Sentiment Analysis Based Fuzzy Decision Platform for the Saudi Stock Market

Alshahrani Hasan A
Computer Science Department
Western Michigan University
Kalamazoo, USA
hasanayedh.alshahrani@wmich.edu

Alvis C. Fong
Computer Science Department
Western Michigan University
Kalamazoo, USA
alvis.fong@wmich.edu

***Abstract*—Investors in the Saudi stock market are used to seeking advice from online resources such as Twitter and discussion forums. In this paper, we introduced some components to enhance decision making using sentiment analysis and simple fuzzy decision. We built two corpora and one lexicon manually, and we did analysis on them using both corpus-based approach, and semantical and lexical based approach. The best model was selected to be the base of a fuzzy decision mechanism provided for investors. We mentioned several performance metrics, but the main metric to count on was recall. The best model was the rule-based approach with minimum and maximum recall of 69% and 96% respectively as we go through different data sets, types, and sizes.**

***Index Terms*—lexicon, sentiment, stock, semantical, fuzzy**

I. Introduction

One of the main applications of sentiment analysis is the prediction of stock market direction. Stock market prices are being discussed every day on microblogs, discussion forums, and stock markets websites; and a huge amount of text is produced everyday. In some research mentioned by [1], news and sentiments can drive the market significantly, and human decisions are affected by emotions and moods. According to a study done by [2] about the Saudi stock market, investors decisions can be irrational and cannot be described by the normal financial theories.

The Saudi investors, as other investors in other markets, tend to seek advice before investing in the market. In this paper, we applied sentiment analysis on a huge amount of documents (forum posts and tweets) to make investors able to predict the Saudi Stock Market (SSM) movement. As the decision of investment is critical we counted on manually constructed two corpora and one lexicon as the manual way is more reliable and trusted than the automatically constructed ones as stated by [3].

There are two main contributions for this paper. First, we introduced the following manually constructed objects: Saudi Stock Market Lexicon (SSML), an annotated Twitter corpus, and an annotated forum posts corpus. Second, we proposed a rule-based fuzzy decision approach based on the sentiment analysis of those three objects. We adopted two main analysis methods: corpus-based approach and semantical-lexical based approach. We used recall as the performance measure because we focus on one class only as we will see later in this paper, the class is either positive or negative as the customer might be interested in buying on a market dunk, or want to buy high and sell higher!! In other wards, using recall is answering the question: are we correctly classifying a good number of the relevant posts ? This number of classified posts can be divided by the total number of posts to know how positive/negative is the market. The real time prediction of the market is beyond the scope of this study.

The rest of this paper is organized as follows. Section 2 and section 3 talk about the related work and data collection respectively. Section 4 is about text preprocessing, section 5 gives details about SSML, and section 6 is the experiment and results. The decision making is discussed in section 7, and we conclude by section 8.

II. Related Work

A text-based decision support system was proposed by [4]. Using a collection of financial text documents, they extract all sequences of events from documents and infer any possible hidden relations between them. The system has four main parts: text processing unit, textual information generalization, event sequence extraction, and classifier-based inference engine. They used the decision tree classifier to make decisions. Online opinions about the stock market were utilized by [5] to predict the stock market prices volatility. They found that sentiment analysis of stock market posts is less accurate than the statistical machine learning methods. They used manually labeled documents and sentiment analysis to label the new documents. After that, they used indexed posts with Support Vector Machine (SVM) classifier to build their model. Their model consists of two main parts: SVM and Generalized Autoregressive Conditional Heteroskedasticity (GARCH). GARCH is used to model the financial time series and SVM is used to estimate GARCH's nonlinear function between the time varying and auto-regression. The system proposed by [1] is a sentiment-based real-time system to monitor the market movements. The authors used variations of classifiers to classify posts taken from an investment forum called HotCopper. The main advantage of HotCopper is the sentiment annotation associated with each post. Naïve Bayes (NB) was employed to classify investors' sentiments, and classification was improved by using the Term Frequency

