdaq	Yes	fiqa_labeled_df	0.297	0.497	**0.505**	0.789	0.802	**0.836**
nasdaq	Yes	fpb_fiqa	0.295	0.448	**0.458**	0.726	0.477	**0.74**
nasdaq	Yes	sem_eval	0.305	0.684	**0.693**	0.752	0.774	**0.783**
nasdaq	No	financial_phrase_bank	0.287	0.389	**0.408**	0.688	0.419	**0.696**
nasdaq	No	fiqa_labeled_df	0.297	0.498	**0.5**	0.789	0.815	**0.823**
nasdaq	No	fpb_fiqa	0.295	0.428	**0.442**	0.726	0.464	**0.74**
nasdaq	No	sem_eval	0.305	0.676	**0.692**	0.752	0.766	**0.791**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.287	0.708	**0.724**	0.688	0.673	**0.702**
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.297	0.522	**0.537**	0.789	0.765	**0.823**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.295	0.462	**0.484**	0.726	0.458	**0.75**
fiqa_fpb_sentfin_neutral	Yes	sem_eval	0.305	0.69	**0.741**	0.752	0.715	**0.798**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.287	0.696	**0.711**	0.688	0.673	**0.702**
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.297	0.518	**0.53**	0.789	0.765	**0.811**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.295	0.461	**0.483**	0.726	0.459	**0.751**
fiqa_fpb_sentfin_neutral	No	sem_eval	0.305	0.696	**0.742**	0.752	0.724	**0.806**
financial_phrase_bank	Yes	fiqa_labeled_df	0.297	0.486	**0.511**	0.789	0.471	**0.814**
financial_phrase_bank	Yes	fpb_fiqa	0.295	0.461	**0.486**	0.726	0.453	**0.739**
financial_phrase_bank	Yes	sem_eval	0.305	0.432	**0.488**	0.752	0.452	**0.814**
financial_phrase_bank	No	fiqa_labeled_df	0.297	0.476	**0.496**	0.789	0.479	**0.802**
financial_phrase_bank	No	fpb_fiqa	0.295	0.448	**0.477**	0.726	0.446	**0.74**
financial_phrase_bank	No	sem_eval	0.305	0.429	**0.488**	0.752	0.452	**0.806**

TABLE B.4. MCC obtained across different experiments using the XLex methodology based on the FinBERT transformer model. The highest values are represented in bold.

Source of the lexicon	Normalized	Evaluation set	MCC on whole dataset			MCC 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	financial_phrase_bank	0.192	0.315	**0.352**	**0.489**	0.414	**0.489**
nasdaq	Yes	fiqa_labeled_df	0.216	0.497	**0.525**	0.594	0.604	**0.673**
nasdaq	Yes	fpb_fiqa	0.209	0.375	**0.415**	0.519	0.451	**0.526**
nasdaq	Yes	sem_eval	0.265	0.417	**0.452**	0.567	0.565	**0.613**
nasdaq	No	financial_phrase_bank	0.192	0.279	**0.326**	0.489	0.378	**0.493**
nasdaq	No	fiqa_labeled_df	0.216	0.503	**0.516**	0.594	0.632	**0.648**
nasdaq	No	fpb_fiqa	0.209	0.334	**0.381**	0.519	0.425	**0.526**
nasdaq	No	sem_eval	0.265	0.415	**0.465**	0.567	0.546	**0.624**
fiqa_fpb_sentfin_neutral	Yes	financial_phrase_bank	0.192	0.454	**0.494**	0.489	0.418	**0.5**
fiqa_fpb_sentfin_neutral	Yes	fiqa_labeled_df	0.216	0.565	**0.608**	0.594	0.53	**0.648**
fiqa_fpb_sentfin_neutral	Yes	fpb_fiqa	0.209	0.394	**0.47**	0.519	0.386	**0.54**
fiqa_fpb_sentfin_neutral	Yes	sem_eval	0.265	0.39	**0.511**	0.567	0.43	**0.636**
fiqa_fpb_sentfin_neutral	No	financial_phrase_bank	0.192	0.441	**0.478**	0.489	0.424	**0.5**
fiqa_fpb_sentfin_neutral	No	fiqa_labeled_df	0.216	0.557	**0.592**	0.594	0.53	**0.624**
fiqa_fpb_sentfin_neutral	No	fpb_fiqa	0.209	0.398	**0.472**	0.519	0.392	**0.543**
fiqa_fpb_sentfin_neutral	No	sem_eval	0.265	0.407	**0.517**	0.567	0.451	**0.648**
financial_phrase_bank	Yes	fiqa_labeled_df	0.216	0.456	**0.531**	0.594	0.465	**0.637**
financial_phrase_bank	Yes	fpb_fiqa	0.209	0.382	**0.468**	0.519	0.361	**0.524**
financial_phrase_bank	Yes	sem_eval	0.265	0.297	**0.483**	0.567	0.371	**0.653**
financial_phrase_bank	No	fiqa_labeled_df	0.216	0.429	**0.489**	0.594	0.5	**0.615**
financial_phrase_bank	No	fpb_fiqa	0.209	0.346	**0.443**	0.519	0.341	**0.523**
financial_phrase_bank	No	sem_eval	0.265	0.288	**0.483**	0.567	0.388	**0.648**

