International Journal of Application or Innovation in Engineering & Management (IJAIEM)
Web Site: www.ijaiem.org Email: editor@ijaiem.org, editorijaiem@gmail.com
Volume 2, Issue 5, May 2013 ISSN 2319 - 4847

Social Networks' Text Mining for Sentiment Classification: The case of Facebook' statuses updates in the "Arabic Spring" Era

Safa Ben Hamouda[1] and Jalel Akaichi[2]

[1]BESTMOD-ISG
The University of Tunisia
Tunis, Tunisia

[2]BESTMOD-ISG
The University of Tunisia
Tunis, Tunisia

ABSTRACT

In recent years, text mining and sentiment analysis have received great attention due to the abundance of opinion data that exist in social networks such as Facebook, Twitter, etc. Sentiments are projected on these media using texts for expressing friendship, social support, anger, happiness, etc. Existing sentiment analysis studies tend to identify user behaviors and state of minds but remain insufficient due to complexities in conveyed texts. In this research paper, we focus on the usage of text mining for sentiment classification. Illustration is performed on Tunisian users' statuses on "Facebook" posts during the "Arabic Spring" era. Our aim is to extract useful information, about users' sentiments and behaviors during this sensitive and significant period. For this purpose, we propose a method based on Support Vector Machine (SVM) and Naïve Bayes. We also construct a sentiment lexicon, based on the emoticons, interjections and acronyms, from extracted statuses updates. Moreover, we perform some comparative experiments between two machine learning algorithms SVM and Naïve Bayes through a training model for sentiment classification.

Keywords: Sentiment analysis, classification, text mining, machine learning.

1. INTRODUCTION

Arab countries, which do not represent an exception for this rule, count a huge number of social network users. In fact, social networks become for Arab people a magic tool to promote freedom of speech, human rights and democracy. Social networks were the spark that contributes to the triggering of unexpected revolutions in some of the Arab countries. At the beginning, it was Tunisia that performed its revolution that laid the country into the "delightful" road of democracy. Then, it was the turn of Egypt, Yemen, Syria, etc., to taste the hard path of freedom.

On social networks interfaces, people had the opportunities to call for the change, by expressing their sentiments through a multiplicity of publications. In fact, statuses, commented images, videos, articles, etc. were shared on social networks to show dictators' crimes, inequalities between regions, etc. During the Tunisian revolution associated period, Facebook has become the collective source of information and one of the most important tools of communication for the Tunisian. At that time, Tunisian Facebook users had tendency to share their feelings, their thoughts and to inform their friends about conditions of their cities or neighborhoods, by sharing videos and pictures and especially posting short posts on walls which was the preferred way to interact with friends and by consequences to push them to care about what they think.

This paper explores the potential applications of text and sentiment mining techniques on statuses update in order to analyze the Tunisian's behavior during the revolution. For this purpose, we choose a random population having Facebook accounts. It includes males and females, students, workers, housewives, etc. The age of targeted population is varying between 21 and 54 years old. In fact, through the application of machine learning algorithms, we aim to identify the nature of the statuses update, and to link them to behaviors and sentiments characteristics. For that purpose, we created our own dataset and then we applied on it two machine learning algorithms: Naïve Bayes and Support Vector Machine. The expected output is to classify the extracted statuses into semantic classes useful, not only for people that aim to know themselves, but also for political decision makers.

The proposed paper is organized as follows. The state of the art and the background of social networks text mining are described in section 2. In section 3, we discuss the methodology of our work and describe the proposed architecture. In section 4, the process of our experiments is described and the results are discussed and evaluated. Finally, section 5 concludes and proposes possible directions for future works.

2. THE STATE OF THE ART

2.1 Text mining

Text mining is known as text data mining [1], or knowledge discovery from textual databases [2] or as an extension of data mining or knowledge discovery from structured databases [3]. In general, it refers to the process of extracting useful information by identifying and exploration of interesting and non-trivial patterns from unstructured text documents. Reference [2] mentioned that text mining uses techniques from information retrieval, information extraction and natural language processing. It also connects with the algorithms and methods of knowledge discovery from databases (KDD) [3], data mining [4], machine learning techniques [5] and statistics.

