Opinion polarity detection in Twitter data combining sequence mining and topic modeling

Asma Ouertatani[1], Ghada Gasmi[2], and Chiraz Latiri[3]

[1]LIPAH, ENSI, University of Manouba, Tunis, Tunisia, [2]LISI, INSAT, University of Carthage, Tunis, Tunisia, [3]LIPAH, FST, University of Tunis El Manar, Tunis ,Tunisia

Abstract. We propose a pipeline process to analyze opinion about festivals and cultural events by automatically detecting polarity in Twitter data. Previous studies have focused in the polarity classification of individual tweets. However, to understand the polarity of opinion on a domain, it is important to find themes or topics that occur in the corpus.

The first phase is to find the optimal number of topics and to identify the major topics via the latent Dirichlet analysis (LDA) topic model. The second stage is to detect polarity in tweets using the sequence mining approach mainly founded on sequences extracted from tweets using a LCM-SEQ algorithm [9]. The results showed that the polarity detection accuracy of the sequence mining was 84.78%, indicating that the proposed method was valid in most cases.

Keywords: topic modeling, LDA, opinion analysis, sequence mining

1 Introduction

With the advent of web 2.0 and social network service evolution, users generated a massive amount of information stored in unstructured online reviews that can not simply be used for further processing by computers. Various researchers have conducted analyses focusing on the exchange of opinions that occurs on social network platforms.

Twitter is an online social network where users post and interact with messages, "*tweets*", restricted to 140 characters.

However, discovering sentiments and opinions through manual analysis of a large volume of textual data is extremely difficult. For that reason, specific preprocessing methods and algorithms are needed in order to mine useful patterns. Hence, in recent years, there have been much interests in the natural language processing community to develop novel text mining techniques with the capability of accurately extracting users' opinions from large volumes of information like Twitter data.

Among various opinion mining tasks, one of them is polarity analysis, i.e. whether the semantic orientation of a text is positive or negative, which focuses on classifying the polarity of individual texts (e.g., web reviews or tweets) by selecting important features through methods such as n-grams [10, 11], word subsequence [12], information gain [6], and recursive feature elimination [1]. When applying machine learning to opinion classification, most existing approaches rely on supervised learning models trained from labeled corpora where each document has been labeled as positive or negative prior to training. A tweet is then classified via algorithms, such as the nave naïve, maximum entropy [11], or support vector machine (SVM) algorithms. However, sentiment classification models trained on one domain might not work at all when moving to another domain. Furthermore, in a more fine-grained opinion classification problem (e.g finding users′ opinions for a particular film festival), topic detection and opinion classification are often performed in a two-stage pipeline process, by first detecting a topic and later assigning a polarity label to that particular topic.

We propose a pipeline process to analyze opinion about festivals and cultural events by automatically detecting polarity in Twitter data. Previous studies have focused on the polarity classification of individual tweets.

However, to understand the polarity of opinion on a domain, it is important to find themes or topics that occur in the corpus. Our goal here is to find the optimal number of topics and to identify the major topics via the latent Dirichlet analysis (LDA) topic model. The second stage detects polarity in tweets using the sequence mining approach mainly founded on sequences extracted from tweets using a LCM-seq algorithm.

The remainder of this paper is organized as follows. Section 2 details the proposed method, which includes a data-preprocessing step; Section 3 presents the analysis results; and Section 4 presents the conclusion of this study and discusses directions for future research.

2 Proposed method

2.1 Preprocessing

The MC2@CLEF2017 lab has released a collection of 70 000 000 microblogs over 18 months dealing with cultural events [7]. Microblogs are in all languages. We used just 5 000 000 tweets from the collection.

Simple and intiutive techniques in the preprocessing phase were evoked as removal links, twitter identifiers, pontuations and stop words.

Clearly cannot be performed without knowing the underlying language detection. Therefore, modern text processing tools heavily rely on highly effective algorithms for language. We employed the Cavnar and Trenkle [5] approach to text categorization based on character n-gram frequencies that have been particularly successful.