REFERENCES

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

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

[3] N. Mohamed and J. Al-Jaroodi, "Real-time big data analytics: Applications and challenges," in *Proc. Int. Conf. High Perform. Comput. Simulation (HPCS)*, Jul. 2014, pp. 305–310.

[4] V. Ravi and S. Kamaruddin, "Big data analytics enabled smart financial services: Opportunities and challenges," in *Proc. Int. Conf. Big Data Anal.*, Hyderabad, India, Dec. 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, Jun. 2015.

[6] J. Smailović, M. Grčar, N. Lavrač, and M. Žnidaršič, "Predictive sentiment analysis of tweets: A stock market application," in *Proc. Int. Workshop Hum.-Comput. Interact. Knowl. Discovery Complex, Unstructured, Big Data*, Maribor, Slovenia. Cham, Switzerland: Springer, 2013, pp. 77–88.

[7] A. Derakhshan and H. Beigy, "Sentiment analysis on stock social media for stock price movement prediction," *Eng. Appl. Artif. Intell.*, vol. 85, pp. 569–578, Oct. 2019.

[8] R. Ren, D. D. Wu, and T. Liu, "Forecasting stock market movement direction using sentiment analysis and support vector machine," *IEEE Syst. J.*, vol. 13, no. 1, pp. 760–770, Mar. 2019.

[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," *Int. J. Inf. Manage.*, vol. 50, pp. 452–462, Feb. 2020.

[10] F.-T. Tsai, H.-M. Lu, and M.-W. Hung, "The effects of news sentiment and coverage on credit rating analysis," in *Proc. Pacific Asia Conf. Inf. Syst. (PACIS)*, 2010. [Online]. Available: https://aisel.aisnet.org/pacis2010/199/

[11] S. Gül, Ö. Kabak, and I. Topcu, "A multiple criteria credit rating approach utilizing social media data," *Data Knowl. Eng.*, vol. 116, pp. 80–99, Jul. 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 Trans. Manage. Inf. Syst.*, vol. 3, no. 3, pp. 1–30, Oct. 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 *Proc. 48th Hawaii Int. Conf. Syst. Sci.*, Jan. 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. 161943–161957, 2020.

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

[16] L. Ziora, "The sentiment analysis as a tool of business analytics in contemporary organizations," *Studia Ekonomiczne*, vol. 281, pp. 234–241, Jan. 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," in *Sentiment Analysis and Ontology Engineering*. Cham, Switzerland: Springer, 2016, pp. 273–311.

[18] X. Tian, J. S. He, and M. Han, "Data-driven approaches in FinTech: A survey," *Inf. Discovery Del.*, vol. 49, no. 2, pp. 123–135, May 2021.

[19] C.-C. Chen, H.-H. Huang, and H.-H. Chen, "Fintech applications," in *From Opinion Mining to Financial Argument Mining*. Cham, Switzerland: 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 Trans. Computat. Social Syst.*, vol. 7, no. 6, pp. 1358–1375, Dec. 2020.

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

[22] A. D'Andrea, F. Ferri, P. Grifoni, and T. Guzzo, "Approaches, tools and applications for sentiment analysis implementation," *Int. J. Comput. Appl.*, vol. 125, no. 3, pp. 26–33, Sep. 2015.

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

[24] S. Taj, B. B. Shaikh, and A. F. Meghji, "Sentiment analysis of news articles: A lexicon based approach," in *Proc. 2nd Int. Conf. Comput., Math. Eng. Technol. (iCoMET)*, Jan. 2019, pp. 1–5.

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

[26] T. Loughran and B. McDonald, "When is a liability not a liability? textual analysis, dictionaries, and 10-ks," *J. 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," *Inf. Retr.*, vol. 12, no. 5, pp. 526–558, Oct. 2009.

[28] A. Sharma and S. Dey, "A comparative study of feature selection and machine learning techniques for sentiment analysis," in *Proc. ACM Res. Appl. Comput. Symp.*, Oct. 2012, pp. 1–7.

[29] S. Malviya, A. K. Tiwari, R. Srivastava, and V. Tiwari, "Machine learning techniques for sentiment analysis: A review," *SAMRIDDHI, J. Phys. Sci., Eng. Technol.*, vol. 12, no. 2, pp. 72–78, 2020.