Therefore, text mining has been used to perform sentiment analysis in public forums, blogs and especially social network sites. In fact, textual information is categorized into two types: *facts*, which are objective expressions about entities, events and their proprieties, and *opinions* which are subjective expressions that describe people's sentiment toward entities, events and their properties [6]. Sentiment analysis, known as opinion mining as well, takes the written text and translates it into different context, such as positives, negatives or neutral.

2.2 Sentiment analysis

2.2.1 An overview of sentiment analysis

With the growing availability of opinion-rich resources such as social networks sites, blogs and forums new challenges arise as people actively use information technologies to seek out and understand the opinions of others in many domains such as asking about a particular product, politics, etc [7]. Instead of conducting costly market studies, customer satisfaction analysis or traditional surveys techniques, sentiment analysis provides companies offering products or services with means to analyze published reviews, to estimate the extent of product acceptance and to determine strategies to improve product or service quality. Sentiment analysis also facilitates policy makers or politicians to analyze public sentiments with respect to policies, public services or political issues.

Several subtasks can be identified within sentiment analysis:

- Determining document subjectivity: often called subjectivity classification: this subtask determines whether a giving text is objective (expressing a fact) or subjective (expressing an opinion or emotion) [8]-[9].
- Determining document orientation: often called sentiment classification or document-level sentiment classification: this subtask determines the polarity of a giving subjective text. In other word, determines whether this text expresses a positive or a negative sentiment on its subject matter [8]-[10].
- Determining the strength of document orientation: this subtask decides whether the positive sentiment expressed by a text on its subject matter is weakly positive, mildly positive or strongly positive [11]-[12].

2.2.2 Sentiment analysis' techniques

The sentiment analysis is a challenging field. During the few last years, it has attracted a lot of researchers. Therefore, we can find a variety of techniques of sentiment analysis in the literature. The two main ones are the machine learning approaches and the dictionary-based one.

a) Machine Learning techniques

The machine learning methods treat the sentiment classification problem as a topic-based text classification problem. Any text classification algorithm can be employed, such as Naïve Bayes, Support Vector Machine or Maximum Entropy, etc.

Reference [10] developed an unsupervised learning algorithm, known as Pointwise Mutual Information and Information Retrieval (PMI-IR), to classify texts as recommended or not recommended.

Reference [13] used three machine learning techniques (Naïve Bayes, classification maximum entropy, and SVM) to classify movie reviews as positive or negative. They tested different feature combinations including unigrams, unigrams + bigrams and unigrams + POS (part-of speech) tags, etc. Those techniques outperformed the human-generated baseline, and the SVM was the technique that gave the best result. Other examples of machine learning approaches in the sentiment analysis area are proposed such as regression models to predict a review's usefulness [14] and a semi-supervised method of performing a binary classification of texts as positive or negative [15]. Reference [16] investigated the utility of Naïve Bayes and SVM on political web logs and they showed that a Naïve Bayes classifier significantly outperforms SVM.

In [17] the authors applied simple online classifier Winnow to classifying polarity of documents. They showed that human agreement can merely achieve 75%-80% of precision and recall on polarity prediction. The recall obtained by Winnow is very poor, achieving only 43% for positive reviews and 16% for negative reviews.

Reference [18] conducted a comparative experiment on sentiment classification for online products reviews using the following classifiers: Passive-Aggressive (PA) Algorithm Based Classifier [19], Language Modeling (LM) Based Classifier [20], and Winnow Classifier. The results of their experiments showed that the Passive-Aggressive algorithm reached the higher accuracy (90, 07%) comparing to the others.
Reference [21] conducted a sentiment analysis of restaurant reviews by building a senti-lexicon, and proposed two improved versions of the Naïve Bayes algorithms. Then, they evaluated their performances by comparing them to the original algorithm Naïve Bayes and SVM. Results showed that the improved versions of Naïve Bayes proved effectiveness.

