A hybrid multilingual fuzzy-based approach to the sentiment analysis problem using sentiwordnet

Youness Madani, Mohammed Erritali, Jamaa Bengourram, Francoise Sailhan

A Hybrid Multilingual Fuzzy-based approach to the Sentiment Analysis problem using SentiWordNet

1st MADANI Youness
GI laboratory
Departement of Computer Sciences
Faculty of Sciences and Techniques
Sultan Moulay Slimane University
Beni Mellal, Morocco
younesmadani9@gmail.com

2nd ERRITALI Mohammed
TIAD laboratory
Departement of Computer Sciences
Faculty of Sciences and Techniques
Sultan Moulay Slimane University
Beni Mellal, Morocco
m.erritali@usms.ma

3rd BENGOURRAM Jamaa
GI laboratory
Departement of Industrial Engineering
Faculty of Sciences and Techniques
Sultan Moulay Slimane University
Beni Mellal, Morocco
bengoram@yahoo.fr

4rd Francoise SAILHAN
National Conservatory of Arts and Crafts(CNAM)
PARIS , France
bengoram@yahoo.fr

***Abstract*—Sentiment Analysis is a new research area that is increased explosively, this domain has become a very active research issue in data mining and natural language processing. Sentiment analysis(opinion mining) consists of analyzing and extracting emotions, opinions or attitudes from product's reviews, movie's reviews..etc, and classify them into classes such as positive, negative and neutral, or extract the degree of importance(polarity). In this paper, we propose a new hybrid approach for classifying tweets into classes based on the fuzzy logic and a lexicon based approach with the use of SentiWordnet. Our approach consists of classifying tweets according to three classes: positive, negative and neutral using SentiWordNet and the fuzzy logic with its three important steps: Fuzzification, Rule Inference/aggregation and Defuzzification. The dataset of tweets to classify and the result of the classification are stored in the Hadoop Distributed File System(HDFS) and we use the Hadoop MapReduce for the application of our proposal.**

***Index Terms*—Twitter,opinion mining, Sentiment analysis,Fuzzy Logic,SentiWordNet, big data, Hadoop**

I. Introduction

Social Networks like Twitter, Facebook, and Google+ are products of the Internet, where people can express their opinions, attitudes, and feelings about products, movies, social events...etc. The big advantage of social networks is that anyone can express what he thinks in a freeway without hindrance, so that demonstrates the big amount of data shared every day that can be used after in several domains like sentiment analysis or opinion mining [1] [2].

Among variety of social networks, Twitter for example, is a popular microblogging website with over 328 millions active users per month and about 500 million tweets per day in over 40 languages. Messages are limited to 280 characters and are known under the name tweets and may include text, URLs, other user mentions and hashtag metadata to messages. These tweets represent the users' opinions and thoughts expressed in short and simple messages. Twitter gives everyone the power to create and share ideas and information instantly and without hindrance[1].

With this freedom to share on Twitter(which launched in 2006) and the expression of personal feelings about a product or brand, many researchers use Twitter in the field of sentiment analysis or opinion mining. This type of classification helps people to make a decision about buying a product/brand or not, by classifying all the tweets related to this brand to know the opinions (positive-negative) expressed in them. Or for example for defining the motivation of someone (motivated, demotivated) from tweets published in his twitter profile.

Sentiment Analysis(SA) sometimes called opinion mining is the process by which text is analyzed to extract opinions, sentiments...etc, and to classify a document or a sentence into classes(positive,negative,neutral) or sometimes it tries to give to a document or a sentence(review) a degree of importance(the polarity). The process of sentiment analysis uses a lot of other domains [3] like Natural Language Processing(NLP) through a combination of pre-processing steps, machine learning algorithms(supervised and unsupervised) and relevant statistical techniques(Lexicon-based approach, corpus-based approach).

Sentiment analysis is used in many domains of our daily life like in economy and Business. It helps to find answers to questions like, ' Why a product selling is low', 'Have users need are satisfied using our services' or' are our users happy or want more'. So with sentiment analysis company don't need to make manual research for having an idea about his products.

[1]https://about.twitter.com/fr/company

Sentiment analysis can also be used to predict the results of an election or for analyzing the motivation of learners in an e-learning platform.

In recent years, after Twitter was launched many researchers use it in sentiment analysis because the number of tweets published every day and also the diversity of domains of the publications. Sentiment analysis over Twitter tries to classify the tweets into classes(positive, negative or neutral). To classify the tweets, researchers use many methods, namely the use of data mining algorithms (supervised and unsupervised), the use of dictionaries like AFINN[2], Sentiwordnet[3] and SenticNet[4]. We find in the literature also some works that use the semantics in sentiment analysis.

Social data and especially Twitter data grow rapidly in size, variety, and complexity with this huge number of user that published tweets every day, so it is the problem of how to store the tweets (necessity of a large volume of storage) and also the problem of the calculation time needed to have the result of the classification. To remedy these problems we decided to work in a big data system using the Hadoop framework and its technologies(HDFS, MapReduce, Apache Flume, Apache Sqoop). In the literature, many works demonstrate how the use of big data technologies affects positively the analysis of social networks data. In [4] authors propose a new Facebook API called SocialMedia API, this API consists in analyzing the contacts who click like or comment on the authors posts. The proposed API is combined between the BOINC project and Facebook for processing Big Data. Chang et al. [5] proposes a new Facebook API that can extract information quickly and process the selected information efficiently (perform Big Data Analytics), they present how to extract information from Facebook from their developed SocialNetwork API, which processes Big Data of the author's network.

In this paper, we propose a new approach to classify tweets into three classes(positive,negative and neutral), using a hybrid approach based on fuzzy logic and the lexical resource SentiWordNet, in a parallel manner by distributing the process of the classification using the Hadoop framework with the Hadoop Distributed File System(HDFS) and the MapReduce programming model.

Our work will be divided into a number of parts, the first concerning the extraction of tweets from Twitter using a Twitter API or Apache Flume, and store them directly in HDFS using Hadoop Sqoop[5]. After the storage step the second one concerning the application of Natural Language Processing(NLP) Methods. And finally, the classification of the tweets stored in HDFS into three classes: positive, negative and neutral, by applying our proposed approach in a parallel way using MapReduce.

The rest of this paper is organized as follows: Section 2 presents the motivation of our work and defines some literature reviews. In section 3, we will present the concepts of the fuzzy logic, and in section 4 we will describe our research methodology (the different text preprocessing methods exist in the literature, how we collect the tweets, how we classify tweets using our proposed method, and how we parallelize our work). Section 5 presents the experimental results. And finally, in section 6 it is the conclusion and the perspectives.

II. Motivation and Literature Review

The classification of tweets or in general SA knows in recent years a great development in scientific research. Scientific researchers try to find new approaches and methods to improve the quality of the tweets' classification. In the literature, we find a lot of sentiment classification techniques that can be divided into machine learning approaches, lexicon based approaches, and hybrid approaches [6]. The Machine Learning approach (ML) applies the famous ML algorithms like Support Vector Machine(SVM), Naive Bayes, Decision Tree...etc, the approaches based on ML try to classify a sentence, document or a phrase into classes(positive, negative, neutral) [7] [8]. Lexicon based approach relies on a sentiment lexicon, a collection of known and precompiled sentiment terms. It is divided into the dictionary-based approach and corpus-based approach which use statistical or semantic methods to find sentiment polarities like the AFINN dictionary, SentiWordnet or SenticNet. On the other hand, the hybrid Approach combines both approaches, a number of researchers have explored the application of hybrid approaches by combining various techniques with the aim of achieving better results than a standard approach based on only one tool [9].

The classifiers based on ML techniques treat the classification of sentiment in a "black-and-white" manner, while in reality sentiment is rarely clear-cut, most of the previous works focus on methods of deterministic algorithms without considering fuzziness of the sentiments. Reality is always far from optimistic. Firstly, sentiment terms are fuzzy. The same word can explain different sentiment orientations even in the same domain. Secondly, sentiments of human are many times fuzzy. For instance, one may use one word to express more than one feeling at the same time.

In recent years, fuzzy approaches have started to emerge for text processing and sentiment analysis, although that the number of papers lies to fuzzy logic still until today a little compared to the papers that use ML techniques or statistical approaches. In the literature, we find some works that use fuzzy logic in sentiment analysis.

*Authors in [10] proposed a fuzzy computing model to identify the polarity of Chinese sentiment words. This paper is mainly embodied in three aspects. The first consists in computing the sentiment intensity of sentiment morphemes and sentiment words using three existing Chinese sentiment

[2]AFINN is a dictionary that contains words with weights between -5 and 5 which expresses the sentimental degree of the word

[3]http://sentiwordnet.isti.cnr.it/ that is a lexical resource for opinion mining, it assigns to each synset of WordNet three sentiment scores: positivity, negativity, objectivity.