978-1-5386-5398-2/18/$31.00 ©2018 IEEE

Inverse Document Frequency (TF-IDF) transformation to rank terms according to their TF-IDF values. The authors used Bernoulli model of Naive Bayes to test their classifier. About 7,200 features were selected including positive and negative bigrams and trigrams. The classification F-score was about 77.50%. In the research done by [6], it was found that stock market textual information is more important than the trading volume in predicting the market volatility of some firms (not for short period of time e.g. day). They used NB algorithm to classify 1,559,621 of stock messages into three main classes: BUY, HOLD and SELL. Then, they measure the bullishness of the market in a specific period of time by aggregating those classified messages into indices. All HOLD messages were ignored because the noise in this group dominates the neutral related messages. Among the measures the authors used is the activity which was measured in thousands of messages and the intensity which is the average number of words per message.

A study by [7] has investigated the correlation between the general mood of people and the market behavior. The authors achieved about 76% accuracy using Self Organizing Fuzzy Neural Networks (SOFNN). SOFNN has two preprocessed inputs: the row Dow Jones Industrial Average (DJIA) values and the sentiment analysis of the tweets. The sentiments were categorized into four main classes: calm, happy, alert, and kind. From sentiment analysis perspective, the problem with this study is the domain of the sentiments. The data was general tweets talking about many things including the stock market, and it is not taken from more specific and related board to the market such as investment forums. In other words, instead of general sentiment such as “happy,” more related features can be selected (like SELL, BUY, Up, Down, etc.) to give clear sentiment about the market direction.

Arabic sentiment analysis is covered in a lot of research summarized by [3]. Some research adopted corpus-based models and some of them adopted lexicon-based models, and each group has some reasons for their decision. In [3], they have built a tool for Arabic sentiment analysis and introduced a corpus and a lexicon to the interested researchers. They measured the accuracy on different sizes of lexicons, different types of corpora and with and without stemming. From a decision making point of view, the problem with their study is the diversity and impurity of the data. They translated 300 words from English to Arabic as their first step to build the lexicon which might not be in the same level of quality as taking terms from Arabic resources. In addition to that, they talked about poor performance because of gathering more than one Arabic dialect in the same experiment (this might affect the decision making process seriously).

The closest work to ours was done by [8]. Hamed et al. collected 1,943 tweets and classified them using machine learning algorithms: NB, SVM, and KNN. They tried to find any relation between the sentiments and the market direction, and they got 24% of relation between the downside of the market and the negative sentiments, and they got 36% of relation between the upside of the market and the positive sentiments. The main difference between this work and our work is, we provided a robust lexicon for SSM, we included both investment forum posts (over long period of time) and tweets with much bigger numbers, and we used more methods of analysis and more performance measures.

III. Data Collection: Corpora and Lexicon

Since the decision of investing in the stock market is risky somehow, we decided to build our lexicon manually to assure more robustness, accuracy, and reliability. The first author of this paper, had a portfolio in the Saudi market for at least 5 years and he has enough experience to gather the key words in this matter. We constructed and labeled two corpora and one lexicon. The first corpus was taken from the very well known Saudi investment forum called Saudishares[1], and the second one is taken from Twitter. The lexicon was built manually out of those corpora.

A. Saudishares Corpus

This forum corpus was collected from Saudishares forum, one of the biggest Saudi investment forums that has been going on for more than ten years now and has tens of thousands of comments and reviews about SSM. We collected 18,695 posts using the tool OutWit Hub[2] with our customized macros and scrapers to navigate through all pages and pick only the bodies of the posts. We designed macros and scrapers to crawl through the forum and select the main posts for the last ten years (not the replies and side comments). We aimed to cover a long period of time to make sure we include all diversities of market terms and to cover all major events and crisis that happened in the Saudi market, such as the crash that happened in 2006, to have all levels of positivity and negativity features in our corpus. Most of the titles were included in the posts themselves, so we excluded all titles. After collecting the posts, we removed some missing values (rows that have almost nothing) to have 18,177 posts left. Then, those post were cleaned and annotated manually to positive, negative, and neutral categories. The number of posts of each class is as follows: 4,029 positive, 1,544 negative and 12,604 neutral.