b) Dictionary-based approaches

Those approaches extract the polarity of each sentence in a document. Afterwards, the sense of the opinion words in the phrase is analyzed in order to classify the sentiment in the text. Generally speaking, the techniques that follow this approach are based on lexicons, and use a dictionary of words mapped to their semantic value [22].
The lexicon of a language is its vocabulary. The first version and the most well-known one is WordNet[1] [23] which is a semantic lexicon where words are grouped into sets of synonyms (called synsets). Another famous example of lexicon is SentiWordNet [24] which is an extension of WordNet. This one is a sentiment lexicon that represents an index of sentiment words, and it has the polarity information of the relevant word irrespective of whether it carries a positive sentiment or a negative one.

In this article, we perform text mining and sentiment analysis on a novel collection which represents Facebook's statuses updates Tunisian users in order to analyze sentiments and behaviors during the revolution of January 2011. Recent research on sentiment analysis [6], has focused on the mining of massive volume of texts with opinions and sentiments. Unlike most text, however, wall posts are comparatively short and they are probably the most popular Facebook

3. METHODOLOGY

In this sub section we propose a general architecture of our analysis (Fig.1). This framework is divided into 6 steps: raw data collection, lexicon development, data preprocessing, feature extraction and sentiment classification.

3.1 Raw data collection

This step consists of extracting statuses update (approximately 260 statuses) from Tunisian users during the Tunisian revolution (from 01-01-2011 to 01-06-2011). This is done through a web application called "I Told You[2]", and stores them in a custom database.

3.2. Lexicon development

This step focuses on the informal language of online social networks. For this reason, three types of lexicons were created: lexicon for social acronyms, lexicon for emoticons and lexicon for interjections.

3.2.1. Acronyms' lexicon

This lexicon denotes the most used acronyms by Tunisian Facebook users (Table.1).

TABLE 1: ACRONYMS'LEXICON

Acronym	Sentiment
LOL	Positive
GR8	Positive

[1] WordNet is a large lexical database of English. Nouns, verbs, adjective and adverbs are grouped into sets of cognitive synonyms (synsets) (Wordnet.com)
[2] www.IToldYou.com

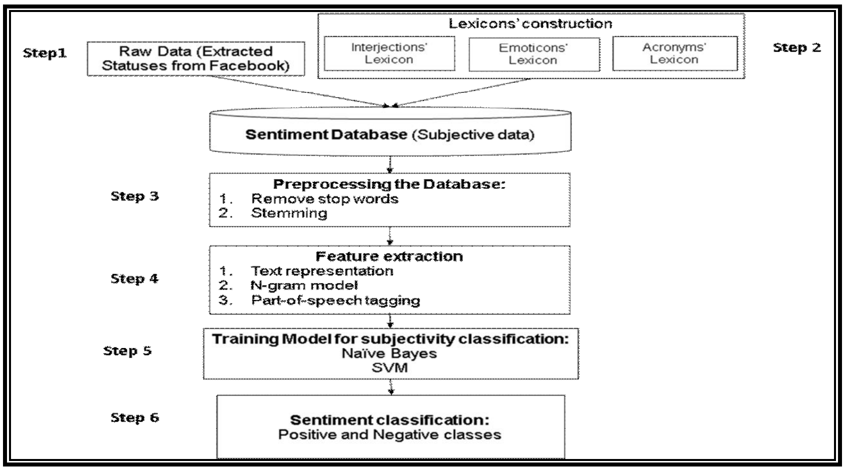

Figure 1 The proposed architecture

3.2.2. Emoticons' lexicon

We collected a lexicon of the most used emoticons on Facebook and annotate them manually whether they express a positive or a negative sentiment (Table 2).

TABLE 2: EMOTICONS' LEXICON

Emoticon	Sentiment
:)	Positive
:-)	Positive
:p	Positive
:(	Negative
:/	Negative
:'(	Negative
;)	Positive
:>	Positive
:-(	Negative
:*	Positive
:<	Negative
:-*	Positive
:-x	Positive
<3	Positive

3.2.3. Interjections' lexicon

This lexicon contains the main interjections[3] used in our data set and in Facebook in general (Table 3).