We used the implementation in the R extension package textcat aims at both exibility and convenience. After the preprocessing phrase we chosed the first 320000 english tweets to be our dataset. Figure 1 presents a words cloud from our dataset. The word cloud principle is based on a text analysis method that allows us to highlight the most frequently used keywords (like : music, Film..)

in a text paragraph.

Fig. 1. words cloud from our dataset

2.2 Topic modelling

Topic modeling is a type of statistical model in natural language processing that aims to find topics in a corpus, group topics together by looking for similarity and co-occurence, and categorize documents in the corpus based on the topic probabilities assigned.
We are specifically using a statistical method called the latent Dirichlet allocation (LDA). Latent Dirichlet Allocation (LDA) is one of the most popular topic models [3]. In the context of LDA, a topic is composed of terms with creation probabilities. For each term position in a document, LDA identifies a topic, and the topic is composed of the terms included in the topic, measured probabilistically. Given a set of documents, LDA provides an algorithm that learns the topics and the terms associated with each topic. LDA requires one input parameter: the number of topics to extract.
And now the question then arises as: What is the best way to determine k (number of topics) in topic modeling?

Optimal number of topics for LDA model :
Before going right into generating the topic model and analysing the output, we need to decide on the number of topics that the model should use. We used 3 metrics to estimate the best fitting number of topics:

- Method based on the harmonic mean :
  This method has first been applied by Griffiths and Steyvers [8].
  We calculated the harmonic mean of a the values sets of $p(w|z, k)$. The model that we will retain by varying k will be the one which will have the highest

value.
z : Per word topic assignment.
w : word.
k : number of topic.

- Density-based method [4]
  The principle is to calculate the similarity (or distance) between all pairs of themes for different models obtained by varying the number of themes. Themes are more independent if the similarity between themes is small.
- Method based on the Kullback-Leibler divergence (KL) [2]
  The measure of divergence is a measure of how the topic1 distribution for document m and the word distribution for topic1 diverges from a second topics expected probability distribution.

The optimal k is the one with the lowest divergence. The three methods required to train multiple LDA models to select one with the best performance. So, the best way is to calculate all metrics at once, the figure 2 represents the Results calculated on the whole dataset:

The three methods agree that somewhere between 75 and 100 topics is optimal for this dataset. To find the best value of the number of topics hyperparameter k we used the perplexity measure for the applicability of a topic model to new data and the 5 folds cross validation over the range of k [75..100]. Perplexity is a measure of how well a probability model predicts a sample. We opted to fit a model with 85 topics. In the figure 3 the plot of the results:

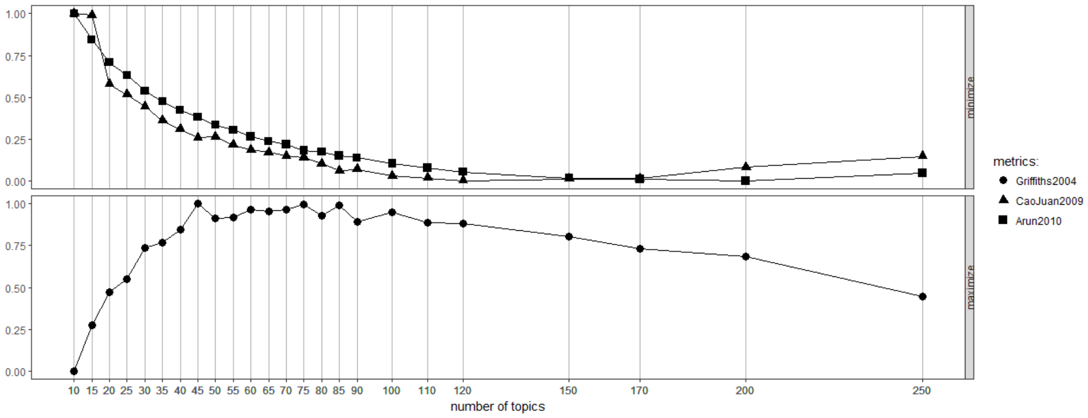

Fig. 2. number of topics

Terms are assigned to a topic with probabilities, so every term in the corpus is given a probability per topic. However, we can use the top terms to get a sense for what each topic covers. Figure 4 shows the topics names. For the second stage of our approach we used the films topic.