[4]http://sentic.net/, talking about SenticNet is talking about concept-level sentiment analysis, that is, performing tasks such as polarity detection and emotion recognition by leveraging on semantics and linguistics instead of solely relying on word co-occurrence frequencies.

[5]Apache Sqoop(TM) is a tool designed for efficiently transferring bulk data between Apache Hadoop and structured datastores such as relational databases.

lexicons. Secondly, authors of this article constructed a fuzzy sentiment classifier and a corresponding classification function of the fuzzy classifier by virtue of fuzzy sets theory and the principle of maximum membership degree, and Thirdly they constructed four sentiment words datasets to demonstrate the performance of their model.

Another work that uses fuzzy sets in sentiment analysis is presented in [11]. In this article, the authors introduced a new fuzzy logic based approach for the text classification especially the classification of Twitter's message. Inputs used in the proposed fuzzy logic-based model are multiple useful features extracted from each Twitter's message. The output is its degree of relevance for each message to an event called "Sandy". For that, they used a number of fuzzy rules and the different defuzzification methods existing in the literature, and for the fuzzification, they selected the trapezoidal-shaped membership function because it is simple and commonly used. The proposed fuzzy system in this work has as inputs 7 linguistic variables and as outputs 1 linguistic variable which is the relevance of the tweet with sandy. As experimental results, they compared five commonly used defuzzification methods, and they conclude that the centroid method is more effective and efficient than the other methods. Additionally, they conducted a comparison with the well-known keyword search method and the results reveal that the proposed fuzzy logic-based approach is more suitable to classify the relevant and irrelevant Twitter's messages.

Dragoni and Petrucci [12] published an article where they discussed how linguistic overlaps between domains can be exploited for computing document polarity in a multi-domain environment using a proposed fuzzy logic model. The proposed approach integrates fuzzy logic for the representation of the polarity associated with linguistic features belonging to a particular domain. As experimental results, authors compare their method with respect to a selection of six baselines implementing well-known techniques, and they conclude that the proposed strategy outperforms all the baselines.

Authors of [9] present a hybrid approach to the sentiment analysis problem at the sentence level. the proposed method takes advantages of the natural language processing (NLP), the sentiWordNet dictionary, and fuzzy sets for estimating the semantic orientation polarity and its intensity for sentences. For demonstrating the use of a hybrid approach authors compared their work with two supervised machine learning algorithms: Naive Bayes (NB) and Maximum Entropy (ME) using four indices that are: Accuracy, Precision, Recall, and F1-score. The Experimental Results show that the proposed hybrid method using sentiment lexicons, NLP essential techniques, and fuzzy sets, significantly improved the results obtained using Naive Bayes (NB) and Maximum Entropy (ME), with a high level of accuracy (88.02%) and precision (84.24%).

Researchers of the article presented in [13] used Neural Network and Fuzzy sets to improve the quality of sentiment classification. This classification method uses advantages of both fuzzy logic and Neural Network NN to build a classifier. To fuzzify the input reviews Authors used the Gaussian Membership function, the MAX principle for the defuzzification, and for the classification they used A Multi-layer perceptron back propagation network (MLPBPN).

In [14] authors proposed a fuzzy rule-based system for obtaining degrees of sentiment, they used a trapezoidal MF for the fuzzification and the max of the output for the step of defuzzification. As experimental results of this work, authors compared the accuracy of the proposed approach with the accuracy of two other machine learning algorithms (Naive Bayes and Decision Trees), and the results show that the fuzzy proposed method achieved the same level of performance as the other two algorithms.

Liu et al. [15] published a paper that uses a fuzzy rule-based system as computational models towards accurate and interpretable analysis of sentiments. Authors used four datasets of film reviews using fuzzy rule-based systems. For that, they used the Tsukamoto fuzzy rule-based system due to the fact that this type of fuzzy systems applies to the classification problems. For the fuzzification step, they used the trapezoid fuzzy membership function, the min/max methods for the application of rules and for the aggregation, and the maximum of outputs for the defuzzification step. As experimental results, they compared the fuzzy rules approach with the computational models learned by using other popular machine learning algorithms (Naive Bayes and C4.5) in sentiment classification. In terms of classification accuracy using four polarity data sets on movie reviews, the results show that the fuzzy rule learning approach performs slightly better than the well known Naive Bayes and C4.5 algorithms, thus indicating the suitability of fuzzy rules approaches for sentiment analysis tasks.

In the paper of Wang et al. [16] a new sentiment computation approach which is defined as public sentiments discriminator (PSD) is proposed based on fuzzy logic and sentiment complexity. Experiments show that the proposed approach(PSD) can achieve similar accuracy and F1-measure but more cognitive results when compared with traditional well-known machine learning methods.

Authors of [17] use the data coming from Twitter to analyze the performance of Indian movies, the authors used Twitter4j Java API for extracting the tweets by authenticating connection with Twitter and stored the extracted data in a MySQL database and used it after for sentiment analysis, for that they used the fuzzy inference system to classify the movie reviews into three classes : hit, flop and average.

the work of Karyotis et al. [18] implemented an adaptive fuzzy modeling method using optimized parameters with the help of a GA for incorporating human emotion into intelligent computer systems, they used a hybrid cloud intelligence infrastructure to conduct large-scale experiments and analyze user sentiments and associated emotions, using data from a million Facebook users.

Morente-Molinera et al. [19] present a novel method that uses sentiment analysis procedures in order to automatically create fuzzy ontologies from free texts provided by users in social networks. In this paper, a novel method that is capable of extracting collective knowledge from users opinions and

represent it in a fuzzy ontology is developed. The novel developed method uses sentiment analysis procedures in order to extract the subjective information that is present in users opinions texts.

III. Fuzzy Logic

In this section, we will present the main concepts of the fuzzy logic: fuzzy sets, fuzzification, fuzzy Rules inference, and Defuzzification.

The fuzzy logic idea is similar to the human being's feeling and inference process, it is an approach of computing based on "degrees of truth" rather than the usual "true or false" (1 or 0) Boolean logic on which the modern computer is based. The theory of fuzzy logic is mainly aimed at turning a black and white problem into a grey problem [20], in the context of set theory, deterministic logic is corresponding to crisp sets [15], the idea of fuzzy logic was invented by Professor L. A. Zadeh of the University of California at Berkeley in 1965 [22].

Fuzzy ideas and fuzzy logic are utilized in our routine life that nobody even pays attention to them. For instance, to answer some questions in certain surveys, most time one could answer with 'Not Very Satisfied' or 'Quite Satisfied', which are also fuzzy or ambiguous answers. Exactly to what degree is one satisfied or dissatisfied with some service or product for those surveys? These vague answers can only be created and implemented by human beings, but not by machines. Is it possible for a computer to answer those survey questions directly as human beings did? it is absolutely impossible. Computers can only understand either '0' or '1', and 'HIGH' or 'LOW'. Those data are called crisp or classic data and can be processed by all machines [21].

A. Fuzzy Sets

The concept of the fuzzy set is only an extension of the concept of a classical or crisp set. The classical set only considers a limited number of degrees of membership such as '0' or '1', that is the membership function value of an object in a crisp set is either 0 or 1. The value of 0 signifies that the object does not belong to the crisp set while the value of 1 indicates that the object completely belongs to the crisp set.

A fuzzy set is a generalization of the classical or crisp set with the range of [0,1]. An object may only partially belong to a fuzzy set. The more the object belongs to the fuzzy set, the higher the degree of membership.

The fuzzy set is a powerful tool and allows us to represent objects or members in a vague or ambiguous way, for example in our case we want to represent the sentiment of a tweet in an ambiguous way.

B. Fuzzification

To implement fuzzy logic to a real problem like in our case the classification of tweets, three consecutive steps are needed, which are: Fuzzification, fuzzy inference, and defuzzification.

Fuzzification is the first step to apply a fuzzy logic system that is the process of making a crisp quantity fuzzy, most variables existing in the real world are crisp or classical variables. One needs to convert those crisp variables (both input and output) to fuzzy variables, fuzzification involves two processes: derive the membership functions for input and output variables and represent them with linguistic variables.

The membership functions transform each crisp value to fuzzy sets. In practice, membership functions can have multiple different types, such as the triangular waveform, trapezoidal waveform, Gaussian waveform, bell-shaped waveform, sigmoidal waveform, and S-curve waveform.

So first we have a crisp variable to fuzzify, the next step is fuzzified it using a membership function (triangular, trapezoidal...) then for this variable, we have a degree of membership in the range of [0,1].