TABLE 3: INTERJECTIONS' LEXICON

Interjection	Sentiment
Wow, waw	Positive
Haha, hihi, hehe	Positive
Oh dear	Negative
Thank you	Positive
Help	Positive
No way	Negative
Oy	Positive

3.3. Data preprocessing

The process of sentiment text preprocessing is similar with the traditional text preprocessing of text classification. The goal behind preprocessing is to clean the dataset by removing words and punctuations that don't have an influence on sentiment classification. This increases the performance of the later the classification task. Therefore, the preprocessing is a primordial task. This involves two main parts: Removing stop words and Stemming.

[3] A list of English interjections is available on http://www.vidarholen.net/contents/interjections/.

- Removing stop words

The stop words are words that do not add meaningful content to the data set (i.e., pronouns, prepositions, conjunctions, etc). So, removing them reduces the space of the items significantly in the training and testing text. Examples of such words include 'the', 'a', 'of', 'and', 'to'. So the first step in preprocessing is removing these stop words.

- Stemming

Stemming is the process of removing prefix and suffix leaving the stem or the root of the word. For example, the words love, loved, loving and loves would be reduced to the root word love. The Porter stemmer [25] is a well-known algorithm for this task.

3.4. Feature extraction

The feature extraction is the process of extracting the main characteristics of the text. For a machine learning algorithm to perform well, it is essential to have features that are descriptive of the text. In our context, we need to be able to identify the words that express sentiment. Previous researches experimented with different features in the sentiment analysis and compare which features worked well and gave the best performance of the classification. The following features are discussed: text representation, the n-gram and the Part of speech (POS) tagging.

- Text representation

The common classifiers and learning algorithms cannot handle the emotional text directly. So, we have to represent in the form that classification algorithm can deal with. The documents are typically represented by feature vector. The two main approaches of document representation are the Bag-of-Words Model (BOW) and the Vector Space Model (VSM) [2].

In bag-of-words model, each word is represented as a separate variable having numeric weight.

Vector space model is now recognized as the best document representation model. Its basic idea is to simplify the document as a vector present in which feature term weight as component. The most popular weighting schema is term frequency inverse document frequency TFIDF [26]. However, reference [13] obtained better performance using presence rather than frequency. A binary-valued feature vectors in which the entries merely indicate whether a term is present (value 1) or not (value 0) formed a more effective basis for review polarity classification than did real-valued feature vectors in which entry values increase with the occurrence frequency of the corresponding term. We choose the presence representation because it yielded a higher accuracy representation than the standard frequency feature representation in previous related research.

- N-gram

One of the main steps of feature extraction is N-gram model [27]. The N-gram is a contiguous sequence of n words. The n-gram model consists of extracting a bag-of words representation of the text's field. Each feature corresponds to an N word, with the value of true if that feature is present and false otherwise. In this work, we explore the usage of unigrams, bigrams, trigrams, unigrams and bigrams and parts of speech as features.

A unigram is a single word, and a bigram is a pair of words that appear next to one another. Unigrams are the most typical type of text feature. Bigrams and trigrams may carry more information. In this master thesis, we extracted those features in order to evaluate their effectiveness in the context of sentiment analysis.

In [28] authors achieved better results with the combination of unigrams and bigrams than just unigrams or just bigrams respectively.

- Part-of-speech tagging

Part-of-speech is similar to the word bigrams discussed above except that instead of pairs of words, they are pairs of grammatical categories. The POS tagging consists of marking up a word in a text to a particular part of speech based on its context and its definition. We use POS as features because the same word may have many different meanings depending on its usage. For instance, "like" as a verb may have a positive connotation. "Like" may also be used as a preposition to cite an example, this does not carry a positive or negative connotation. In English, we have 9 parts of speech: noun, verb, article, adjective, preposition, pronoun, adverb, conjunction and interjection.