B. Twitter corpus

Twitter corpus consists of 8,940 tweets including missing (no text, just few symbols) values; after removal of missing values the number was 8853 tweets. The most popular domain-related Twitter accounts were selected to be on the focus of our corpus construction process. We have chosen about 30 Twitter accounts according to some indicators such as the number of tweets, number of followers, and the number of likes. These accounts in general give advices to the investors, like when to buy and when to sell, good and bad news, and so forth. We consider only some of those popular accounts who give textual information, because some of them are just referring the followers to other websites that have charts explaining the movement of the market. We extracted tweets from Twitter using Twitter Application Program Interface (API) and the

[1] www.saudishares.net/vb/

[2] https://www.outwit.com/products/hub/

python library tweepy[3]. We employed the python tool developed by Yanofsky and available on GitHub[4]. As we did in the Saudishares corpus, we labeled Twitter corpus (8,853 tweets) manually taking in consideration the importance of the right decision in the stock market. To minimize the risk of false positive labeling (when used later in automated labeling), we label a tweet (and post) as positive if it is clearly saying something positive about the market such as a recommendation to buy, an encouraging announcement, or at least assuring the normal investment environment with no high risk. The same process was used for the negative documents; we make sure it is clearly negatively opinionated. In this corpus, there are 2,224 positive, 612 negative and 6,017 neutral documents.

IV. Text Preprocessing and Features Reduction

In the preprocessing stage, we cleaned and processed both Twitter and Saudishares corpora to be annotated and used to train the models. The preprocessing stage went through the following steps: removing unwanted parts such as urls, removing stop words, removing non-Arabic characters, and tokenization.

We decided not to do stemming for some reasons. First, we work with Arabic informal text (Saudi dialect) which has to be processed in a special customized way as it has so many slangs, special abbreviations, and semi-permanent spelling mistakes(even native speakers fall in them frequently). Some of those spelling mistakes are tricky, such as the word ”ظلال,Dilaal” which means shades; it can be mistakenly written as ”ضلال,Dalaal” which means misguidance. Second, we searched the literature and found that the Arabic stemmers are still suffering from serious weaknesses as mentioned by Larkey2006.

V. Saudi Stock Market Lexicon (SSML)

SSML is a domain-dependent sentiments lexicon constructed specifically for the Saudi stock market investors, or any other parties, to make decisions. SSML contains terms and their sentiment polarities (positive and negative) with two levels of weight, A and B, where level A is stronger than level B in both positive and negative polarities. Our lexicon is constructed manually by analyzing both corpora of Twitter and Saudishares forum. The lexicon is divided into two main parts: positive and negative. Each one of those parts is split into two sub-parts A and B. Part A contains the most unique words that can be used alone to give a very clear indication about the polarity of the document. As an example of these words is a word like ”congratulation,مبروك,” ”green, اخضر,” or the word ”excellent, ممتاز;” the occurrence of such words in a sentence can give enough sign of positivity. It is not very common (as far as we know) to say ”very green” or ”not green.” Part B is less intensive than part A as it might be affected heavily by other features such as negation or intensification (e.g. very, quite…etc.).

Table I: Number of terms in the Saudi stock market lexicon (SSML).

	Group A		Group B	
	un-stemmed	stemmed	un-stemmed	stemmed
Positive	1184	266	702	445
Negative	1026	290	949	653

Negative lexicon part A has clear identifying tokens that work very well to separate the negative documents from others (as we will see shortly). This is because negative terms in the stock market are mostly mentioned in pure negative context, while some positive words can be mentioned in both positive and neutral context. For example, the sentence: ”good morning,صباح الخير,” has no contribution towards the polarity of the tweet, but can be misclassified as positive because of the word ”good,الخير,” which is in the positive lexicon. Consequently, we supported the positive classification by other features such as the stock name, numerical values, and Arabic negation words list.

Our lexicon is reviewed and revised by two experts in stock market to make sure we included most of the key terms. Although our lexicon did a great job in the process of decision making in the Saudi stock market investment, we might do more enhancements in the future such as POS tagging and using some of the words as seeds to expand SSLM to include more Arabic sentiment words. That needs more manual work as there are some positive words in the stock market domain but neutral in other domains such as ”اخضر,green” or ”دخول,entrance.” The word “green” is positive sentiment associated with the up direction of the market (means the stock is increasing) and the word “entrance” gives an impression that the market is safe to invest in. The size of each section of our lexicon is shown in Table I.

