Opinion argumentation based on combined Information Retrieval and topic modeling

Seif Sendi[1] and Chiraz latiri[2]

[1] ISAMM, University of Manouba, Tunis, Tunisia
sendiseif@gmail.com
[2] LIPAH, FST, University of Tunis El Manar, Tunis ,Tunisia

Abstract. Argumentation mining is a text mining task, which aims at automatically detecting the argumentative structure concealed in a huge amount of text data. Previous researches in this field have focused on the classification of text sentences as arguments and the detection of relations between them. Different corpora have been used, such as newspaper articles and online debates. Due to the explosion of social networks, microblogging platforms like Twitter and Facebook have become interesting tools to evaluate public opinion on different domains.

In this work, we propose a new pipeline process to achieve the goal of argumentation mining, based on 70 millions of twitter-microblogs released from MC2 CLEF-2018 lab dealing with cultural events. Our approach is based on Information Retrieval protocol combined with sentiment analysis and topic modeling using Latent Dirichlet Allocation (LDA).

Keywords: Opinion mining, topic modeling, information retrieval.

1 Introduction

Social network became an important source of information considering the number of users and the data exchanged. Twitter is one of the most famous platforms in the world, millions of people express themselves every minute by simply sharing a 140 characters short text (*tweet*).SS The first question which should be asked is can we use this large data ? The answer is yes, thanks to the TwitterAPI, people can have a free access to twitter's data. But that's not enough, unless we know what we are looking for.

Dealing with such data requires precautions : We must understand that a normal person can't analyse (manually) a huge volume of text data, that's why we need to ask computers to do it for us, using specific methods and algorithms. The analysis of twitter text data must be different from the other types of text, because tweets are pressed, noisy and often unstructured.

Our purpose is to explore twitter data and to identify the argumentative parts of it, but first we need to know how can we consider a tweet as an argument.

Mochales Palau and Moens [1] consider an argument as a set of premises, pieces of evidence (e.g. facts), offered in support of a claim. The claim is a proposition, an idea which is either true or false, put forward by somebody as true.
Based on this definition, we want to extract the most ranked argumentative tweets to create their summary which present the relevant information about a given topic.

To achieve our goal, we create a pipeline containing three popular techniques : Information Retrieval, sentiment analysis and topic modeling. Each technique has it's own particularities; Information Retrieval, based on the indexing and the querying, allows a simple searching process of the whole data to get the result of the query. Sentiment analysis aims to categorize opinions about something (product, event, etc.). Topic modeling allows the discovery of hidden semantic topics into the text data.
We believe that combining these techniques can achieve the aim of argumentation mining, by identifying the most argumentative tweets within a 70 millions microblogs dataset of cultural events. The results confirm that our pipeline succeeded to identify the most 100 argumentative tweets compared to the baseline.

2 Proposed approach

The MC2 CLEF2018's data contains 18 text files, each one represents a collection of tweets which contains 16 topics in different languages during one month. Each topic is a festival name, there are 12 festivals in english and 4 in french.
Since our goal is to extract the most 100 argumentative tweets of each topic, we created a pipeline and we put these 18 files through it. This pipeline is composed of information retrieval, topic modeling and sentiment analysis.

2.1 Information retrieval

Information retrieval is the process of obtaining relevant information about a collection of text data, we choose the Indri platform to make our IR process. A specific format called TRECText (or TRECWeb) is required, any other format won't be recognized. As our input files are text documents, we need to get the correct format, that's why we have used Perl language to do the job. We wrote a program which accepts a *.txt* file as input and return a *.trec* file as output without changing the content of our tweets. Example of TRECText format :

```
<DOC>
<DOCNO>600364867825082368</DOCNO>
<TEXT>Taubira huée au festival de Cannes</TEXT>
</DOC>
<DOC>
<DOCNO>600364266265387009</DOCNO>
<TEXT>Christiane #Taubira nétait pas à sa place au Festival de #Cannes</TEXT>
</DOC>
```

Creating the index of our text data allows us to do the querying, for each topic we wrote a lexicon-based query in Indri language, example :

Art festival # weight[m](1.0 # band(Art festival) 0.5 # or(abnormal aborted))

We demand the engine to return the best 500 tweets for each topic based on their scores which have been calculated by default with Indri. It uses a query likelihood function with Dirichlet [2] prior smoothing to weight terms. The formulation is given by :

c(w;D) = count of word in the document

c(w;C) = count of word in the collection

$| D |$ = number of words in the document

$| C |$ = number of words in the collection

numerator = c(w;D) + mu * c(w;C) / $| C |$

denominator = $| D |$ + mu

$$\text{score} = \log(\text{ numerator / denominator })$$

By default, mu is equal to 2500, which means that for the very small documents you're using, the score differences will be very small. [3]

We have made this process twice, one for english topics and one for french, so we have two result files which are considered as our baseline.