3.5. Creating a training model for the text polarity

Every classification task starts with a training dataset. Supervised learning algorithms usually require hand-labeled training data with a class (Positive or Negative). The training dataset should be rich and varied, providing the algorithm with a healthy cross section of the types of texts records for which classification may be needed in the future.

As for what percentage of the data should be used for training and what percentage for testing, it depends on the data set size. Typically 60% to 90% of instances are used for training and the remainder for testing. The more data there is

the more that can be used for training and still get statistically significant test predictions. In our context, we split our preprocessed data into two sets: 60% of our preprocessed data is used for the training set and 40% for the testing set.
We labeled our training set with POS if the status expresses a positive sentiment or NEG otherwise. Then, the task is to generate a *classification model* that is able to assign the correct class to a new status update.
To measure the performance of the classification model, we classify the testing set with the training model and compare the classified result (predicted labels by the model) with the true labels.

3.6. Classification using machine learning methods

We want to gauge the effectiveness of well-known sentiment classification algorithms on our novel corpus of Facebook statuses. We focus on two different machine learning techniques: Naïve Bayes and Support Vector Machine to determine their applicability in our context and compare their accuracies.

3.6.1. Naïve Bayes (NB)

A Naïve Bayes classifier is a probabilistic classifier based probability models that incorporate strong independence assumptions among the features. Our NB classifier assigns a given statuses the class. We choose to use a Naïve Bayes classifier because of its simplicity and its performance in previous studies on our chosen feature set representation [13, 16].

3.6.2. Support Vector Machine (SVM)

Support Vector Machine classification [29] method is a very effective way for classification, and its results are better than other classification algorithms, in general such Naïve Bayes and decision trees, etc. The aim of the SVM is to identify a hyper-plane that separates two classes of data. The chosen hyper-plane creates the largest margin between the two classes to make the points belonging to different classes and also make those points away from the hyper-plane as far as possible. In other word, using SVM classification method is equivalent to solving a constrained optimization problem.
We choose SVM as the classifier because of its often reported best performance and it has been adopted by many previous text classification studies [13]-[16]-[30]-[31].

4. EXPERIMENTS AND DISCUSSIONS

4.1. Experimental setting

Our experiments are divided into two phases: a Training phase and a Classification phase.
The training phase consists of building a training model. Based on this model, unlabeled statuses updates are classified in the classification phase. The preprocessing step eliminates the stop words and other unnecessary characters. The stemming process is performed as well. After that, a part of speech for each word from the training set is tagged. In the feature extraction step, the *n-gram* pattern made up of *n* word, as explained in section III, is extracted. We extracted 4 *n-gram* patterns:

- A unigram pattern: made up of one word for extracting a sentimental pattern.
- A unigram + bigram pattern: a combination of a unigram and bigram pattern.
- A unigram + trigram pattern: a combination of a unigram and trigram pattern.
- A unigram + bigram + trigram pattern: a combination of a unigram, a bigram and trigram pattern.

Once we have extracted the features in the training step, a training model is generated according to each algorithm. For the Naïve Bayes algorithm, a model is built by calculating the probability value based on the frequency of the extracted pattern. And for the SVM algorithm, a model is built by finding the optimal hyper-plane in order to divide the two classes.
Our set of experiments compares the two machine learning techniques by making different combinations of the extracted features.
To perform our experiments, we use the WEKA machine learning toolkit, version 3.6.8 [31]. We use the Naïve Bayes' implementation and we chose Naïve Bayes' multinomial event-driven model. For SVM, we use the SMO kernel implementation. SMO, sequential minimal optimization, breaks the large quadratic optimization problem into the smallest quadratic optimization problems, which can be solved analytically [13].
For the validation, we chose to use the validation technique that was used for all classifiers Stratified 10-fold cross validation. In stratified 10-fold cross validation, the training set is randomly divided into 10 sets with approximately equal size with additional constraint that the fold distribution are similar to the original distribution of the classes. For each *fold*, the classification process is applied 10 times with one of the folds as test and the remaining (10 - 1= 9) folds as training set. This procedure is repeated for each of the 10 groups. The cross-validation score is the average performance across each of the 10 training runs [16].