As we mentioned earlier, we work with un-stemmed tokens in this study. We preserved all forms of the word (including spelling mistakes that are somehow frequent) specifically the key words as they have been used repeatedly in the stock market discussion forums. Although the difference between the number of stemmed and un-stemmed terms is big especially in group A, at this stage of our research, we care about the comprehension, and accurate decision more than the size of the text being processed. A light stemmer might kill a very useful key word such as ”happiness.” The word happiness in Saudi dialect is ”وناسه,” both of the words ”وناسه,happiness,” and الناس,people,” are mapped to the same stem: ”ناس,people.” This means some neutral words might be taken as key words or key words might be treated as normal words depending on what is in the lexicon.

VI. Experiments

Our experiments went through two main stages as in the following sections. We tried a variety of algorithms, data sizes, and data types to select the best and make it the cornerstone of our decision making process.

[3] https://github.com/tweepy/tweepy

[4] https://gist.github.com/yanofsky/5436496

Table II: Corpus-based classification results using SVM.

Data Set	Balanced?	Accuracy	Recall	Precision	F-Score
Posts 3 classes	U	70%	36%	77%	32%
	B	70%	39%	33%	36%
Posts 2 Classes	U	70%	51%	96%	49%
	B	70%	51%	82%	41%
Tweets 3 classes	U	70%	41%	72%	40%
	B	70%	42%	35%	38%
Tweets 2 classes	U	70%	50%	49%	50%
	B	70%	64%	83%	63%

Table III: Accuracy of statistical learning algorithms for both tweets and forum posts with different sizes.

			Algorithms Accuracy						
Data Sets			LDA	QDA	PDA	RF	C5.0	KNN	NNET
Raw Posts	3 Classes	Balanced	66%	63%	67%	63%	67%	61%	68%
		Unbalanced	74%	72%	74%	77%	76%	63%	76%
	2 Classes	Balanced	81%	79%	81%	80%	80%	77%	81%
		Unbalanced	84%	84%	84%	85%	85%	83%	85%
Cleansed Posts	3 Classes	Balanced	67%	64%	67%	66%	67%	61%	69%
		Unbalanced	75%	73%	75%	76%	76%	74%	77%
	2 Classes	Balanced	81%	79%	81%	82%	80%	79%	81%
		Unbalanced	84%	84%	84%	84%	85%	84%	85%
Raw Tweets	3 Classes	Balanced	65%	65%	65%	67%	65%	57%	67%
		Unbalanced	73%	70%	73%	74%	75%	71%	75%
	2 Classes	Balanced	80%	80%	81%	77%	80%	75%	82%
		Unbalanced	86%	85%	87%	86%	86%	86%	87%
Cleansed Tweets	3 Classes	Balanced	67%	68%	67%	71%	66%	62%	68%
		Unbalanced	73%	72%	73%	76%	76%	73%	75%
	2 Classes	Balanced	78%	79%	78%	81%	79%	76%	78%
		Unbalanced	87%	86%	87%	84%	85%	83%	86%

A. Corpus-Based approach (CBA)

Support vector machine (SVM) was employed to classify documents in both tweets corpus and Saudishares corpus as it is recommended by some resources, such as [9], as the best way to classify text. We provided SVM with manually annotated corpora (about 80% of the data for training) and we got the results shown on Table II. We tried several combinations of the following items of data: posts or tweets, two or three classes, and balanced or unbalanced data. In our study, we care about recall, precision, and f-score more than accuracy. As we can see in Table II, the best F-score we got is 63% and the best recall is 64%. We used the R package RTextTools, developed by [10], that has the function SVM implemented in it. The SVM arguments used are: Kernel= radial, Gamma=1, and Cost=1. More details about how SVM works can be found in [11]. In the following section, we will use the same data but with our designed rules and compare it with CBA. We will focus on how to get the best results for investors to make the right and safe decision.

B. Semantical and Lexical Based Approach (SLBA)

In this section we used two approaches, statistical approach (machine learning) and rule-based approach with both semantical and some lexical features as follows.