C. fuzzy Rules

The second step in a fuzzy logic system (FLS), is the application of the rules on the input and output variables, a rule base is constructed to control the output variable. A fuzzy rule is a simple IF-THEN rules with a condition and a conclusion for example **if** positivity is high **and** negativity **is** low **then** sentiment is positive. The inputs and the outputs of the rules have fuzzy values.

D. Defuzzification

The third step in an FLS is the defuzzification. Despite the fact that the bulk of the information we assimilate every day is fuzzy, most of the actions or decisions implemented by human or machines are crisp or binary. The decisions we make are binary, the hardware we use is binary. So we need to defuzzify the outputs to have an only crisp value that gives us the final result of our problem(the problem of classification for example).

This process of "defuzzification" has the result of reducing a fuzzy set to a crisp single-valued quantity, or to a crisp set, the output of a fuzzy process can be the logical union of two or more fuzzy membership functions defined on the universe of discourse of the output variable.

There is a lot of defuzzification methods in the literature such as Max-membership principle, Centroid Method(also called center of area or center of gravity), Weighted Average Method and Mean-max membership (also called middle-of-maxima).

IV. Research methodology

In this section, we going to present the different steps of our work. As we have presented earlier the aim of our work is to classify tweets (sentiment analysis) related to a domain, product, movie reviews...etc. We classify each tweet according to three classes: positive, negative or neutral using the fuzzy logic system and the lexicon-based dictionary SentiWordNet.

Our proposed approach is hybrid, that is it combined between a dictionary-based approach using SentiWordNet and the fuzzy logic system with its three important steps(Fuzzification, Rule Inference, Defuzzification). The first

stage of our method consists of using the SentWordNet dictionary to calculate two measures: **positivity** and **negativity**, subsection 4.4 gives more details on how to calculate these two measures. The second stage consists of using the positivity and the negativity as inputs of our fuzzy logic system to find in the output the class of the tweet, subsection 4.5 explain all that in details.

Our contributions appear in the following points :

- The first contribution of our work consist in proposing two measures which we have called **Positivity** and **Negativity** of the tweets. The calculation of these two measures is based on the SentiWordNet dictionary and also the POS tags using the ApacheNLP. At the end of this step, each tweet to classify will have the degree of positivity and that of Negativity.
- The Second contribution consists in how to use the measures calculated earlier in a fuzzy logic system. The positivity and the negativity of the tweets will play the role of inputs for the fuzzy logic system, and by applying the three stages of the fuzzy logic(fuzzification, rules inference, and defuzzification) on these two measures, we find in the output the class of the tweet (Positive, Negative or neutral). Using the values of the Positivity and the Negativity we can decide with the help of the fuzzy logic system if the tweet is negative, positive or neutral.
- The third contribution of our work is the parallelization of our proposed approach using the Hadoop framework with the Hadoop Distributed File System(HDFS) and the Hadoop MapReduce. For that, we propose a new MapReduce algorithm based on our proposed approach. That is to say, our approach deals with the problem of scalability.
- Our approach is multilingual. It takes into account all the languages.

To make our proposed approach, we have to implement different steps either in the collection of tweets, the preparation of the tweets for the classification (text preprocessing methods) or in the classification step using our hybrid approach.

A. Tweets dataset

Due to the privacy policy of Facebook profiles, our work focuses on Twitter, where most of the contents and activities shared online are open and available. In addition, there is a lot of API and framework that can help us to collect a large amount of data from Twitter.

The first step of our work is the collection of the tweets to classify which is a very important step. In our work, we used two methods for retrieving tweets from Twitter, the first is a Twitter API called Twitter4j, and the second is Apache Flume.

- Twitter4j[6] gives us the possibility to retrieve tweets according/linked to a product, a hashtag or a movie review in a specified time (for example between June 2015 and July 2017). The retrieved tweets are stored in a relational database such as Mysql and after we transform them to HDFS directly using Apache Sqoop.
- Apache Flume[7] gives us the possibility to extract and store the tweets directly in HDFS.

For using the Twitter4j API or the Apache Flume, we need to create a twitter application [23] in the twitter development space[8]. This application gives as 4 parameters which are very important for retrieving tweets(Consumer Key, Consumer Secret, Access Token and Access Token Secret).

After we create the twitter application, we get parameters such as Access Token, Access Token Secret, Consumer Key (API Key) and Consumer Secret (API Secret), these parameters will be used after by the Twitter4j API or by Apache Flume to collect and to build a dataset of tweets for the analysis(classification step).

As we want to work with Hadoop MapReduce, we store the tweets in text files, and each tweet will occupy a line for facilitating its processing for the classification(because MapReduce works on the input line per line).

B. How we made a multilingual approach?

Most existing sentiment analysis approaches are devoted to English-language data, while a great share of information is available in other languages. With the growth of the Internet around the world, users write comments in different languages. Sentiment analysis in an only single language increases the risks of missing essential information in texts written in other languages.

As mentioned in the title of this article, our sentiment analysis approach is multilingual. That is, it takes into account all the languages over the world. in Twitter, we find for example tweets written in French, Spanish, English...etc.

After we collect tweets-using the Twitter4j API or Apache Flume- and store them in HDFS, Our idea is to translate each tweet written in another language, from this language to English. For the translation we use a java project API called WhatsMate API[9], this API allows us to translate words or phrases from a language to another one.

In our case, we use WhatsMate API to translate tweets written in another language for facilitating the process of the classification and working only with English to make our approach multilingual.

To avoid the problems of translation and the errors that can be caused especially, when translating the whole tweet. We propose to translate it word by word. For that before the translation-if the tweet is not written in English- we split

[6]Twitter4J (twitter4j.org/) is an unofficial Java library for the Twitter API. With Twitter4J, you can easily integrate your Java application with the Twitter service. Twitter4J is an unofficial library.

[7]https://flume.apache.org/, Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and faults tolerant with tunable reliability mechanisms and many failovers and recovery mechanisms. It uses a simple extensible data model that allows for an online analytic application.

[8]https://apps.twitter.com/

[9]http://api.whatsmate.net/

the tweet into words by removing white spaces, commas, and other symbols, and after all that, we translate it word by word.

So the first step to do after the collection of tweets -related to a subject(for example to an event, a product...etc)- is to detect its language. If the detected language is different from English, we translate it in order to make all our dataset written in English.

C. Tweets pre-traitment

After the steps of the collection of tweets and translate those written in another language than English, we have constructed our dataset, but before the classification, we must apply some text pre-processing methods to prepare the tweets for the classification and to delete the noise exists in them. Several works from the literature demonstrate that the application of the text preprocessing methods on the tweets improves the quality of the classification [24] [25]. In our work, we apply some text preprocessing methods such as :

- Tokenization: Which is the phase of splitting the tweet into terms or tokens by removing white spaces, commas and other symbols etc. This step is very important in our work because we focus on individual words, either in the phase of translation or in the phase of classification.
- Removing numbers: that not express any emotions or attitudes. In general, numbers are no use when measuring sentiment and are removed from tweets to refine the tweet content.
- Removing Stopword: There is a kind of word called stopword. They are words of common function in a sentence, such as 'a', 'the', ',', 'to', 'at', etc. These words seem useless for the analysis of the Feeling, therefore they should be deleted.
- Removing Punctuations: We don't need pits as characteristics, these are only symbols for separate sentences and words so we delete them from tweets.
- Stemming: Stemming is another very important process. In our work and because we focus on English language we use the Porter stemming [8].
- Effect of negation: we use a list of words which express the negation such as: not, do not, will not, never, cannot, does not ..., after classification if the tweet is positive or negative Then we use this type of pre-processing text, the idea is that if the tweet, for example, is positive but contains a negation then it will be negative and vice versa. In general, the negation acts directly on the orientation of the polarity of the tweet, therefore, the treatment of the negation must be taken into account in the classification of tweets. For example, Tweet 1 is a positive tweet, in which the polarity is scope by the expression "love":
  (1) I love reading. (Positive Tweet).
  The tweet 2 use the form of the negation" do not," is the negation of the Tweet 1:
  (2) I do not love reading. (negative tweet).
  The proposed solution to this problem is to reverse the polarity.
- Extraction of opinion words: The important step in the text pre-processing methods is the Part-Of-Speech(POS) tags, it gives us the type of each word in the tweets(Verb, Noun, Adjective, Adverb...). In this step, we use the hypothesis which says that only verb, adverb, and adjective can express opinions in a tweet. From this hypothesis, we decide to delete each word of the tweet that is not a verb, adjective or adverb. In this step, we use the Apache OpenNLP library[10]. This text pre-processing technique helps us in calculating the positivity and the negativity measures.

The tweets respond to a set of rules specific to the Twitter service, therefore before to be able to analyze, we will clean up the message of the tweet of certain these specificities (#, rt, @,).