4.2. Results and discussions

To evaluate the performance of sentiment classification, we use the standard classification performance metric used in previous information retrieval and text classification studies: The accuracy (1), the precision (2), the recall (3) and the F-measure (4).

$$Accuracy = \frac{a+d}{a+b+c+d} \qquad (1)$$

$$\Pr ecision = \frac{a}{a+b} \qquad (2)$$

$$\operatorname{Re} call = \frac{a}{a+c} \qquad (3)$$

$$F - measure = \frac{2 * \Pr ecision * \operatorname{Re} call}{\Pr ecision + \operatorname{Re} call} \qquad (4)$$

Where:
- *a:* the number of statuses *correctly assigned* to this class.
- *b*: the number of statuses *incorrectly assigned* to this class.
- *c:* the number of statuses *incorrectly rejected* to this class.
- *d:* the number of statuses *correctly rejected* to this class.

We compared the performances of different feature sets using Naïve Bayes and SVM classifiers. We chose 60% of our dataset as training data and the remaining 40% as testing data. We used 10-fold cross validation to conduct the evaluation. Table 4 summarizes the accuracy for all seven feature sets. We highlighted in bold font the best accuracy.

TABLE 4: CLASSIFIER'S ACCURACY

Features	**#of features**	**Accuracy NB (%)**	**Accuracy SVM (%)**
Unigrams	610	72.15	**74.68**
Unigrams + Bigrams	1629	**74.05**	**75.31**
Unigrams+ Trigrams	1655	71.33	71.97
Unigrams + Bigrams + Trigrams	2688	72.61	71.97

Comparing to unigram features, and when we use unigrams and bigrams as features, the accuracy improved for Naïve Bayes (from 72.15% to 74.05%). There was an increase for SVM (from 74.68% to 75.31%). In reference [13], there was a decline for both Naïve Bayes and SVM. And when we use unigrams and trigrams as features, and comparing to unigrams and bigrams there was a decrease of the accuracy for Naïve Bayes (from 74, 05% to 71, 33 %) and SVM (from 75, 31% to 71, 97 %).
Moreover, when we compare the unigrams features to the combination of the three n-gram as features, there is a slight improvement for Naïve Bayes' accuracy (from 72, 15% to 72, 61%), but a decrease of SVM's accuracy (from 74.68% to 71, 97%)
The two classifiers reached their highest accuracies when a combination of unigrams and bigrams is used as features. However the lowest accuracies were reached when we use a combination of unigrams and trigrams. Therefore, using the combination n-gram (unigram + bigram) as a feature tends to be the best in terms of accuracy for Naïve Bayes and SVM.
In addition to, we found out that the SVM's algorithm outperformed the Naïve Bayes in all cases. This confirms the previous published researches [13]-[28]. The SVM reached its highest accuracy (75, 31%) in the case of using the combination of unigrams and bigrams as features.
In table 5, we zoomed into the performance of NV and SVM in the case of using the combination of unigrams and bigrams as features by calculating the precision, the recall and the F-measure.
Using NB, a Facebook status that has a positive sentiment is correctly identified as such with 74, 1% recall. However for SVM, the recall is less than the NV's one, which means a Facebook status that has a positive sentiment is correctly identified as such with 71, 2% recall.
For SVM, a Facebook status given a positive classification is 89.2 % likely to be correct. But a Facebook status given a negative classification is only 60% likely to be correct.

Although the accuracy of SVM seems good (75.31%), the precision and recall indicate that the numbers are confusing. This is clearly visible in the F-measure rates (89.2% for Positive and 60% for Negative). One possible explanation for these results is that the model in this case study was built with many more positive statuses than negative ones, and the test data contains mostly positive statuses. Therefore, we can conclude that most of the published statuses on Facebook during the revolution have a positive sentiment. This would be surprising due to the rough conditions that the users have been enduring.