1) Statistical approach: we used SSML assisted with some lexical features (e.g. digit characters) to classify documents (tweets/posts). We employed several supervised machine learning algorithms mentioned by [12] and trained them (with 80% of the data) to be able to classify our documents into one of the three categorical responses: positive, negative, and neutral. We used Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), Neural Network (NNET), K-Nearest Neighbors (KNN), Penalized Discriminant Analysis (PDA), Decision Tree C5.0 (C5.0), and Random Forests (RF).

Before we apply those machine learning algorithms, we first generated several data frames based on our lexicon SSML and the lexical features. We extracted several features from the documents and we selected the top eight qualifying predictors (variables) as listed below. Those predictors were selected according to the Best Subset Selection (BSS) method explained by [11], which fit a least squares regression for each possible set of p predictors. The top selected attributes are as follows:

1. Document length: number of tokens per document.
2. Noise similarity: how similar is the document to the noise set, where noise set is a set that contains the useless words.
3. Positivity A: the intersection between the document and the positive lexicon A.
4. Positivity B: the intersection between the document and the positive lexicon B.
5. Negativity A: the intersection between the document and the negative lexicon A.
6. Negativity B: the intersection between the document and the negativity lexicon B.
7. Digits occurrence: categorical value (0 or 1) to indicate whether the document contains numbers or not. This is useful because the nature of tweets and posts in the Saudi stock market might include a very short advice consisting of a number and very few words. The words represent the name of the target stock and the number represents the next price target that the stock expected to reach.
8. Stocks occurrence: a number to quantify how many stocks targeted by the document. This is a good indication of positive sentiment as it is a common way in the Saudi share market forums to give advices to invest in some stocks. For example if the tweet is: “SARCO 31.90,” that means, SARCO is expected to go up to reach the next price level at 31.90 very soon.

We repeated the experiment by applying the machine learning algorithms mentioned in Table III on the defined predictors many times with a different data set in each time. The data sets are formed according to either the data is clean (processed) or not, balanced or not and if it contains two or three classes (two classes means positive and negative). The results are listed in Table III. It is noticeable that the accuracy difference between two and three classes datasets is between 10% and 20%, two classes' data is more accurate. Although we removed all kinds of unwanted characters and symbols, that did not enhance the accuracy that much (less than 2%). The accuracy of unbalanced data (one class is a majority) is higher than the accuracy of balanced data. As we can see from LDA accuracy, when we deal with two classes only, the accuracy is higher than the case of three classes in both posts and tweets. The unbalanced data in this case, always give higher accuracy than the balanced data, which might be misleading sometimes as we mentioned before. For this reason, we count on other measures when it comes to decision making later in this paper.

Table IV: Rule-based classification performance for data set of three balanced classes

		Accuracy	Recall	Precision	F-score
Posts	Positive	70%	81%	66%	73%
	negative	74%	81%	71%	76%
Tweets	Positive	73%	74%	73%	74%
	negative	76%	69%	81%	74%

2) Rule-based approach: rule-based classification means using IF-THEN conditions to classify data. This process goes through three main stages as stated by [13], rule creation stage, rule ranking measure stage, and classification stage, which will be shown in our algorithms shortly. Our rules are divided into two categories; one is to deal with the two classes' data set and the other one is to deal with the three classes' data set.

Three Classes Rules

In the three classes' data set, we included the three labels: positive, negative, and neutral. The problem here is when a document is not classified as positive or negative, it falls in the big (default) class which is neutral. The size of neutral class is much bigger than the other two classes (69% in posts and 68% in tweets). This means big number of correctly classified documents as neutral and a high accuracy as a result. Because of that, the accuracy might not be the best metric for decision making process. So, we focused on other measures like recall, precision, and F-score, [14]. As shown in Table IV, we got good results on all performance metrics for such big data with all classes in consideration.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Fscore = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

The trader might be interested more in market positivity or negativity, depending on his eagerness to buy or sell. Because of the indistinct boundaries between those three classes and because it is more useful towards decision making to focus on one class only, we adopted a method called One-vs-All (OVA) classification, that was explained by [15] to classify documents into two categories: target and otherwise. The target category is either positive or negative depending on what we are trying to measure. If the investor is interested in negativity more than positivity to be able to buy shares with low prices, then in this case the two categories are: negative and otherwise. When the target class was the positive, we sampled 8,058 documents out of the Saudishares corpus having 50% of them positive and 50% otherwise, and we sampled 4,448 documents of the tweets corpus divided into two equal halves positive and otherwise. We used 10-fold cross validation to get the best results. The same process is repeated as well for the negative class as the target class.