2.2 Topic modeling

The baseline from the previous step does not confirm that the 500 resulting tweets are talking about the same topic because it is an index-quey process, the system is not intelligent enough to provide a list of tweets which have exactly the same topic simply by a lexicon query. That is why we need to use a machine learning technique to train a model that confirms the matching result.

Topic modeling is a modern technique frequently used in machine learning and natural language processing, it aims to extract topics from a collection of text documents. It helps in understanding our data, organizing it, and discovering hidden information into it. There are many methods that are used to obtain topic models, in this work we focused on Latent Dirichlet Allocation (LDA).

LDA is a popular method of topic modeling which considers each document as a blend of topics, and each topic as a blend of words. We imagine that we have the following tweets (documents):

- Tweet 1 : Jazz festival is the best, i can't wait for the next year.
- Tweet 2 : Rock festival is always amazing !!
- Tweet 3 : Jazz music is relaxing !!
- Tweet 4 : I don't like rock music.

The LDA model discover the topics in these tweets, it may produce these results :

- Topic 1 : 50 % Festival, 15% Jazz, 15% Rock.. (This topic deals with festivals)
- Topic 2 : 50 % Music, 15% Jazz, 15% Rock..(This topic deals with music)
- Tweet 1 and 2 : 100% topic 1
- Tweet 3 and 4 : 100% topic 2

The hard part when working with LDA is how to find the number of topics. In general, the use of metrics, like Griffiths2004 [4] or Arun2010 [5], is required to calculate the fitting optimal number of topics. But in our case, this number is known (see previous section).

The preprocessing phase is needed because of the noise of tweets, we have removed punctuations, URLs, stop words (english and french), twitter identifiers and we have done the stemming process. The figure below presents a wordcloud of our tweets after cleaning, and it contains the most frequently words.

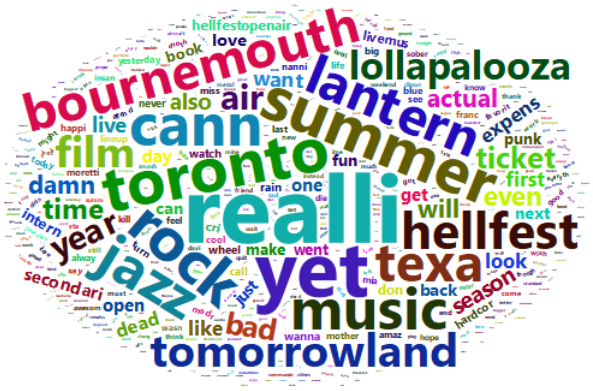

Fig. 1: Word cloud of our corpus

After cleaning our tweets, we have used the LDA function provided by the *topicmodels* package in R. This function was fed with our baseline results.

As LDA supports two methods, VEM and Gibbs Sampling, we need to choose which fits our model.

The estimates for the topic distributions for the documents are included which are the estimates of the corresponding variational parameters for the VEM algorithm and the parameters of the predictive distributions for Gibbs sampling. In additional slots the objects contain the assignment of terms to the most likely topic and the log-likelihood which is $\mathbf{log\ p(w} \mid \alpha, \beta\mathbf{)}$ for LDA with VEM estimation and log $\mathbf{p(w \mid z)}$ for LDA using Gibbs sampling.[6]

The estimation using Gibbs sampling requires specification of values for the parameters of the prior distributions, that's why we choose VEM.

The aim is to score each tweet by how much it belongs to his expected topic. We can examine the per-document-per-topic probabilities called **Gamma**. This will result k number of clusters or groups of words, we manually annotate these groups to do the matching between score and topic for every single tweet.

