Topic Modeling and Sentiment Analysis with LDA and NMF on Moroccan Tweets

Nassera Habbat(✉), Houda Anoun, and Larbi Hassouni

RITM Laboratory, CED ENSEM Ecole Superieure de Technologie Hassan II University, Casablanca, Morocco
nassera.habbat@gmail.com, houda.anoun@gmail.com, lhassouni@hotmail.com

Abstract. Twitter is one of the most popular social media platforms. Due to its simplicity of use and the services provided by twitter API, it is extensively used around the world, including Morocco. It provides huge volume of information and is considered as a large source of data for opinion mining.

The aim of this paper is to analyze Moroccan tweets, in order to generate some useful statistics, identify different sentiments, and extract then visualize predominant topics. In our research work, we collected 25 146 tweets using Twitter API and python language, and stored them into MongoDB database. Stored tweets were preprocessed by applying natural language processing techniques (NLP) using NLTK library. Then, we performed sentiment analysis which classifies the polarity of twitter comments into negative, positive, and neutral categories. Finally, we applied topic modeling over the tweets to obtain meaningful data from Twitter, comparing and analyzing topics detected by two popular topic modeling algorithms; Non-negative Matrix Factorization (NMF) and Latent Dirichlet Allocation (LDA). The observed results show that LDA outperforms NMF in terms of their topic coherence.

Keywords: Twitter · Moroccan tweets · Sentiment analysis · Topic modeling · NLP · Python · MongoDB · LDA · NMF

1 Introduction

With the rapid development of Web 2.0, a large number of users publish messages on social networks to express their opinions about different topics.

Twitter is one of the most popular social media platforms. It plays a fundamental role in the diffusion of information. In fact, about 500 million tweets are published every day and around 200 billion tweets are posted every year [1]. As for Morocco, there are 453.5 thousand active users monthly in twitter [2].

Twitter allows its users to post short messages limited to 288 characters. It is considered as an important source for understanding people's emotion (Sentiment analysis) and discovering the most discussed topics (Topic modeling). In this paper, we focused on Moroccan tweets. We collected 25 146 tweets published by Moroccan users from 26 April 2020 to 08 June 2020 and stored them in MongoDB database. After analyzing

M. Ben Ahmed et al. (Eds.): SCA 2020, LNNS 183, pp. 147–161, 2021.
https://doi.org/10.1007/978-3-030-66840-2_12

them, we managed to stand out some useful statistics, identify the sentiments contained in these tweets, and extract the most discussed topics in tweets comparing two of the most popular methods: Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF).

In this paper we make the following contributions:

- Generation of useful statistics from collected tweets
- Sentiment analysis on tweets
- Topic modeling on Moroccan tweets comparing LDA and NMF

So, the two big research questions which will be answered through data analysis are:

- What are the feelings expressed through collected Moroccan tweets? And which are the most discussed subjects in these tweets?
- What is the best topic model according to our use case?

For that, we organized our paper as follows. Section 2 gives a brief literature review. Methods and tools are described in Sect. 3. In Sect. 4, we present our experiments results. We end with a conclusion where we summarize this paper and outline our future work.

2 Literature Survey

2.1 Sentiment Analysis

There are several researches discussing opinion mining in Social Media especially Twitter, which contains posts (Tweets) written in different languages.

In some researches, the authors focused on sentiment analysis of tweets written in one language. For example, in [3], they analyzed English tweets and classified them into three categories (Positive, Negative, and Neutral tweets) using two classifiers Naïve-Bayes and K-Nearest Neighbors (K-NN). They assumed that K-NN is the best classifier in their situation, because it gives more accurate predictions than Naïve-Bayes. Furthermore, the authors in [4] worked on sentiment lexicon for sentiment analysis of Saudi dialect tweets (SaudiSenti) and compared it to Large Arabic sentiment dictionary (AraSenti). For that, they used a dataset of tweets previously labeled, that comprises 5400 tweets dealing with various topics. To evaluate their experiments, they used a dataset of 1500 tweets in modern standard Arabic (MSA) and Saudi dialects and distributed over three categories: positive, negative and neutral, The results showed that because AraSenTi identified most of the neutral tweets either as positive or negative, it was outperformed by SauDiSenti in terms of the precision, recall, and F measure.