- URL and @: The first step is to delete the URL and the word begins with the '@' symbol. We will not follow the content of the Web links, so the URL will be deleted. The '@' symbol has always a username monitoring, which is unnecessary so that the entire word begins with '@' could be deleted.
- Hashtag #: The word begins with '#' is a hashtag. A hashtag is different from other words, it gives a label or a subject on the tweet. Usually, the tag speaks of the subject to which people say in this tweet, and not on the attitudes of the people. This word may provide information but not important. We have therefore decided not to delete the entire word, but simply delete the symbol '#', and treat the tag as a normal word in a tweet.

D. How we use SentiWordNet?

In this subsection, we will describe how we use SentiWordNet in our work to make a hybrid approach with the fuzzy logic concepts, and how we use it to calculate the two proposed measures positivity and negativity. SentiWordNet gives us the possibility to know for each word its degree of positivity if the word is positive(express a positive sentiment), and its degree of negativity if the word is negative. This degree is between 0 and 1 if the word is positive and between 0 and -1 if it is negative.

From all that, we propose two measures that we have called **Positivity** and **Negativity** of the tweets. Before the calculation of these two measures, it is necessary to detect the language of the tweet and translate it; if it is not written in English, and also apply different text preprocessing methods presented earlier to extract only the opinion words, which will be after used by SentiWordNet to calculate our two proposed measures. For the calculation of the positivity and the negativity, we need to know for each opinion word, if it is positive or negative, for that we calculate its degree from SentiWordNet and if the calculated value is greater than 0 the word is positive, otherwise(if it is less than 0) the word is negative.

[10]The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text.

For calculating the positivity measure of a tweet, we calculate the average of the sum of each positive opinion word's degree, and because the degree of each positive word in sentiwordNet is between 0 and 1, the positivity of the tweet will also be between 0 and 1.

in the same way, for the negativity measure we calculate the average of the sum of each negative opinion word's degree, and because the degree of each negative word in sentiwordNet is between 0 and -1, the negativity of the tweet will also be between 0 and -1, and for making the negativity of the tweet between 0 and 1 we multiply it by -1.

The following formulas (1) and (2) show how we calculate the positivity and the negativity.

$$Positivity = \frac{\sum_{i=1}^{P} SentiWordNet(Wp_i)}{P} \quad (1)$$

$$Negativity = -1 \times \frac{\sum_{i=1}^{N} SentiWordNet(Wn_i)}{N} \quad (2)$$

Where:

- P is the number of positive opinion words in the tweet.
- N is the number of negative opinion words in the tweet.
- Wp_i is the i-th positive opinion word of the tweet.
- Wn_i is the i-th negative opinion word of the tweet.
- $SentiWordNet(Wp_i)$ calculate the sentimental degree of the positive opinion word Wp_i using SentiWordNet dictionary.
- $SentiWordNet(Wn_i)$ calculate the sentimental degree of the negative opinion word Wn_i using SentiWordNet dictionary.

The algorithm 1 shows the different steps followed to calculate the positivity and the negativity of the tweets.

With :

- **Translate(Tweet)** : is a function that translate each non-English tweet to English.
- **TextPreProcesssing(Tweet)**: Apply the different Text Preprocessing methods on the tweets to reduce the noise exist in it.
- **OpinionWord(Tweet)**: is a function that allows verifying if a word in a tweet is an opinion word or not. That is if such word is a verb, adjective or adverb.
- **SentiWordNet(Word)**: calculate the polarity of each opinion word of the tweet.

So at the end of this process, we find the crisp value of the positivity and the negativity. For example, if we want to classify a tweet T the first thing to do is the text preprocessing(TP) methods to extract the opinion words, suppose that after the TP methods step we find 4 opinion words that are: "happy, bad, exciting, sad", which are 2 positive words: "happy" with the degree 0.75 and "exciting" with the degree 0.89 from SentiWordNet, and 2 negative words: "bad" with the degree -0.68 and "sad" with the degree -0.57. By applying our method $the\ positivity = (0.75 + 0.89)/2 = 0.82$ and

Algorithm 1 Positivity and Negativity of tweets

Require: Positivity Value & Negativity Value
Require: tweet's class
P ← 0
N ← 0
c ← 0
d ← 0
if Tweet is not in English **then**
Tweet ← Translate(Tweet)
end if
Tweet ← TextPreProcesssing(Tweet)
SE[] ← Split(Tweet)
OW[] ← OpinionWord(SE)
for all word ∈ OW **do**
if SentiWordNet(word)>0 **then**
P ← P+SentiWordNet(word)
c ← c+1
else if SentiWordNet(word)<0 **then**
N ← N+(SentiWordNet(word) × -1)
d ← d+1
end if
end for
Positivity ← P/c
Negativity ← N/d

$the\ negativity = ((-0.68 - 0.57)/2) \times -1 = 0.625$. So for the tweet T, the degree of positivity is equal to 0.82 and the negativity is equal to 0.625.

E. Proposed Fuzzy Logic System

In this subsection, we describe our hybrid sentiment analysis approach based on SentiWordNet and the fuzzy logic system(FLS). As presented before the fuzzy logic system begins with a crisp value and after, fuzzify it using different steps(fuzzification, Rules inference). And finally, return a crisp value in the output using the defuzzification methods(centroid, Mean/, Max...). Figure 1 presents the general structure of a fuzzy logic system.

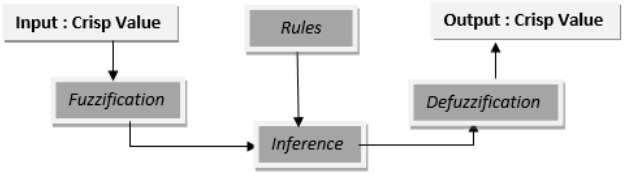

Fig. 1. Fuzzy Logic System

From figure 1, and as a comparison with our proposed approach, the input(crisp value) of the fuzzy logic system is the two measures (positivity and negativity) calculated with SentiWordNet, and the output(crisp value) is the class of the tweet(positive, negative or neutral).

As presented earlier, the first step is the definition of the input and the output variables of our proposed FLS. In our case and because we want to classify the tweets according

to three classes(positive, negative and neutral), we define two input variables: the **positivity** and the **negativity** of the tweets; and one output variable which is the class(sentiment) of the tweet.

In an FLS each variable either in the input or in the output is called linguistic variable, and each linguistic variable has a number of values that can take, this values called linguistic terms or the fuzzy sets. In our case, we have two linguistic variables in the input which are the positivity and the negativity, and each one has three linguistic terms which are **low, moderate and high**. This means that the positivity and the negativity variables can take three possible values, or in other words, can belong to three different fuzzy sets. In the same way, in the output, we have a linguistic variable which is the class of the tweet and it can also take three different linguistic terms which are positive, negative and neutral.

After we have defined the linguistic variables and their linguistic terms in the input and the output the next step of our FLS is the definition of the crisp values of the inputs with which we will begin our approach. For that and as explained in subsection 4.4, we use the SentiWordNet dictionary and the different text preprocessing methods to calculate the positivity and the negativity of the tweet that will play the role of input's crisp values.

*1) **Fuzzification Step**:* The next step after we calculate the crisp value of each input is the fuzzification step, in which we fuzzify the input variables using the membership function (MF) of each linguistic term. That is, calculating the degree of belonging of the input to each fuzzy set. In this work, we use two membership functions which are: trapezoidal-shaped MF and the triangular MF.

the trapezoidal-shaped MF is a function that depends on four scalar parameters a, b, c, and d, as given by the formula 3:

$$f(x) = \begin{cases} 0 & \text{si } x \leqslant a \\ \dfrac{x-a}{b-a} & \text{si } a \leqslant x \leqslant b \\ 1 & \text{si } b \leqslant x \leqslant c \\ \dfrac{d-x}{d-c} & \text{si } c \leqslant x \leqslant d \\ 0 & \text{si } d \leqslant x \end{cases} \tag{3}$$

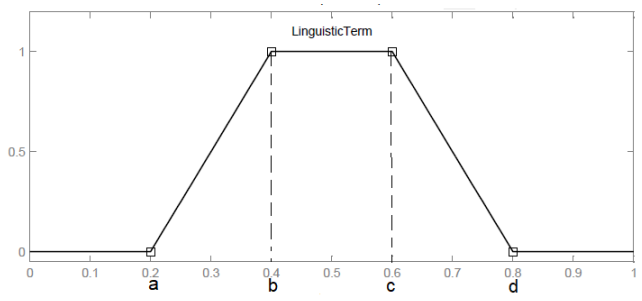

Fig. 2. Fuzzy Logic System

The figure 2 shows a trapezoidal-shaped MF for a linguistic term with a=0.2, b=0.4, c=0.6 and d=0.8.