As shown in algorithm 1 in the Appendix, our algorithm consists of three main parts: feature extraction part (steps 3-13), scoring part (steps 14-16), and documents labeling part (steps 17-20). The input is a cleansed corpus of documents (tweets or posts), and the output is annotated corpus. Using the concept of bag-of-words, we utilized our lexicon to compute the values such as posA, posB, negA, negB and so forth. The variable posA for example means the occurrences of words from the positive lexicon part A in the document. The other abbreviations are:positive lexicon A (PLA), positive lexicon B (PLB), negative lexicon A (NLA), negative lexicon B (NLB), companies' names list (CL), negation words list (NegationL).

The values C1, C2, th1 and th2 are integer values chosen carefully to maximize the separability between target and otherwise classes. As an example of the best combination of those values is the set of {3, 6, 4, 1} for {C1, C2, th1, th2} respectively. From the contingency matrix (confusion matrix), we noticed that the negative class is more separable than the positive class, as indicated by the precision. When the target is the negative class, we had higher precision which means that the negative lexicon is more robust than the positive one. In such noisy data, we preferred to learn only one target class (recognition-based or one-class approach), as stated by [16], to resolve the problem of imbalanced data and to help to make the right market decision. Moreover, our choice to deal with the data as it (three classes, imbalanced, noisy) is to be as close as possible to the real world data in the Saudi stock market.

Two Classes Rules

In algorithm 2 shown in the Appendix, we handled only positive and negative classes. Rules were built to sharpen the boundaries between positive class and negative class as much as possible. We developed rules for both positive and negative classes but we showed the positive class in algorithm 2 as an example. We did the experiment several times with balanced (same number of positive and negative documents) and unbalanced data. For the Saudishares balanced corpus, we had 3,088 posts, 50% positive and 50% negative, and for the unbalanced Saudishares data, the number of positive posts was 4,029 and the number of negative posts was 1,544 (5573 total). For the tweets, we had 1,224 tweets for the balanced data set, and 2836 for the unbalanced data set (2,224 positive and 612 negative).

Although the rule is heavily driven by the lexicon SSML, the other features also give a good support and enhancement towards the decision quality; for example, the feature repL shown in algorithm 2, represents the occurrence of repeated letters (such as the word gooood) in the document. The percentage of documents that contain repeated letters in the tweets corpus (positive and negative class only) for example is about 13% of the documents number. Another feature is the occurrence of digits in the document. It is mostly a clear sign of an advice being given to the investors.

The balanced data gives less accuracy than the unbalanced, which is expected as the imbalance ratio (defined by [16] as minority class/majority class) is high. The positive class is majority in both posts and tweets; and as a result, the imbalance ratios are about 38% and 27% in posts and tweets respectively. The recall scores are very high for both tweets and posts (between 96% and 98%), and the minimum accuracy was 68% of balanced tweets data set (for details see Table V in the Appendix).

VII. Rule-based fuzzy decision

With the method rule-based approach, we got the best results with respect to the recall metric. According to that, we adopted rule-based approach as our way to make decisions. Since we are dealing with some uncertainty of the market risk level, the theory of fuzzy sets that was introduced by [17], can be employed as the decision making mechanism in this paper. The fuzzy logic is able to handle the partial truth, which means that the truth is not only either zero or one, it can be true up to a limit. In other words, the membership of an element x to a set U can be partial instead of making strict decision either it is a member or not. For example, instead of saying today is cloudy, we can make fuzzy decision by saying today is 70% cloudy and 30% sunny.