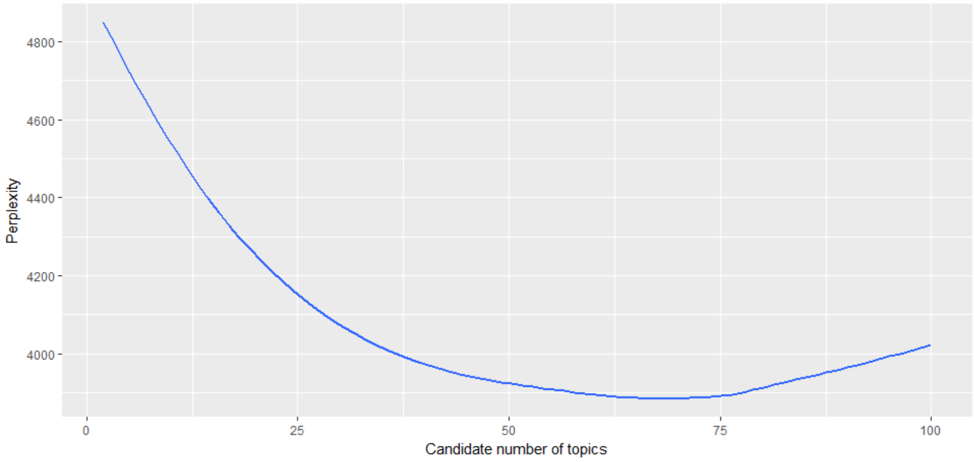

Fig. 3. 5-fold cross-validations of a topic modeling

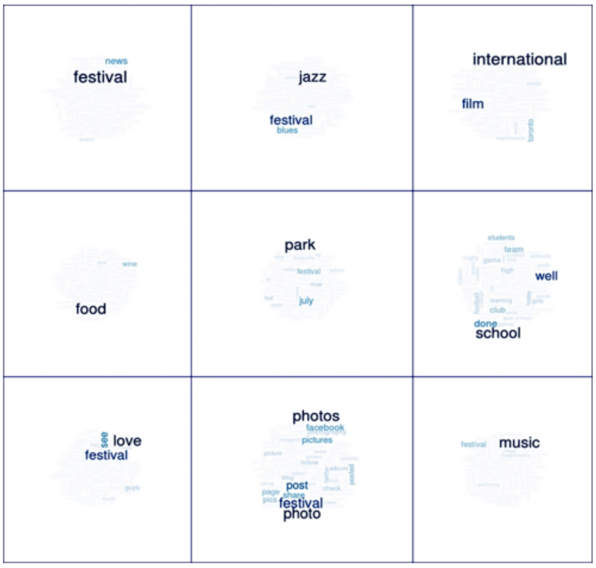

Fig. 4. topics names

2.3 Polarity detection

Before starting the phase of analysis of polarity one must go through the stage of the analysis of subjectivity to remove the objective tweets of our collection. To do this, we used the subjectivity lexicon [1], and N-gram as features and the

[1] http://mpqa.cs.pitt.edu/lexicons/subj-lexicon

naïve bayes as classifier.
For the polarity detection, we used lexique Wordnet sentiment, $Tf * idf$ and the algorithm LCM-SEQ [2] to extract all frequent item sequences. to use it as features.

Lcm-seq : is an efficient algorithm for enumerating frequent sequence patterns from a sequential database. In addition to its high speed, LCM-SEQ can be applied in a variety of ways, as it can assign a positive or negative weight to each sequence and only extract frequent sequence patterns that appear in a specified window width [9].
For a vocabulary V, the set of finite sequences on V is expressed by V^*. A sequence pattern is an arbitrary sequence $s = a_1....a_n \quad V^*$, and $P = V^*$ expresses the set of all sequence patterns on V. The sequence database on V is the sequence set $S = s_1, ..., s_m$. We denote the the size of S by $|S|$. For sequence pattern $p \in P$, a sequence database including p is called an occurrence of p. The denotation of p, denoted by $\theta(p)$ is the set of the occurrences of p. $|\theta(p)|$ is called the frequency of p, and denoted by$Freq$. For given constant $\alpha \in N$, called a minimum support, sequence pattern p is frequent if $Freq(p) \geq \alpha$. In our approach, we used a value min sup equal to 100.