Other researches, focused on sentiment analysis of tweets in relation with an event. For example, in [5], authors analyzed tweets during the 10 days of United Nations Climate Change Conference in Paris in 2015 (COP21). In their analysis, they used data collected between 30 November and 09 December 2015 (a total of 1,602,543 tweets comprising keywords in relation with COP21). They discussed the top Twitter accounts based on in-degree, out-degree, the number of tweets by day etc. In [6], the Paris attacks event that took place on November 2015 is analyzed. In this work, the authors evaluated the information spread around the world based on geo-located tweets about this event.

2.2 Topic Modeling

In addition to sentiment analysis on twitter, some researchers have analyzed content of tweet (topic modeling) using different methods. In [7], authors compared three methods: Latent Semantic Indexing (LSI), Non-negative Matrix Factorization (NMF), and Latent Dirichlet Allocation (LDA). In their experiments, they adopted the dataset provided at the GitHub website, scraped from the websites of two Hindi newspapers named Amar Ujala and Navbharat, and they found that NMF model performed little better than LDA model which was better than LSI, using perplexity and coherence as metrics to evaluate the topic models.

We outline below, some researches which compared two of the most popular topic modeling; LDA and NMF. The authors in [8] used the TC-W2V measure that is generally more sensitive to changes in the top terms used to represent topics and they found that NMF consistently achieves higher coherence scores than LDA in the case of 94.7% of all 300 experiments (results for $k = 10$ and $k = 50$ topics) and in their study, they limited on 210,247 English EP (European Parliament) speeches from the official website of the EP. In addition to the metrics already mentioned, there are other tools for evaluation of topic models, such us the intrinsic and extrinsic scores using UCI and UMass, which compared in [9], using Wikipedia as an external resource (its index occupies 27 Gb of disk space) and 20 newsgroup as a local resource (its index occupies 38M). In their research, they shown that intrinsic evaluation (concerning 20 newsgroups) had better performance than extrinsic using Wikipedia as an external resource, mainly because of the index size and disk usage, regarding speed, UCI performed 6% to 8% faster in comparison to UMass. In general, UMass and UCI executed reasonably well with both providing satisfactory results when estimating the correlation with human evaluation scores.

In [10], the authors combined two text mining techniques (sentiment analysis and topic modeling), to analyze 16 million tweets collected using the streaming API, during 51 days from September to October 2019. The authors used Valence Aware Dictionary and sEntiment Reasoner (VADER) for sentiment analysis around Brexit with stock prices and British pound sterling. To discover the most popular daily topics of discussion on Twitter using "Brexit" as keyword, they applied LDA model for topic modeling. The same algorithms were used in [11] to analyze tweets relating to climate change and compare the results over time between different countries. Those results showed that the USA is behind in its discussion of topics concerning policies and plans to address climate change compared to the UK, Canada, and Australia.

3 Tools and Methods

In this section, we present our implemented architecture. Then, we describe different tools used to generate statics from twitter data and perform sentiment analysis and topic modeling.

3.1 Proposed Architecture

Figure 1 details the proposed architecture of our system. Our system is based on many layers; each layer performs a task ranging from collecting data (tweets) to visualization of results. These layers will be described in the next sections.