TABLE 5: RESELTS FOR UNIGRAMS + BIGRAMS USING NB AND SVM

	NB			SVM		
Polarity	**Precision (%)**	**Recall (%)**	**F-measure (%)**	**Precision (%)**	**Recall (%)**	**F-measure (%)**
Positive	75	74.1	75	89.2	71.2	89.2
Negative	73.1	74	73.1	60	83.3	60

5. CONCLUTION AND FUTURE WORKS

In this paper, we have investigated the utility of sentiment classification on a novel collection of dataset which is Tunisian Facebook users. The originality of this collection leads not only on the nationality of the users, but also on the period of posting their statuses updates which is the Tunisian revolution. This period was very special and unique for them, so their wall posts are with no doubt unique and encouraging to analyze.

Using the most well-known machine learning algorithms, we conducted a comparative experimental procedure between the Naïve Bayes and the SVM algorithms by combining different feature extractors. Those algorithms can achieve high accuracy for classifying sentiment when combining different features. Although Facebook statuses have unique characteristics compared to other corpuses (Reviews, News, etc), machine learning algorithms are shown to classify statuses with similar performance.

Finally, the overall performance of the proposed methodology is satisfactory, however, we would like to further improve our research by tracking changes within people's sentiment on a particular topic, explore the time dependency of our data and analyze their trendy topics dynamically. It would be very interesting to involve the temporal feature on this kind of analysis and not to focus solely on previous posts or discussions.

References

[1.] M. A. Hearst, "Text data mining: Issues, techniques, and the relationship to information access", Presentation notes for UW/MS workshop on data mining, 1997.

[2.] R. Feldman, and J.Sanger, The Text Mining Handbook: Advanced Appraoches in Analyzing Unstructered Data, 1st ed., New York: Cambridge University Press, 1995.

[3.] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, From Data Mining to Knowledge Discovery: An Overview, Advances in Knowledge Discovery and Data Mining, MIT Press, 1-36, Cambridge,1996.

[4.] D, Hand, H. Mannila, and P. Smyth. "Principles of Data Mining", MIT Press. Cambridge, MA. . 2001

[5.] T. Mitchell. "Machine Learning", McGraw Hill Cambridge, MA, 1997.

[6.] B. Liu, "Sentiment Analysis and Subjectivity", Invited Chapter for the Handbook of Natural Language Processing, Second Edition. March, 2010

[7.] B. Pang, and L. Lee, "Opinion Mining and sentiment analysis". Foundations and Trends in Information Retrieval, 2(1-2): 1-135, 2008.

[8.] B. Pang, and L. Lee, "A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on Minimum Cuts", In: Proceedings of the 42nd Meeting of the association for Computational Linguistics (ACL'04), Barcelona, ES, 2004, pp. 271–278.

[9.] T. Wilson, "Fine-grained subjectivity and sentiment analysis: Recognizing the intensity, polarity, and attitudes of private states", University of Pittsburgh, 2008.

[10.] P.D. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews", In Proceedings of the 40th annual meeting on association for computational linguistics. Philadelphia, Pennsylvania, 2002.

[11.] T. Wilson, J. Wiebe, and R. Hwa, "Just how mad are you? Finding strong and weak opinion clauses", In: Proceedings of the 21st Conference of the American Association for Artificial Intelligence (AAAI'04). San Jose, US., 2004, pp. 761–769.

[12.] A-M. Popescu, and O. Etzioni, “Extracting product features and opinion from reviews”, In Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, 339-346. Vancouver, British Columbia, Canada: Association for Computational Linguistics, 2005.

[13.] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques”. In P. Isabelle (Ed), In Proceeding of conference on empirical methods in natural language, Philadelphia, US, 2002, pp.79-86, Association for Computational Linguistics.

[14.] X. Zhang, and F. Zhu, “The influence of Online consumer reviews on the demand for experience goods: The case of video games”, In 27th international conference on information systems (ICIS), Milwaukee,. AISPress, 2006.