Now let S be the set of safe decisions. We call this set fuzzy, because the degree of membership to this set is not binary (yes or no), it has some degree of membership as explained in the paper by [18]. So the items of this set (decisions) can have membership degree in the interval [0, 1], where zero is not very safe at all, 1 is very safe, and some decisions fall in between. The membership function of the set S is denoted by μ_S. It can be written as $\mu_S : X \longrightarrow [0, 1]$, Where X is the universal of the fuzzy set. As in the decision diagram shown in figure 1, the corpus has some degree of positivity and some degree of negativity between 0 and 1, depending on the percentage of positive and negative documents in the corpus. So to make a decision about corpus D polarity, we use equation 5, where T refers to the target class, and O refers to the otherwise class.

$$\mu_S(D) = max \quad [\mu_T(D) - \mu_O(D), 0] \tag{5}$$

As the difference between the two sets (target and otherwise) gets higher, the level of decision safety gets higher as well. Again, the target of the investor might be the positive class or the negative class, depending on their investment plan. Some investors look at the market downside as an opportunity to buy.

VIII. Conclusion

In this paper we manually built our own lexicon and labeled corpora specifically for the domain of Saudi stock market. The lexicon and the two corpora were analyzed using two methods: corpus-based approach and semantical and lexical based approach. We have selected the rule-based approach as the best way to classify documents. The classified documents were used as an input to a fuzzy decision mechanism for the Saudi stock market. We counted more on recall than any other metric as it is more realistic towards our goal. In the future work, we will apply our strategy over a specific period of time and compare our results with the real movement of the Saudi market. Some enhancements might include more noise filtering, more features extraction, and lexicon expansion.

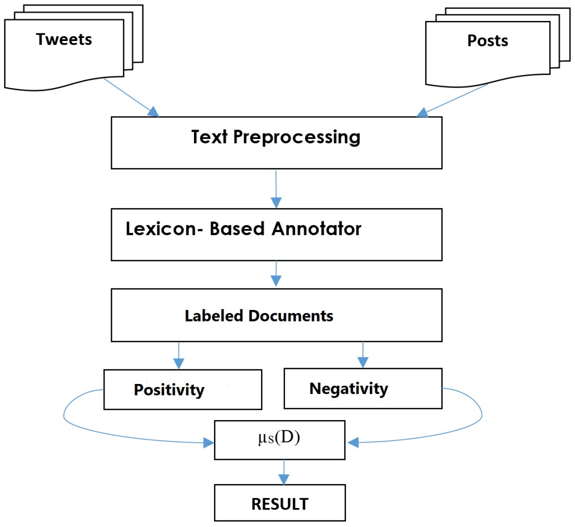

Figure 1: The fuzzy decision main components

References

[1] J. Bollen, H. Mao, and X. Zeng, "Twitter mood predicts the stock market," *Journal of computational science*, vol. 2, no. 1, pp. 1–8, 2011.

[2] M. I. M. Alnajjar, "Behavioral Inferences of Tadawul Investor," *International Journal of Business and Management*, vol. 8, no. 24, p. 17, 2013.

[3] N. A. Abdulla, N. A. Ahmed, M. A. Shehab, M. Al-Ayyoub, M. N. Al-Kabi, and S. Al-rifai, "Towards improving the lexicon-based approach for arabic sentiment analysis," *International Journal of Information Technology and Web Engineering (IJITWE)*, vol. 9, no. 3, pp. 55–71, 2014.

[4] S. W. K. Chan and J. Franklin, "A text-based decision support system for financial sequence prediction," *Decision Support Systems*, vol. 52, no. 1, pp. 189–198, 2011.

[5] D. D. Wu, L. Zheng, and D. L. Olson, "A decision support approach for online stock forum sentiment analysis," *IEEE Transactions on systems, man, and cybernetics: systems*, vol. 44, no. 8, pp. 1077–1087, 2014.

[6] W. Antweiler and M. Z. Frank, "Is All That Talk Just Noise? The Information Content of Internet Stock Message Boards ," pp. 1259–1294, 2004.

[7] A. Mittal and A. Goel, "Stock prediction using twitter sentiment analysis," *Standford University, CS229 (2011 http://cs229. stanford. edu/proj2011/GoelMittal-StockMarketPredictionUsingTwitterSentimentAnalysis. pdf)*, vol. 15, 2012.