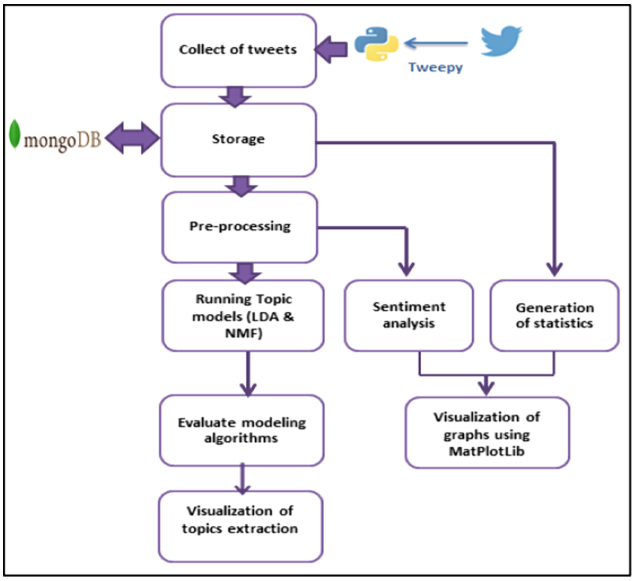

Fig. 1. Proposed architecture

As shown in the figure above, Twitter data has been collected and stored into MongoDB database using tweepy and pymongo libraries. We only selected tweets geolocated in Morocco, then those tweets were preprocessed, to generate some statistics, perform a sentiment analysis, and visualize topics extraction comparing two topic models: LDA and NMF. Different tools and algorithms used in this architecture are described below.

3.2 Sentiment Analysis and Statistics of Twitter Data

Tweepy

Tweepy [12] is an open source Python library that gives a very convenient way to access the Twitter API with Python.

To represent Twitter's models and API endpoints, Tweepy includes a set of classes and methods, and transparently manage various implementation details, such as:

- Data encoding and decoding
- HTTP requests
- Results pagination
- OAuth authentication
- Rate limits
- Streams

MongoDB

MongoDB [13] is a distributed, universal, document-based database. It is classified

as a NoSQL database program; it has been designed by MongoDB Inc. For modern application developers, and licensed under the Server Side Public License (SSPL).

A database in mongoDB is organized into collections, and each collection contains documents. In our case, the tweets are stored in collections. To manipulate our mongoDB data base using python, we used PyMongo driver presented below.

PyMongo

PyMongo [14] is a Python distribution containing tools for working with MongoDB. This library allows Python scripts to connect and perform the CRUD (Create/Read/Update/Delete) operations on the mongoDB database.

Seaborn and Matplot

Seaborn [15] is a Python data visualization library based on MatPlotLib. It gives a high-level interface to draw attractive and informative statistical graphics,

MatPlotLib [16] is a 2D plotting library for the Python programming language and its numerical mathematics extension NumPy. It was inspired by MATLAB in the start. It provides an object-oriented API for embedding plots into applications using general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK+.

It can be used in:

- Python scripts,
- The Python and IPython shells,
- The Jupyter notebook,
- Web application servers,
- Four graphical user interface toolkits.

MatPlotLib allows us to generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., using just a few lines of code.

3.3 Topic Modeling

Algorithms

LDA

Latent Dirichlet Allocation [17] is a generative probabilistic model used for topic extraction in a given text collection. These topics are not strongly defined – as they are identified on the basis of the probability of co-occurrences of words contained in them.

As shown in Fig. 2, the boxes are "plates" that represent replicates:

The upper outer plate represents documents,

The upper inner plate denotes the repeated choice of topics and words within a document;

The lower plate marks the latent topics hidden in the document collection.

Formally, we define the following terms:

A word is the basic unit of discrete data, defined to be an item from a vocabulary indexed *by {1,..., V}*. These words are represented using unit-basis vectors that have a single component equal to one and all other components equal to zero. Consequently,