2.3 Sentiment analysis

Sentiment analysis is a text classification tool that uses NLP and machine learning to analyse text and extract the writer's sentiment about a given topic, it is widely used in online debates and social media platforms. Since our data is a collection of tweets dealing with cultural events, we can apply the sentiment analysis on it to discover what people feel about each topic.

The same baseline is considered as our input, and since we have cleaned the tweets (see previous section), the data is ready to be analysed. We have used the R *sentiment* package, which perform sentiment analysis by scoring individual words based on the affinity score defined by NRC [3].

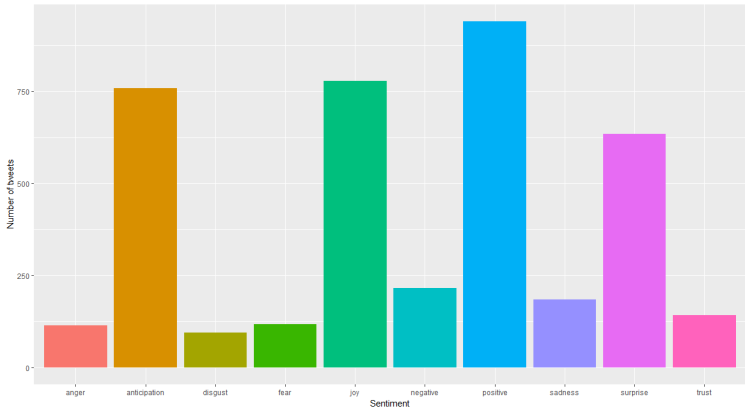

Fig. 2: Histogram of sentiments.

[3] The NRC Emotion Lexicon is a list of English words and their associations with eight basic emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive).

To get a sentiment score of each tweet in our corpus, we applied the *get nrc sentiment()* function. This function doesn't stop at providing a positive or negative score for a tweet, it goes deeper to tell us the other emotions that a tweet contains (fear, joy, ..), in addition, it counts the frequency of each emotion word. With these sentiments, we can calculate a new score which reflects how the writer of the tweet have felt.

3 Results

After finishing the process, we have three scores for each tweet, as our goal is not particularly one of them, we need to put these scores together. The new calculated score is the key to get a new ranking.

$$\text{Score} = IR_{score} + LDA_{score} + Sentiment_{score}$$

Based on this score, we re-ranked the tweets. To test our approach, we need to compare the new results with the baseline. We have picked the top 3 tweets from Toronto festival, and located the same tweets in the baseline ranking.

1	It is music festival time #Toronto #festival #love my city	-1.326510	↑
2	Could the Toronto Film Festival get any more liberal? Doubt it	-1.412866	↑
3	Is it even a Toronto music festival if there isn't torrential downpour	-1.554710	↑

Fig. 3: Sample of the new rank

42	It is music festival time #Toronto #festival #love my city	-3.18689	↓
26	Could the Toronto Film Festival get any more liberal? Doubt it	-3.14673	↓
11	Is it even a Toronto music festival if there isn't torrential downpour	-3.09191	↓

Fig. 4: Sample of the baseline rank

The whole ranking is changing after the process, besides w didn't show the 500 tweets of each topic, we just take the best 100 between them. It's remarkable that the top ranked tweets have become more argumentative than before.

4 Conclusion

Argumentation mining is a new research field which aims to automatically extract the argumentation structure hidden in a collection of text data. Previous studies have been through the classification of sentences into argument and non-argument, based on a manually annotated corpus and using static machine learning algorithms.
In this paper, we tried to achieve the same goal with a different style, because we think that annotating a corpus of tweets is a hard task, besides it is not accurate due to human errors. We tried to combine computational techniques that made our results more specific, especially with the ranking method. Our model provides an easy result list which contains a summary of ranked tweets according to their scores.

References

1. Raquel Mochales and Marie-Francine Moens. 2011 : Argumentation mining. Artificial Intelligence and Law.

2. ChengXiang Zhai and John Lafferty. A study of smoothing methods for language models applied to information retrieval. ACM Transactions on Information Systems, 2004.

3. Paul Ogilvie and Jamie Callan,2012 : Experiments using the Lemur toolkit.

4. Thomas L. Griffiths and Mark Steyvers, 2004 : Finding scientific topics

5. Karthik Arun, 2010 : Optimal Number of topics for LDA

6. Bettina Grun, Johannes Kepler and Kurt Hornik : Topicmodels: an R package for fitting topic models