[15.] A. Esuli, and F. Sebastiani. “Determining the semantic orientation of terms through gloss analysis”, In: Proceedings of the 14th ACM International Conference on Information and Knowledge Management (CIKM’05), Bremen, DE, 2005, pp.617–624.

[16.] K. T. Durant, and M. D. Smith, “Mining Sentiment Classification from Political Web Logs”, WEBKDD’06, Philadelphia, Pennysylvania, USA, 2006, ACM 1-59593-444-8.

[17.] M. Hurst and K. Nigam. “Retrieving topical sentiments from online document collections”, in Document Recognition and Retrieval XI, 2004, pp. 27–34.

[18.] A. Hang, M. Vibhu, and D. Mayur, “Comparative Experiments on Sentiment Classification for Online Product Reviews” American Association for Artificial Intelligence (www.aaai.org), 2006.

[19.] Y. Shalev-Shwartz, and A. Singer, “Online and batch learning of pseudo-metrics”, In Proceedings of the 21st International Conference on Machine Learning, 2004.

[20.] A. D. Manning, and H. Sch¨utze, “Foundations of Statistical Natural Language Processing”, Cambridge (Massachusetts) and London (England): The MIT Press, 1999.

[21.] H. Kang, S.J. Yoo, and D. Han, “Senti-lexicon and improved Naive Bayes algorithms for sentiment analysis of restaurant reviews”, Expert Systems Applications 39, 6000-6010, 2012.

[22.] K. Denecke, “Using SentiWordNet for Multilingual Sentiment Analysis”, Proceeding IEEE 24th International Conference Data Eng. Workshop (ICDEW 2008), IEEE Press, 2008, pp. 507-512.

[23.] A. Fellbaum, Wordnet: An Electronic Lexical Database, MIT Press, 1998.

[24.] B. Esuli, and F. Sebastiani, “Determining term subjectivity and term orientation for opinion mining”, In Proceedings EACL-06, the 11rd Conference of the European Chapter of the Association for Computational Linguistics, 2006.

[25.] M. F. Porter, “An algorithm for Suffix Stripping”, Program 14(3), July 1980, pp. 130-137.

[26.] A. Sebastiani, “Machine Learning in automated text categorization”, ACM Computing Surveys, 34: 1-47, 2002.

[27.] B. Carpenter. “Scaling high-order character language models to gigabytes”, In proceedings of the 2005 association for computational linguistics software workshop, 2005, (pp. 1-14. Ann Arbor: ACL

[28.] A. Go, L. Huang, R. Bhayani, “Twitter sentiment classification using distant supervision”, In: CS224N Project Report, Stanford, 2009.

[29.] T. Joachims, “Text categorization with support vector machines: Learning with many relevant features”, In Proceedings of the 10th European Conference on Machine Learning Springer, Berlin, 1998, pages 487–494.

[30.] R. Zheng, J. Li, H. Chen, and Z. Huang, “A Framework for Authorship Identification of Online Messages: Writting-Style Features and Classification Techniques”, J.Am. Soc. Information Science and Technology (JASIST), vol. 57, no. 3, 2006 , pp.378-393.

[31.] A. H. Witten, and E. Frank, “Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations”, Morgan Kaufmann, 2000.

AUTHOR

Safa Ben Hamouda a master student at the Higher Institute of Management of Tunis, will receive the Master degree in Business Intelligence from the Higher institute of Management of Tunis in June 2013 and has received the fundamental license of computing science of management in 2010 from the Higher Institute of Management.

Jalel Akaichi received his PhD in Computer Science from the University of Sciences and Technologies of Lille (France) and then his Habilitation degree from the University of Tunis (Tunisia) where he is currently an Associate Professor in the Computer Science Department. Jalel Akaichi has published in international journals and conferences, and has served on the program committees of several international conferences and journals. He is currently the Chair of the Master Science in Business Intelligence. Jalel Akaichi visited and taught in many institutions such as the State University of New York, Worcester Polytechnic Insitute, INSA-Lyon, University of Blaise Pascal, University of Lille 1, etc.