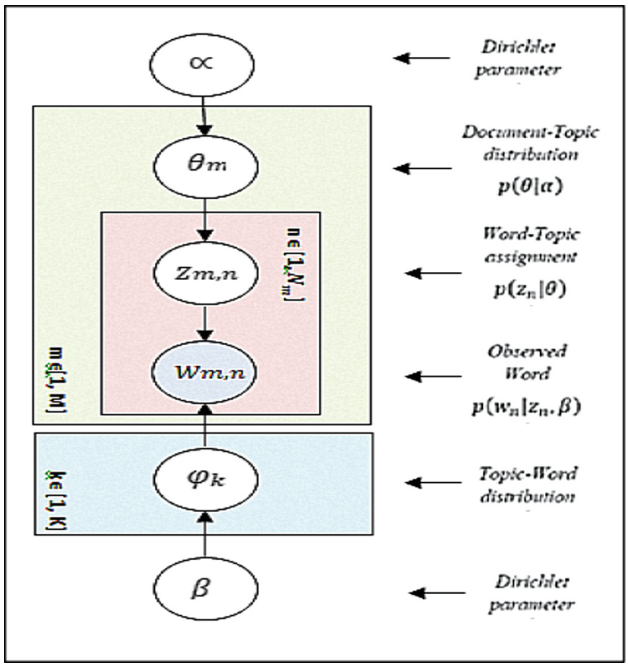

Fig. 2. Hierarchical graphical model for LDA.

using superscripts to denote components, the *vth* word in the vocabulary is represented by a V-vector w such that $w^v = 1$ and $w^u = 0$ for u ≠ v.

A sequence of N words is a document, denoted by $D = (w_1, w_2, \ldots, w_N)$, where w_n is the *n-th* word in the sequence.

A corpus is a collection of M documents di denoted by $C = (d_1, d_2, \ldots, d_M)$.

The generative process for each document in the text archive is illustrated as follows:

1. For the *m-th* (m = 1, 2, · · ·,M) document d in the whole M document-corpus, choose $\theta_m \sim$ Dirichlet(α);
2. For each word $w_{m,n}$ in the document d:

(a) Choose topic assignment $z_{m,n} \sim$ Multinomial (θ_m);
(b) Find the corresponding topic distribution $\varphi_{z_{m,n}} \sim$ Dirichlet(β);
(c) Sample a word $w_{m,n} \sim$ Multinomial ($\varphi_{z_{m,n}}$).

By repeating the procedures in the above generative process for M times with each corresponding to one document, which is plotted in graphical model language with "boxes" and "circles" as Fig. 1 (α and β are two Dirichlet-prior hyper-parameters). Obviously, we can easily arrive at the probability of the D corpus:

$$P(D|\alpha, \beta) = \prod_{m=1}^{M} \int P(\theta_m|\alpha)\, F(\theta, \varphi) d\theta_m \quad (1)$$

Where:

$$F(\theta,\ \varphi) = \prod_{n=1}^{N} \sum_{z_{m,n}} P\left(z_{m,n}|\theta_m\right) P\left(w_{m,n}|\varphi_{z_{m,n}}\right). \tag{2}$$

Nmf

Non-negative Matrix Factorization (NMF) [18], is a linear algebraic optimization algorithm, that factors the original high-dimensional data into a low-dimensionality representation with non-negative hidden structures, these data are viewed as coordinate axes in the transformed space with geometric perspectives.

Briefly, NMF tries to express the complex source matrix as a product of two matrices with much lower dimensionally.

Suppose we factorize a matrix X into two matrices W and H so that $X \approx WH$, NMF has an inherent clustering property, such that W and H represent the following information about X:

✓ X (Document-word matrix)—input that hold which words appear in which documents.
✓ W (Basis vectors)—the topics (clusters) discovered from the documents.
✓ H (Coefficient matrix)—the membership weights for the topics in each document.

We can calculate W and H by optimizing over an objective function (like the EM algorithm [19]), in parallel we update both W and H iteratively until convergence.

In the following objective function, we measure the error of reconstruction between X and the product of its factors W and H, based on Euclidean distance:

$$\frac{1}{2}\|X - WH\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{m} \left(X_{ij} - (WH)_{ij}\right)^2 \tag{3}$$

Using the objective function, we get the following values of W and H, which can be derived via the update rules:

$$W_{ic} \leftarrow W_{ic} \cdot \frac{(\mathbf{X}H)_{ic}}{(WHH)_{ic}} \qquad H_{cj} \leftarrow H_{cj} \cdot \frac{(W\mathbf{X})_{cj}}{(W\mathbf{W}\mathbf{H})_{cj}}$$

These updated values are calculated in parallel operations, and we re-calculate the reconstruction error, repeating this process until convergence, and using the new W and H.