3 Results

3.1 Experimental validation

For the phase of the subjectivity analysis we used as a training corpus introduced in Pang/Lee ACL 2004 [3] we used the Subjectivity lexicon and N-gram as features.
For the polarity detection we used the sentiment140 data as a training data [4], and we used the frequent item sequences as features for naïve bayes classifier.

3.2 Evaluation protocol

As evaluation meteric we used the classifier Accuracy.
The accuracy can be defined as the percentage of correctly classified instances :

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{1}$$

Where TP, FN, FP and TN represent the number of true positives, false negatives, false positives and true negatives, respectively.
The following table illustrates the results for the naïve bayes classifier :

[2] http://research.nii.ac.jp/uno/code/LCM-SEQ.html
[3] http://alias-i.com/lingpipe/demos/tutorial/sentiment/read-me.html
[4] http://help.sentiment140.com/for-students]

	accuracy
Subjectivity lexique	75.14%
N-gram	80.2%
Subjectivity lexique + N-gram	81.5%

Table 1. subjectivity analysis results

	accuracy
lexique wordnet	75.14%
$Tf * idf$	79.80%
frequent item sequences	82.5%
All	84.78%

Table 2. polarity detection results

4 Conclusion

The polarity detection aims to automatically classify the customer opinion and provide comprehensive understanding of customer feedback from raw data on the Web. In all of the social network platforms, Twitter has been one of the most popular sources for marketing information research and sentiment classification. The work described in this paper is a step towards efficient classification of tweets using the topic modelling.

References

1. A. Abbasi, H. Chen, S. Thoms, and T. Fu. Affect analysis of web forums and blogs using correlation ensembles. *IEEE Transactions on Knowledge and Data Engineering*, 20(9):1168–1180, 2008.
2. R. Arun, V. Suresh, C. Veni Madhavan, and M. Narasimha Murthy. On finding the natural number of topics with latent dirichlet allocation: Some observations. *Advances in Knowledge Discovery and Data Mining*, pages 391–402, 2010.
3. D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
4. J. Cao, T. Xia, J. Li, Y. Zhang, and S. Tang. A density-based method for adaptive lda model selection. *Neurocomputing*, 72(7):1775–1781, 2009.
5. W. B. Cavnar, J. M. Trenkle, et al. N-gram-based text categorization. *Ann Arbor MI*, 48113(2):161–175, 1994.
6. T.-Y. Chu, J. Lu, S. Beaupré, Y. Zhang, J.-R. Pouliot, S. Wakim, J. Zhou, M. Leclerc, Z. Li, J. Ding, et al. Bulk heterojunction solar cells using thieno [3, 4-c] pyrrole-4, 6-dione and dithieno [3, 2-b: 2, 3-d] silole copolymer with a power conversion efficiency of 7.3%. *Journal of the American Chemical Society*, 133(12):4250–4253, 2011.
7. L. Ermakova, L. Goeuriot, J. Mothe, P. Mulhem, J.-Y. Nie, and E. Sanjuan. CLEF 2017 Microblog Cultural Contextualization Lab Overview (regular paper). In *Experimental IR Meets Multilinguality, Multimodality, and Interaction, CLEF,*

Dublin, Ireland, 11/09/2017-14/09/2017, volume 10456 of *Lecture Notes in Computer Science*, http://www.springerlink.com, 2017. Springer.

8. T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences*, 101(suppl 1):5228–5235, 2004.
9. T. Nakahara, T. Uno, and K. Yada. Extracting promising sequential patterns from rfid data using the lcm sequence. In *Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, 2010.
10. A. Pak and P. Paroubek. Twitter as a corpus for sentiment analysis and opinion mining. In *LREc*, volume 10, 2010.
11. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.
12. R. Xia, C. Zong, and S. Li. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences*, 181(6):1138–1152, 2011.