[8] A.-R. Hamed, R. Qiu, and D. Li, "Analysis of the relationship between Saudi twitter posts and the Saudi stock market," in *Intelligent Computing and Information Systems (ICICIS), 2015 IEEE Seventh International Conference on*. IEEE, 2015, pp. 660–665.

[9] A. Abbasi, H. Chen, and A. Salem, "Sentiment analysis in multiple languages: Feature selection for opinion classification in web forums," *ACM Transactions on Information Systems (TOIS)*, vol. 26, no. 3, p. 12, 2008.

[10] L. Collingwood, T. Jurka, A. E. Boydstun, E. Grossman, and W. H. van Atteveldt, "RTextTools: A supervised learning package for text classification," 2013.

[11] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An introduction to statistical learning*. Springer, 2013, vol. 112.

[12] M. Kuhn, "Caret package," *Journal of Statistical Software*, vol. 28, no. 5, pp. 1–26, 2008.

[13] A. K. H. Tung, *Rule-based Classification*. Boston, MA: Springer US, 2009, pp. 2459–2462. [Online]. Available: https://doi.org/10.1007/978-0-387-39940-9_559

[14] T. Fawcett, “An introduction to ROC analysis,” *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.

[15] R. Rifkin and A. Klautau, “In defense of one-vs-all classification,” *Journal of machine learning research*, vol. 5, no. Jan, pp. 101–141, 2004.

[16] S. L. Phung, A. Bouzerdoum, and G. H. Nguyen, “Learning pattern classification tasks with imbalanced data sets,” 2009.

[17] L. A. Zadeh, “Fuzzy sets,” *Information and control*, vol. 8, no. 3, pp. 338–353, 1965.

[18] M. J. Wierman, “An Introduction to the Mathematics of Uncertainty,” *Creighton University*, 2010.

Appendix

Tables and Pseudo-Codes

Algorithm 1 Rules for classification of three classes using the method One-Vs-All.

```
1:  **Input**:Cleansed Corpus of documents,PLA, PLB, NLA, NLB, CL,NegationL
2:  **Output**: Annotated corpus.
3:  **for** each document $d_i$ **do**
4:      $d_i \leftarrow$ tokenize $(d_i)$
5:      $M \leftarrow$ number of numerical values in $d_i$
6:      $posA \leftarrow d_i \cap$ PLA
7:      $posB \leftarrow d_i \cap$ PLB
8:      $negA \leftarrow d_i \cap$ NLA
9:      $negB \leftarrow d_i \cap$ NLB
10:     $comSim \leftarrow d_i \cap$ CL
11:     $qSim \leftarrow d_i \cap$ NegationL
12:     $repL \leftarrow$ either 0 or 1 to represent the absence or presence of repeated letters in $di$
13:     **end for**
14: **end for**
15: $pScore \leftarrow C1*posA + posB$ // positive score
16: $nScore \leftarrow C1*negA + negB$ // negative score
17: $SCORE \leftarrow Pscor - C2 * Nscor + C2 * comSim + C2/10 * M - C2 * qSim$
18: **if** $((SCORE > th1 \vee Pscor > th2) \vee (comSim > 0 \wedge M > 0))$ **then**
        Label = POSITVE
19: **else**
20:     Label = OTHERWISE
21: **end if**
```

Algorithm 2 Classification rules for data set of two classes: positive and negative.

```
1: In Addition to steps 4-12 in algorithm 1, we do
2: $pScore \leftarrow 2 * posA + posB + M + repL$ // positive score
3: $nScore \leftarrow negA + negB$ // negative score
4: **if** $(pScore \geq nScore) \vee |di| <$ Threshold1 $\wedge di \cap$ stock list $>$ Threshold2 $\wedge\ di \cap$ NLA $== 0)$ **then**
       LABEL = POSITVE
5: **else**
6:     LABEL = NEGATIVE
7: **end if**
```

Table V: Rule-based classification performance for data set of two classes: positive and negative.

		Accuracy	Recall	Precision	F-score
Posts	Balanced	70%	96.50%	63%	76%
	Unbalanced	82%	96.50%	81.80%	88%
Tweets	Balanced	68%	96.70%	61%	76%
	Unbalanced	85%	97.70%	85%	91%