Topic Coherence

UCI

UCI (or CV measure) [7] is automatic coherence measure to assess topics depending their understandability, this coherence measure handles words as facts and restricts to be always based on comparing word pairs.

UCI is derived from Pointwise Mutual Information (PMI) [20] that used to calculate words associations and word sense disambiguation, PMI measures how much one variable tells about the other and is formally defined as following:

$$PMI(w_i,\ w_j) = \log \frac{p\left(w_i\ ,\ w_j\right)}{p(w_i)p\left(w_j\right)} \tag{4}$$

Where the mutual information between words w_i and w_j compares the likelihood of observing the two words together to the likelihoods of observing them independently.

Then, the UCI measure is as follows:

$$Score_{UCI}(w_i, w_j) = \log \frac{p(w_i, w_j) + \varepsilon}{p(w_i)p(w_j)} \tag{5}$$

$p(w_i)$: The probability of word w_i appears in the corpus,

$p(w_j)$: The probability of word w_j appears in the corpus,

$p(w_i, w_j)$: The probability of the word w_i appears together with the word w_j on the corpus.

UMass Measure

UMass [11] calculates the correlation of words in a given document based in conditional probability.

The conditional probability of an event w_i given that event w_j has happened is:

$$P(w_i|w_j) = \frac{p(w_i \cap w_j)}{p(w_j)}; \; P(w_j) > 0 \tag{6}$$

Applying this concept, the equation of UMass measure is defined as follows:

$$Score_{UMass}(w_i, w_j) = \log \frac{D(w_i, w_j) + \varepsilon}{D(w_i)} \tag{7}$$

$D(w_i, w_j)$: The number of documents that contain words w_i and w_j.

$D(w_i)$: The number of documents containing w_i.

Being w_i always a word with more frequency than w_j.

4 Experiments and Empirical Results

In this section, we describe the collected data, and the different steps of pre-processing using NLP techniques. Finally, we present the results of our research work.

4.1 Data Collection and Storage

In this research, we collected 25 146 tweets published by Moroccan users from 26 April 2020 to 08 June 2020 and stored them in MongoDB database (Table 1).

Table 1. Description of the collected Dataset.

Start date	End date	Number of collected tweets
Apr 26, 2020	June 08, 2020	25 146 Tweets

We used Twitter API (provided by Twitter Platform) to pull data from Twitter. After creating an account on: https://apps.twitter.com, we had the authorization to access the database by using four secret keys (consumer key, consumer secret key, access token and access secret token) and collect tweets using the REST API.

To get Moroccan tweets written in one of the standard languages (Arabic, French and English), we filter tweets by place and language. To handle these data, we used Python library Tweepy, and to store the collected data in a MongoDB database we used Python library Pymongo.

4.2 Preprocessing Data

Our stored tweets were preprocessed using natural language processing techniques. We present, below, the most important procedures we applied:

1. Translation of Arabic and French texts to English using a python script based on google translate,
2. Conversion of all letters to lower case,
3. Removal of stopwords and punctuation using NLTK Python library [21] that provides a list of stopwords as well as punctuation symbols for many languages
4. Elimination of hyperlinks, hashes (tags were preserved) and usernames with preceding "@" which expresses a reply to another user
5. Removal of words with less than three characters
6. Tokenization which consists of broking sentences into sections (tokens). For instance, consider the sentence, before tokenization: Never give up!, and after tokenization it comes: {'Never', 'give', 'up','!'}
7. Stemming technique which consists of eliminating suffixes and affixes and getting the word base, For example: studying, study, student to study, Argue, arguing, argues to argue
8. Lemmatization of words (e.g., converting each word to its base form). We used Wordnet Lemmatizer, and we provide for each word its part of speech tag (POS TAG) (e.g., noun, verb etc.) using pos_tag method of NLTK library. Like studied to study, learned to learn etc.