[30] M. S. Neethu and R. Rajasree, "Sentiment analysis in Twitter using machine learning techniques," in *Proc. 4th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT)*, Jul. 2013, pp. 1–5.

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

[32] A. Yadav and D. K. Vishwakarma, "Sentiment analysis using deep learning architectures: A review," *Artif. Intell. Rev.*, vol. 53, no. 6, pp. 4335–4385, Aug. 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, Mar. 2020.

[34] D. Tang, B. Qin, and T. Liu, "Deep learning for sentiment analysis: Successful approaches and future challenges," *WIREs Data Mining Knowl. Discovery*, vol. 5, no. 6, pp. 292–303, Nov. 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," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

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

[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. 131662–131682, 2020.

[38] X. S. Huang, F. Perez, J. Ba, and M. Volkovs, "Improving transformer optimization through better initialization," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 4475–4483.

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

[40] T. Loughran and B. Mcdonald, "Textual analysis in accounting and finance: A survey," *J. Accounting Res.*, vol. 54, no. 4, pp. 1187–1230, Sep. 2016.

[41] P. C. Tetlock, "Giving content to investor sentiment: The role of media in the stock market," *J. Finance*, vol. 62, no. 3, pp. 1139–1168, 2007.

[42] C. Dougal, J. Engelberg, D. García, and C. A. Parsons, "Journalists and the stock market," *Rev. Financial Stud.*, vol. 25, no. 3, pp. 639–679, Mar. 2012.

[43] U. G. Gurun and A. W. Butler, "Don't believe the hype: Local media slant, local advertising, and firm value," *J. Finance*, vol. 67, no. 2, pp. 561–598, Apr. 2012.

[44] 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 *Proc. 15th Annu. Int. Conf. Comput. Sci. Educ. Comput. Sci.*, 2019, pp. 37–42.

[45] T. Loughran and B. McDonald, "Measuring readability in financial disclosures," *J. Finance*, vol. 69, no. 4, pp. 1643–1671, Aug. 2014.

[46] S. Krishnamoorthy, "Sentiment analysis of financial news articles using performance indicators," *Knowl. Inf. Syst.*, vol. 56, no. 2, pp. 373–394, Aug. 2018.

[47] P. J. Stone, D. C. Dunphy, and M. S. Smith, *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA, USA: MIT Press, 1966.

[48] S. M. Mohammad, S. Kiritchenko, and X. Zhu, "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets," 2013, *arXiv:1308.6242*.

[49] D. T. Vo and Y. Zhang, "Don't count, predict! An automatic approach to learning sentiment lexicons for short text," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Short Papers)*, vol. 2, 2016, pp. 219–224.

[50] F. Viegas, M. S. Alvim, S. Canuto, T. Rosa, M. A. Gonçalves, and L. Rocha, "Exploiting semantic relationships for unsupervised expansion of sentiment lexicons," *Inf. Syst.*, vol. 94, Dec. 2020, Art. no. 101606.

[51] T. Bos and F. Frasincar, "Automatically building financial sentiment lexicons while accounting for negation," *Cognit. Comput.*, vol. 14, no. 1, pp. 442–460, Jan. 2022.

[52] H. Saif, Y. He, M. Fernandez, and H. Alani, "Adapting sentiment lexicons using contextual semantics for sentiment analysis of Twitter," in *Proc. Eur. Semantic Web Conf.* Cham, Switzerland: Springer, 2014, pp. 54–63.

[53] N. Kaji and M. Kitsuregawa, "Building lexicon for sentiment analysis from massive collection of html documents," in *Proc. Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn. (EMNLP-CoNLL)*, 2007, pp. 1075–1083.

[54] H. Kanayama and T. Nasukawa, "Fully automatic lexicon expansion for domain-oriented sentiment analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2006, pp. 355–363.

[55] N. Kaji and M. Kitsuregawa, "Automatic construction of polarity-tagged corpus from HTML documents," in *Proc. COLING/ACL Main Conf. Poster Sessions*, 2006, pp. 452–459.

[56] W. L. Hamilton, K. Clark, J. Leskovec, and D. Jurafsky, "Inducing domain-specific sentiment lexicons from unlabeled corpora," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, p. 595.

[57] O. Araque, G. Zhu, and C. A. Iglesias, "A semantic similarity-based perspective of affect lexicons for sentiment analysis," *Knowl.-Based Syst.*, vol. 165, pp. 346–359, Feb. 2019.

[58] W. Zhao, T. Joshi, V. N. Nair, and A. Sudjianto, "SHAP values for explaining CNN-based text classification models," 2020, *arXiv:2008.11825*.

[59] K. E. Mokhtari, B. P. Higdon, and A. Başar, "Interpreting financial time series with Shap values," in *Proc. 29th Annu. Int. Conf. Comput. Sci. Softw. Eng.*, 2019, pp. 166–172.