On the other hand, the triangular MF is a function that depends on three scalar parameters a, b and c, as given by the formula 4 :

$$f(x) = \begin{cases} 0 & \text{si } x \leqslant a \\ \dfrac{x-a}{b-a} & \text{si } a \leqslant x \leqslant b \\ \dfrac{c-x}{c-b} & \text{si } b \leqslant x \leqslant c \\ 0 & \text{si } c \leqslant x \end{cases} \tag{4}$$

Figure 3 shows a triangular MF for a linguistic term with a=0.3, b=0.5 and c=0.7.

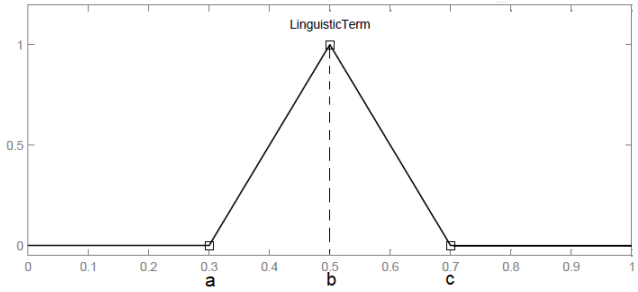

Fig. 3. Fuzzy Logic System

In our case, we need to fuzzify the input variables using one of the MFs presented earlier. For that, we have to define the MF of each linguistic term of the inputs. The linguistic variables "positivity" and "negativity" have three linguistic terms (three fuzzy sets), so we need to define three MFs, one for the fuzzy set "low", one for "moderate" and also another one for "high". The next step for calculating the MFs is the definition of the parameters a, b, c and d for each linguistic term. The choice of these parameters depends on the domain of application of the FLS, and also needs an expert in this domain. For example in our case, the values a, b, c, and d will be in the range [0;1], because the inputs "positivity" and "negativity" have values between 0 and 1 as explained in 4.4. The optimal values of these parameters are calculated empirically.

As we presented in section 3, if we want to work without the fuzzy logic concepts, we define for each linguistic term(low, moderate and high) a range between 0 and 1. For example, the positivity or the negativity is low if it is between 0 and 0.4, moderate if it is between 0.4 and 0.6 and high if it is between 0.6 and 1. So using the classical set each value of the positivity or the negativity is either belongs to a set(low, moderate or high) with a degree of belonging equal to 1, or not belongs to a set with a degree of belonging equal to 0.

Reality is always far from optimistic. Firstly, sentiment terms are fuzzy. The same word can explain different sentiment orientations even in the same domain. Secondly, sentiments of human are many times fuzzy. For instance, one may use one word to express more than one feeling at the same time. In this article, and to take into account the fuzziness of the sentiments, we use fuzzy logic to classify each tweet. That is, each value of positivity and negativity will belong to each fuzzy set with

a degree of belonging between 0 and 1. For that, we define the MFs for each linguistic term using the optimal values of the parameters(a, b, c...).

Figure 4 and 5 show respectively the Trapezoidal MF and Triangular MF for our two inputs: Positivity and Negativity.

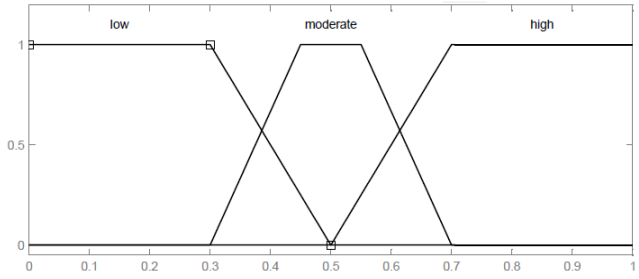

Fig. 4. Trapezoidal MF for the input variables

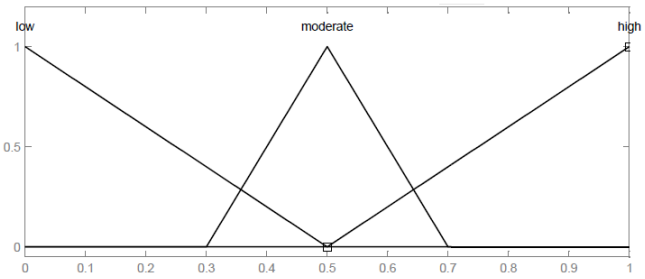

Fig. 5. Triangular MF for the input variables

From these two figures(4 and 5), for example for the linguistic variable moderate, the values of the parameters are : a=0.3, b=0.4, c=0.6 and d=0.7 using the Trapezoidal MF; a=0.3, b=0.5 and c=0.7 using the Triangular MF.

After the membership functions are defined for both input and output, the next step is to define the fuzzy control rules.

*2) **Rules Inference**:* The next step after the fuzzification of the inputs is the step of the definition and the application of the different rules of our problem. That is to say, combine membership functions with the control rules to derive the fuzzy output. We have two inputs and one output with three linguistic terms, for that we have defined nine fuzzy rules using the IF-THEN model with the AND logic operation between the values of the inputs. The nine rules of our FLS are the following:

- **IF** Positivity is low **AND** Negativity is low **THEN** Class is Neutral.
- **IF** Positivity is moderate **AND** Negativity is moderate **THEN** Class is Neutral.
- **IF** Positivity is high **AND** Negativity is high **THEN** Class is Neutral.
- **IF** Positivity is low **AND** Negativity is moderate **THEN** Class is Negative.
- **IF** Positivity is low **AND** Negativity is high **THEN** Class is Negative.
- **IF** Positivity is moderate **AND** Negativity is high **THEN** Class is Negative.
- **IF** Positivity is moderate **AND** Negativity is low **THEN** Class is Positive.
- **IF** Positivity is high **AND** Negativity is moderate **THEN** Class is Positive.
- **IF** Positivity is high **AND** Negativity is low **THEN** Class is Positive.

After the application of the different rules of our system, the next step is the implication of them to generate the value of each output term. In our case and because we use the AND operation between the inputs, the outputs take the minimum value between them. The last step in the rules inference is the aggregation of the results obtained for each output to find one value for each one. For example, for the output **neutral**, we will find three different values by applying three rules, and for finding the final value of the output "neutral", we calculate the maximum of these three values.

*3) **Defuzzification**:* After the fuzzification of the inputs and the application of the nine rules of our FLS, we find the degree of belonging of our output(class of the tweet) to each output fuzzy set(positive, negative and neutral), and to find the final result of our FLS that have to be in the form of a crisp value, we need to apply the defuzzification step.

The defuzzification process is meant to convert the fuzzy output back to the crisp or classical output to the control objective. The fuzzy conclusion or output is still a linguistic variable, and this linguistic variable needs to be converted to the crisp variable via the defuzzification process.

In this work we use 4 defuzzification techniques that are commonly used, which are:

- **Max-Membership principle:** this method calculates the maximum between the value of belonging of the output to each fuzzy set.
- **Centroid Method :** sometimes called Centre of area or centre of gravity, The Center of Gravity method (COG) is the most popular defuzzification technique and is widely utilized in actual applications. This method is similar to the formula for calculating the center of gravity in physics.
- **Weighted average Method :** This method is only valid for symmetrical output membership functions.
- **Mean-Max Method :** This method (also called middle-of-maxima) is closely related to the first method, except that the locations of the maximum membership can be non-unique (i.e., the maximum membership can be a plateau rather than a single point).

After we find the final crisp value of the output(CVO), the final step is to compare the result obtained with three range : tweet is negative if CVO is between 0 and 0.4, it is neutral if CVO is strictly greater than 0.4 and strictly less than 0.6, and it is positive if CVO is between 0.6 and 1.

F. schematization of our work

Figure 6 gives the different steps of our proposed approach :

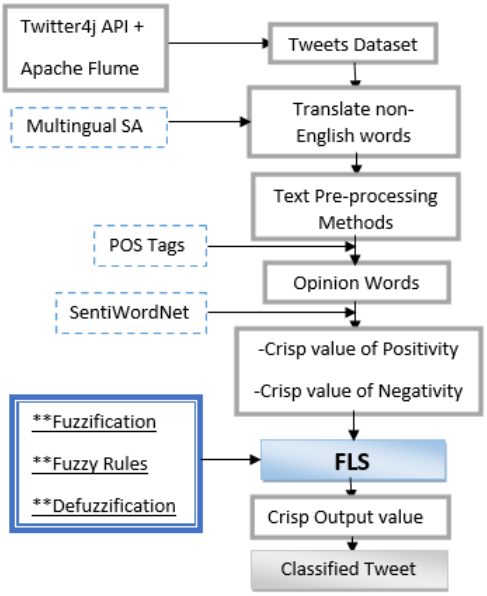

Fig. 6. Differents Steps of our approach

G. example of application

In this subsection, we describe with an example, how we classify a tweet according to three classes(Positive, Negative or Neutral) using our proposed approach. For that, we assume that we want to classify a tweet T. After the translation of T if it is written in another language than English, the application of the text pre-processing methods and after extracting the opinion words from it. The first step consists in calculating the input variables "positivity" and "negativity"(crisp values), by the proposed method described earlier using the lexicon-based dictionary SentiWordNet. Suppose that after all these steps we find that the crisp values of the inputs are: **positivity P=0.68** and **negativity N=0.35**.