[60] X. Xiaomao, Z. Xudong, and W. Yuanfang, "A comparison of feature selection methodology for solving classification problems in finance," *J. Phys., Conf. Ser.*, vol. 1284, no. 1, Aug. 2019, Art. no. 012026.

[61] 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. 97531–97554, 2022.

[62] E. Kokalj, B. Škrlj, N. Lavrač, S. Pollak, and M. Robnik-Šikonja, "Bert meets shapley: Extending shap explanations to transformer-based classifiers," in *Proc. EACL Hackashop News Media Content Anal. Automated Rep. Gener.*, 2021, pp. 16–21.

[63] S. Consoli, L. Barbaglia, and S. Manzan, "Fine-grained, aspect-based sentiment analysis on economic and financial lexicon," *Knowl.-Based Syst.*, vol. 247, Jul. 2022, Art. no. 108781.

[64] 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 *Proc. 12th Lang. Resour. Eval. Conf.*, 2020, pp. 5065–5072.

[65] M. Yekrangi and N. Abdolvand, "Financial markets sentiment analysis: Developing a specialized lexicon," *J. Intell. Inf. Syst.*, vol. 57, no. 1, pp. 127–146, Aug. 2021.

[66] J. Fang and B. Chen, "Incorporating lexicon knowledge into SVM learning to improve sentiment classification," in *Proc. Workshop Sentiment Anal. Where AI Meets Psychology (SAAIP)*, 2011, pp. 94–100.

[67] 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, Jan. 2022.

[68] A. H. Huang, H. Wang, and Y. Yang, "FinBERT: A large language model for extracting information from financial text," *Contemp. Accounting Res.*, vol. 40, no. 2, pp. 806–841, May 2023.

[69] 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 *Proc. IEEE 13th Int. Conf. Data Mining Workshops*, Dec. 2013, pp. 945–954.

[70] 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," in *Proc. 11th Int. Workshop Semantic Eval. (SemEval)*, 2017, pp. 519–535.

[71] S. Mazzanti. (2020). *Shap Values Explained Exactly How You Wished Someone Explained to You*. Accessed: Jul. 5, 2021. [Online]. Available: https://towardsdatascience.com/shap-explained-the-way-i-wish-someone-explained-it-to-me-ab81cc69ef30

[72] S. Lundberg. (2018). *SHapley Additive exPlanations*. Accessed: Jan. 29, 2023. [Online]. Available: https://github.com/slundberg/shap

[73] M. Honnibal and I. Montani. (2019). *spaCy Meets Transformers: Fine-Tune BERT, XLNet and GPT-2*. Accessed: Feb. 10, 2023. [Online]. Available: https://explosion.ai/blog/spacy-transformers

[74] N. Lathia. (2019). *When is a Neural Net Too Big for Production*. Accessed: Mar. 10, 2023. [Online]. Available: https://neal-lathia.medium.com/when-is-a-neural-net-too-big-for-production-4315452193ef

MARYAN RIZINSKI received the B.S. and M.S. degrees in electrical engineering and information technologies from Ss. Cyril and Methodius University in Skopje, where he is currently pursuing the Ph.D. degree in computer science. He is an Engineering Manager with Bosch, with over ten years of industry experience leading globally-distributed software engineering teams. He is also a Lecturer of computer science with Boston University's Metropolitan College, where he teaches and facilitates networking and data science classes. 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 the Internet of Things (IoT) and fiber-optics infrastructure projects and has been mentoring and consulting startup IT companies. 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 the IoT architectures.

HRISTIJAN PESHOV received the Bachelor of Science degree in software engineering and information systems from the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, in 2022. He is currently 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 Ss. Cyril and Methodius University in Skopje, Skopje, in 2013 and 2016, respectively, where he is currently pursuing the Ph.D. degree. He is a Teaching and Research Assistant with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University. His research interests include data science, natural language processing, semantic web, web technologies, and computer networks.

MILOS JOVANOVIK received the Ph.D. degree in the field of computer science and engineering from Ss. Cyril and Methodius University in Skopje, in 2016, with a Ph.D. thesis in the domain of linked data. He is currently an Associate Professor with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje. He is also a Senior Research and Development Knowledge Graphs Engineer with OpenLink Software, London, U.K. He has published over 50 scientific articles 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 the Ph.D. degree in computer science. From March 2011 to 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. He is currently a Full Professor with the Faculty of Computer Science and Engineering, Ss. Cyril and Methodius University in Skopje, and a Visiting Research Professor with Boston University. He 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. He is the author of more than 200 journal articles and conference papers, and seven books. He has been involved in more than 70 research and industry projects, of which more than 40 projects as a Project Leader. His research interests include data science, machine learning, NLP, FinTech, semantic web, e-commerce, technology for development, ESG, and climate change.