After we calculate the crisp values for the inputs, the next step is the fuzzification of these crisp values. For that, we use the Trapezoidal-shaped MF presented in formula 3, and the results showed in figure 2. From all that, we calculate the degree of belonging of the positivity P and the negativity N to each fuzzy set(Low, High and Moderate) as the following :

- f(P,low)= **0**, because $P = 0.68 \geq d = 0.5$.
- f(P,moderate)=$\frac{d-P}{d-c} = \frac{0.7-0.68}{0.7-0.55} =$ **0.13**, because $c = 0.55 \leqslant P = 0.68 \leqslant d = 0.7$.
- f(P,high)=$\frac{P-a}{b-a} = \frac{0.68-0.5}{0.7-0.5} =$ **0.9**, because $a = 0.5 \leqslant P = 0.68 \leqslant b = 0.7$.
- f(N,low)=$\frac{d-N}{d-c} = \frac{0.5-0.35}{0.5-0.3} =$ **0.75**, because $c = 0.3 \leqslant N = 0.35 \leqslant d = 0.5$.
- f(N,moderate)=$\frac{N-a}{b-a} = \frac{0.35-0.3}{0.45-0.3} =$ **0.33**, because $a = 0.3 \leqslant N = 0.35 \leqslant b = 0.45$.
- f(N,high)= **0**, because $N = 0.35 \leqslant a = 0.5$

Note that the values of the parameters a, b, c, and d depends on the domain of application of the fuzzy logic and also on the range of definition of the inputs(in our case between 0 and 1). In our example of application, their optimal values are calculated empirically.

After the fuzzification step, it is the step of the application and the implication of our nine rules as presented below :

- IF (P is low)=0 AND (N is low)=0.75 THEN (Tweet is Neutral)=$\min(0, 0.75) =$ **0**.
- IF (P is moderate)=0.13 AND (N is moderate)=0.33 THEN (Tweet is Neutral)=$\min(0.13, 0.33) =$ **0.13**.
- IF (P is high)=0.9 AND (N is high)=0 THEN (Tweet is Neutral)=$\min(0.9, 0) =$ **0**.
- IF (P is low)=0 AND (N is moderate)=0.33 THEN (Tweet is Negative)=$\min(0, 0.33) =$ **0**.
- IF (P is low)=0 AND (N is high)=0 THEN (Tweet is Negative)=$\min(0, 0) =$ **0**.
- IF (P is moderate)=0.13 AND (N is high)=0 THEN (Tweet is Negative)=$\min(0.13, 0) =$ **0**.
- IF (P is moderate)=0.13 AND (N is low)=0.75 THEN (Tweet is Positive)=$\min(0.13, 0.75) =$ **0.13**.
- IF (P is high)=0.9 AND (N is moderate)=0.33 THEN (Tweet is Positive)=$\min(0.9, 0.33) =$ **0.33**.
- IF (P is high)=0.9 AND (N is low)=0.75 THEN (Tweet is Positive)=$\min(0.9, 0.75) =$ **0.75**.

The next step is the aggregation of these rules for each output fuzzy set.

- Tweet is neutral$\rightarrow \max(0, 0.13, 0) = 0.13$
- Tweet is Negative$\rightarrow \max(0, 0, 0) = 0$
- Tweet is Positive$\rightarrow \max(0.13, 0.33, 0.75) = 0.75$

After all these steps, we find the degree of belonging of the output to each output fuzzy set. And using the MFs of the output presented in figure 7 and the Centroid method, we defuzzify the output to find its final crisp value (class of the tweet) as presented in figure 8.

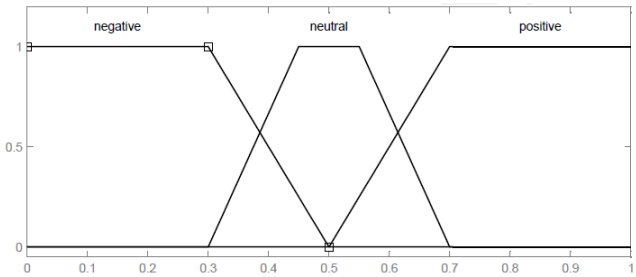

Fig. 7. Trapezoidal MFs for the output

After the Defuzzification step using the Centroid method and as presented in figure 8, the final crisp value obtained by applying our approach is equal to 0.759, and because this value is between 0.6 and 1, the tweet T is **positive**.

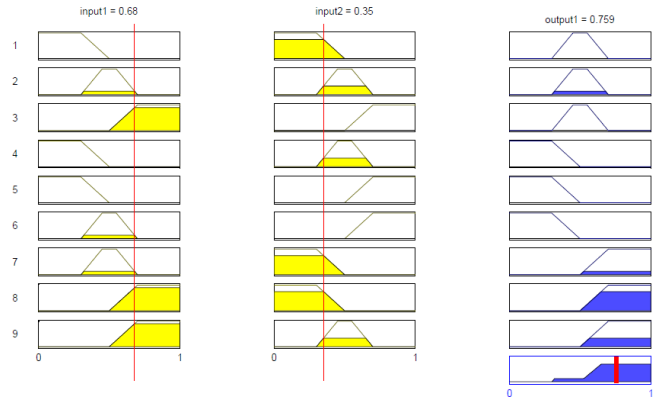

Fig. 8. Final result of our FLS

H. parallelization step

Another contribution of our work is the parallelization of our proposed approach using the Apache Hadoop framework, with its Hadoop Distributed File System(HDFS) and the programming model MapReduce. We use HDFS to distribute and to share the storage of the dataset of tweets between all machines of our cluster Hadoop. That is to say, after we collect the tweets either with the Twitter4j API or with Apache Flume, we store them in HDFS(if we use Apache Flume the tweets were directly stocked in HDFS, but if we use the Twitter4j API we need first to stock the retrieved tweets in a SQL database, and after using Apache Sqoop, we transform these tweets from SQL database to HDFS).

After we stock the tweets in HDFS, it is the step of the application of our approach, and because we want to parallelize the classification, we use the MapReduce programming model. The input of each iteration in the MapReduce algorithm is a tweet to classify, and the output is a classified tweet. the result of the classification of each tweet will be stocked in the HDFS.

Figure 9 shows the different steps for the parallelization of our work.

As presented in figure 9, we stocked the input of our work(dataset of tweets to classify) and the output (classified tweet) in the HDFS. The classification process using our proposed approach is done using the MapReduce programming model. The input of the MapReduce is a tweet, and after the application of all our steps on it, we get in the output the result of the classification(positive, negative or neutral).

Our MapReduce algorithm followed for the classification of the tweets with our proposed approach is presented in Algorithm 2:

With :

- **Fuzzification(Positivity AND Negativity)** allows fuzzifying the inputs using the crisp values of them and the trapezoidal membership function.
- **Implication(Fuzzy Rules) & Aggregation(Fuzzy Rules)** apply the different rules to find the fuzzy value of each output term(positive, negative or neutral).

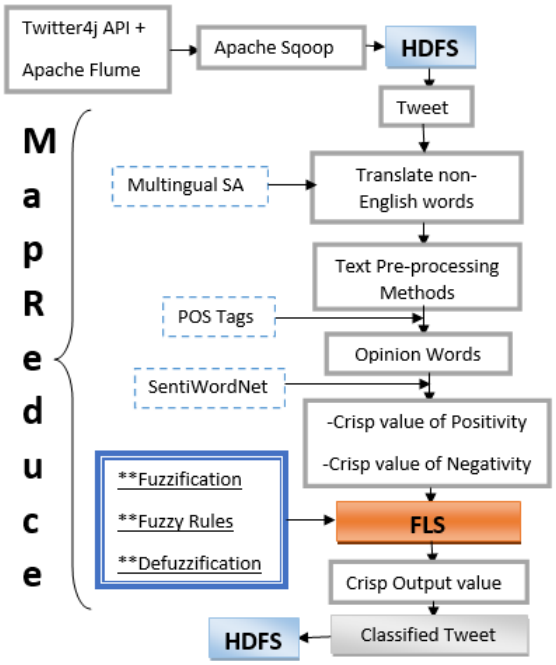

Fig. 9. Differents Steps for the Parallelisation

- **Defuzzification(FuzzyOutput)** return the final crisp value of the output(class of the tweet) using a defuzzification method.
- **write(tweet, sentiment)** is a MapReduce function that allows storing the result of the classification in HDFS in the form of key/value, the tweet to classify as a key of the MapReduce function and the class of the tweet (the result of the classification) as value, that is to say, we stock the result in HDFS as two columns one for the tweet and one for its class.

V. Experimental Results

In this section, we going to present some experimental results of our work. For that, we construct a dataset using Twitter4j API, this dataset contains the tweets published between January 2016 and January 2017 in relation or contain the word "iPhone". That is to say, our dataset contains the tweets about the word "iPhone". The tweets collected will be stored in HDFS using Apache sqoop.

As presented earlier, our FLS contains two input variables (positivity and negativity) and one output variable (class of the tweet). The range of each variable is between 0 and 1, and each one has three linguistic terms(linguistic values or fuzzy sets): low, moderate and high for the input variables; negative, neutral and positive for the output variable. Based on the value of the positivity and the negativity, we calculate the degree of belonging of each one to each input fuzzy set(low, moderate and high) using the membership functions. After that, we apply the different fuzzy rules to find the degree of belonging of the tweet's class to each output fuzzy set(Positive, Negative and Neutral). The final step which is the defuzzification(by

Algorithm 2 MapReduce programming model

Require: tweet's class
P ← 0
N ← 0
c ← 0
cc ← 0
if Tweet is not in English **then**
Tweet ← Translate(Tweet)
end if
Tweet ← TextPreProcesssing(Tweet)
SE[] ← Split(Tweet)
OW[] ← OpinionWord(SE)
for all word ∈ OW **do**
if SentiWordNet(word)>0 **then**
P ← P+SentiWordNet(word)
c ← c+1
else if SentiWordNet(word)<0 **then**
N ← N+(SentiWordNet(word) × -1)
cc ← cc+1
end if
end for
Positivity ← P/c
Negativity ← N/cc
Fuzzification(Positivity AND Negativity)
Implication(Fuzzy Rules)
FuzzyOutput ← Aggregation(Fuzzy Rules)
CrispOutput ← Defuzzification(FuzzyOutput)
if CrispOutput≥ 0.6 **then**
sentiment ← positive
else if CrispOutput≤ 0.4 **then**
sentiment ← negative
else if 0.4<CrispOutput< 0.6 **then**
sentiment ← neutral
end if
write(tweet,sentiment)

the application of a defuzzification method) gives us the final decision, that is if the tweet is negative, positive or neutral.

We use SentiWordNet for calculating the initial values(crisp values) for the inputs, two membership functions for the fuzzification that are: Trapezoidal MF and Triangular MF, nine IF-THEN rules and four defuzzification methods. From that, we have eight possible combinations for making a choice to which fuzzification/defuzzification methods used for the classification. Table 1 shows the result obtained for the classification and the error rate after the classification of the tweets using eight possible combinations : Trapezoidal MF/Max-Membership, Trapezoidal MF/Centroid, Trapezoidal MF/Weighted average, Trapezoidal MF/Mean-Max, Triangular MF/Max-Membership, Triangular MF/Centroid, Triangular MF/Weighted average, Triangular MF/Mean-Max.

From the table 1, the best fuzzification/defuzzification combination is the one in which we have used the Trapezoidal MF for the fuzzification and the Centroid method for the

TABLE I
CLASSIFICATION AND ERROR RATE USING DIFFERENTS FUZZIFICATION/DEFUZZIFICATION COMBINATIONS.

Fuzzification	Defuzzification	CR	ER
Trapezoidal MF	Max-Membership	62%	38%
	Centroid	84%	16%
	Weighted average	72%	28%
	Mean-Max	75%	25%
Triangular MF	Max-Membership	62%	38%
	Centroid	79%	21%
	Weighted average	72%	28%
	Mean-Max	72%	28%

defuzzification with a high classification rate(84%) and an error rate equal to 16%. Based on these results, we use in our proposed FLS the Trapezoidal-shaped MF for fuzzifying the inputs and the centroid method for finding the final crisp value of our system.

Another experiment of our work consists in demonstrating the effect of using the fuzzy logic on the classification of tweets, and how it can improve the results. For that, we compare our proposed approach(SentiwordNet+Fuzzy Logic) to an approach based only on Sentiwordnet. Figure 10 presents the results obtained for the classification rate (CR) and the error rate (ER).

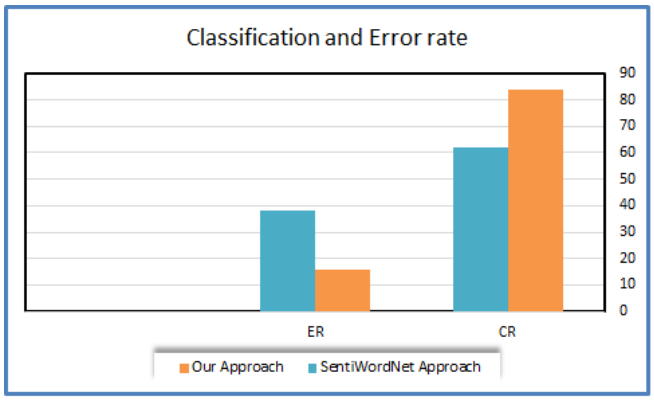

Fig. 10. The Effect of Fuzzy Logic on the Classification

From figure 10, we remark that our approach increases the percentage of the classification rate(from 62% to 84%) and decreases that of the error rate(from 38% to 16%) compared to the approach that uses only the SentiWordNet. That is to say, by adding the fuzzy logic we improve the quality of our system.

To demonstrate the results obtained using our proposed approach, we compare our method to some other techniques from the literature. Such as : a lexicon based approach method based on AFINN dictionary and WordNet [24], an approach based on semantic similarity that calculates the degree of relevance between the tweet and three opinion documents(Approach 1), an approach that calculates the semantic similarity between each word of the tweet and the words **negative** and **posi-**

tive(Approach 2) [25], and finally the approach based only on SentiWordNet.

For the evaluation of our approach, we have chosen two tweets datasets. The first provided by Twitter Sentiment Analysis Training Corpus, and it can be downloaded at http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/. This dataset contains approximately a million and a half classified tweets, each row is marked as 1 for positive sentiment, and 0 for negative sentiment. The second -which can help us for the evaluation of the neutral tweets- is the sentiment140 dataset provided by Go, A., Bhayani, R. and Huang, L. in their paper entitled "Twitter sentiment classification using distant supervision[11].", and it can be downloaded at https://www.kaggle.com/kazanova/sentiment140/downloads/traini This dataset contains 1,600,000 tweets extracted using the Twitter API. The tweets have been annotated (0 = negative, 2 = neutral, 4 = positive) and they can be used to detect sentiment(positive, negative or neutral).

Figure 11 shows the results obtained.

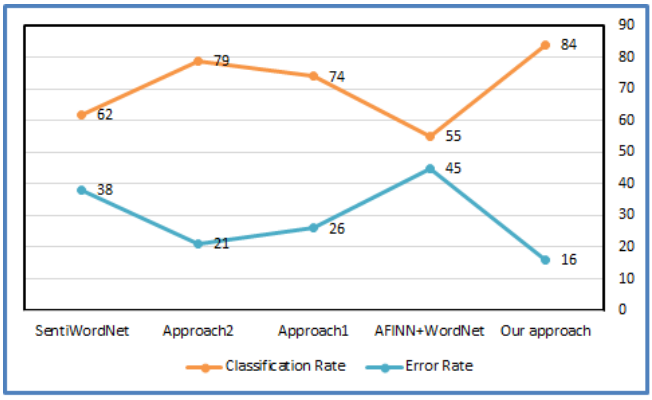

Fig. 11. Comparison Results

According to figure 11, our approach based on fuzzy logic and sentiWordNet outperforms the other methods with a classification rate equal to 84% and with 16% for the error rate, all that demonstrates how the fuzzy logic can improve the quality of the classification.

In recent years, A number of researchers start using hybrids approaches by combining various techniques (ML techniques, dictionary-based approaches...) than standard approaches with only one tool for improving the results of classification. Appel et al. [9] demonstrate that using a combination of sentiment lexicons, NLP tools, and fuzzy sets techniques outperform approaches that use only machine learning algorithms (which demonstrate our obtained results). Authors of this approach use the Sentiwordnet for extracting the polarity of words and after using a fuzzy logic inference with five fuzzy sets (Poor, Slight, Moderate, very, most). Also, Dragoni et al. [12] use also the fuzzy logic for calculating the polarity information based on dictionaries like senticNet. The last experiment we have done is by comparing our approach with these two recently published works which are similar to our method. Figure 12 presents a comparison using the precision and accuracy metrics between our approach, Appel approach, Dragoni approach, Naive Bayes NB approach and a maximum entropy ME approach.

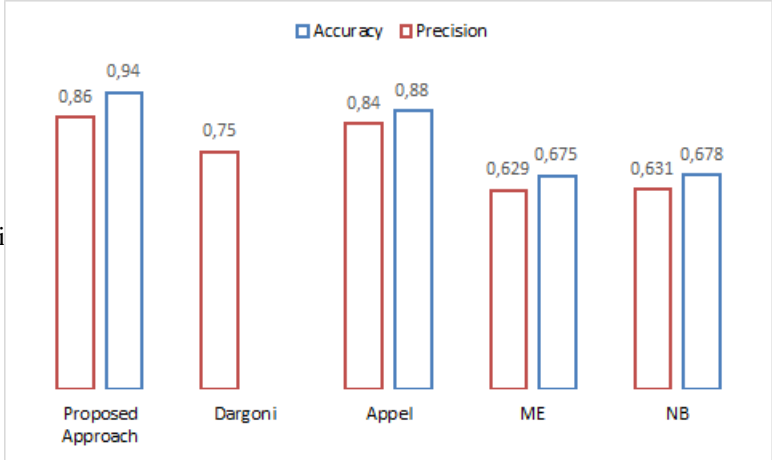

Fig. 12. Comparison Results 2

From figure 12, using approaches that use more than one technique(our approach, Appel, and Dargoni) give good results than the methods based on only on machine learning algorithms (NB and ME). The figure demonstrates also that our approach outperforms the Appel and Dragoni approaches with 94% for the accuracy and 84% as precision.

VI. Conclusion

In this paper, we have presented an approach based on fuzzy logic and the lexicon-based approach, using the dictionary SentiWordNet for classifying the tweets into three classes(positive, negative or neutral), this work gives us the opportunity to familiarise with the concepts of the fuzzy logic, and also how we can calculate the polarity of words from SentiWordNet, and how we can combine between the results obtained from SentiWordNet and the fuzzy logic system.

Our fuzzy logic system uses the results obtained by using SentiWordNet to give at the end the tweet's sentiment(positive, negative or neutral).

The proposed sentiment analysis method is multilingual, and it is developed in a parallel way using the Hadoop framework with the Hadoop Distributed File System(HDFS) and the MapReduce programming model.

The experimental results show that by using the fuzzy logic in the classification of tweets, we improve the quality of our method and that our approach outperforms some other techniques from the literature.

As perspectives of this work, our next article will be in the same way, that is to say, the classification of the social networks(Facebook, Twitter...etc) data using fuzzy logic but this time using the notion of information retrieval systems, and also using deep learning algorithms.

[11]https://pdfs.semanticscholar.org/52e2/bd533323ddf97073d034bae40a46eda55fd4.pdf

REFERENCES

REFERENCES

[1] Morente-Molinera, J. A., Kou, G., Peng, Y., Torres-Albero, C., & Herrera-Viedma, E. Analysing discussions in social networks using group decision making methods and sentiment analysis. *Information Sciences*, 447, 157168, 2018. doi:10.1016/j.ins.2018.03.020.

[2] Urea, R., Kou, G., Dong, Y., Chiclana, F., & Herrera-Viedma, E.. A review on trust propagation and opinion dynamics in social networks and group decision making frameworks. *Information Sciences*, 478, 461475, 2019. doi:10.1016/j.ins.2018.11.037.

[3] Morente-Molinera, J. A., Kou, G., Samuylov, K., Urea, R., & Herrera-Viedma, E. Carrying out consensual Group Decision Making processes under social networks using sentiment analysis over comparative expressions. *Knowledge-Based Systems*, 2018. doi:10.1016/j.knosys.2018.12.006.

[4] V. Chang, A cybernetics Social Cloud, *The Journal of Systems and Software*, 2016. http://dx.doi.org/10.1016/j.jss.2015.12.031.

[5] Chang, V., A proposed social network analysis platform for big data analytics ,*Technological Forecasting & Social Change*, 2017. https://doi.org/10.1016/j.techfore.2017.11.002.

[6] W. Medhat, A. Hassan, et H. Korashy, "Sentiment analysis algorithms and applications: A survey ", *Ain Shams Engineering Journal*, vol. 5, no 4, p. 1093-1113, dec. 2014,DOI: 10.1016/j.asej.2014.04.011.

[7] C. Catal et M. Nangir, "A sentiment classification model based on multiple classifiers", *Appl. Soft Comput.*, vol. 50, no Supplement C, p. 135-141, janv. 2017.

[8] A. Tripathy, A. Agrawal, et S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach", *Expert Syst. Appl.*, vol. 57, no Supplement C, p. 117-126, sept. 2016.

[9] Orestes Appel, Francisco Chiclana, Jenny Carter, Hamido Fujita, A Hybrid Approach to the Sentiment Analysis Problem at the Sentence Level, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.05.040.

[10] B. Wang, Y. Huang, X. Wu, et X. Li,"A Fuzzy Computing Model for Identifying Polarity of Chinese Sentiment Words", *Computational Intelligence and Neuroscience*, 2015. [En ligne]. Disponible sur: https://www.hindawi.com/journals/cin/2015/525437/.

[11] K. Wu, M. Zhou, X. S. Lu, et L. Huang, "A Fuzzy Logic-Based Text Classification Method for Social Media Data".*2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) Banff Center*, Banff, Canada, October 5-8, 2017.

[12] M. Dragoni et G. Petrucci, "A fuzzy-based strategy for multi-domain sentiment analysis", *International Journal of Approximate Reasoning*, vol. 93, p. 59-73, feb. 2018,DOI:10.1016/j.ijar.2017.10.021.

[13] J. B. Sathe and M. P. Mali, A hybrid Sentiment Classification method using Neural Network and Fuzzy Logic,*2017 11th International Conference on Intelligent Systems and Control (ISCO)*,p 93-96,jan 2017,DOI:10.1109/ISCO.2017.7855960.

[14] C. Jefferson, H. Liu, et M. Cocea, "Fuzzy approach for sentiment analysis", *in 2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, p. 1-6.

[15] H. Liu et M. Cocea, "Fuzzy rule based systems for interpretable sentiment analysis", *in 2017 Ninth International Conference on Advanced Computational Intelligence (ICACI)* , 2017, p. 129-136.

[16] X. Wang, H. Zhang, et Z. Xu, "Public Sentiments Analysis Based on Fuzzy Logic for Text ",*International Journal of Software Engineering and Knowledge Engineering*, vol. 26, no 09n10, p. 1341-1360, nov. 2016,DOI :10.1142/S0218194016400076.

[17] Dipak Damodar Gaikar, Bijith Marakarkandy, Chandan Dasgupta, (2015) "Using Twitter data to predict the performance of Bollywood movies", *Industrial Management & Data Systems*, Vol. 115 Issue: 9, pp.1604-1621, https://doi.org/10.1108/IMDS-04-2015-0145.

[18] Karyotis, C., Doctor, F., Iqbal, R., James, A., & Chang, V. A fuzzy computational model of emotion for cloud based sentiment analysis. Information Sciences, 433-434, 448463, 2018. doi:10.1016/j.ins.2017.02.004.

[19] Morente-Molinera, J. A., Kou, G., Pang, C., Cabrerizo, F. J., & Herrera-Viedma, E. An automatic procedure to create fuzzy ontologies from users opinions using sentiment analysis procedures and multi-granular fuzzy linguistic modelling methods. Information Sciences, 2018. doi:10.1016/j.ins.2018.10.022.

[20] L. Zadeh, "Fuzzy logic: A personal perspective", *Fuzzy Sets Syst.*, vol. 281, pp. 4-20, Dec., 2015.

[21] Y. Bai et D. Wang, "Fundamentals of fuzzy logic controlfuzzy sets, fuzzy rules and defuzzifications", *in Advanced Fuzzy Logic Technologies in Industrial Applications*, Springer, 2006, p. 17-36.

[22] Zadeh L. A. (1965) Fuzzy Sets. *Intl J. Information Control* 8:338-353.

[23] Y. Madani, J. Bengourram, et M. Erritali, "Social Login and Data Storage in the Big Data File System HDFS", *in Proceedings of the International Conference on Compute and Data Analysis*, New York, NY, USA, 2017, p. 91-97.

[24] Y. Madani, J. Bengourram, et M. Erritali, "A parallel Semantic sentiment analysis", *in Proceedings of the 3rd International Conference on Cloud Computing Technologies and Applications CloudTech'17*, Rabat, Morocco, Oct 24-26, 2017.

[25] Madani, Y., Erritali, M. & Bengourram, J. "Sentiment analysis using semantic similarity and Hadoop MapReduce", *Knowl Inf Syst* (2018). https://doi.org/10.1007/s10115-018-1212-z.