4.3 Experimental Results

In this Section, we present the results of our analysis on Moroccan tweets; statistics, sentiment analysis and comparative study between NMF and LDA models.

Moroccan Tweets Statistics

We used seaborn [15] to visualize some statistics about collected Moroccan tweets concerning used languages (Fig. 3), places (Fig. 4) and the different types of accounts (Fig. 5).

Matplotlib enables many others libraries to run and plot on its base including Wordcloud, used to generate Fig. 6.

By analyzing collected Moroccan tweets, we found that the most used language is Arabic, followed by French. English language comes in third position as shown in Fig. 3.

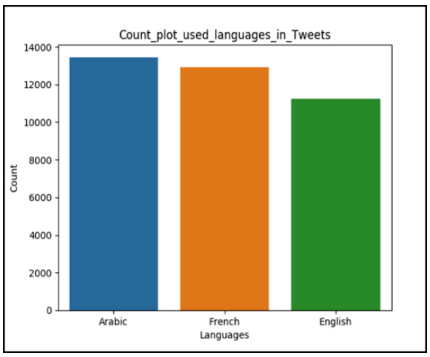

Fig. 3. Statistics about used languages active in collected tweets

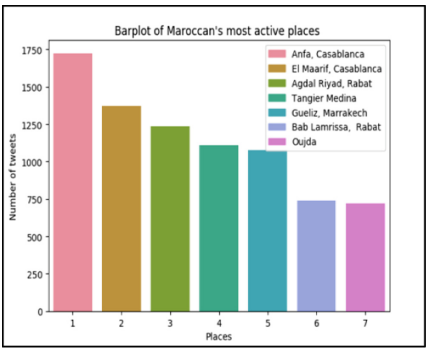

Fig. 4. Statistics about Moroccan's most places in terms of published tweets.

In Fig. 5, the bar graph presents Moroccan's most active places in terms of posted tweets. The first one is Anfa, and El Maarif (in Casablanca city), followed by Agdal (in Rabat city). Tangier, Gueliz (in Marrakech city), and Oujda city come last.

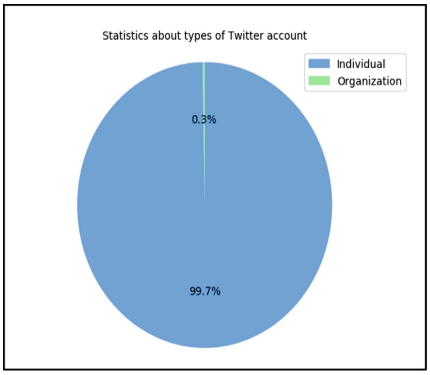

Fig. 5. Statistics about types of Twitter accounts.

Fig. 6. WordCloud of Moroccan #Hashtags.

Figure 5 shows that 99.7% of the accounts used in the analyzed tweets are individual accounts, whereas 0.3% of them are organization accounts.

Finally, Fig. 6 presents the Wordcloud of #Hashtags mentioned in collected tweets. It shows that the top cited words are: "Morocco", "Corona_Maroc", "Voyage", "Travel", "Repost", "Ramadan" and "COVID_19".

Sentiment Analysis

In order to determine the general emotion of each collected tweet, we used textblob library [22] which allows us to do sentiment analysis in a very simple way.

Textblob provides a trained model generated using Naïve Bayes algorithm. This model calculates a polarity score which is a float within the range [−1.0, 1.0], where

negative value indicates text expressing a negative sentiment and positive value indicates text expressing a positive sentiment.

We applied this model to our pre-processed tweets. The table below shows some examples of tweets and their calculated polarity (Table 2):

Table 2. Example of tweets and their calculated polarity.

Tweet	Polarity
رمضان مبارك كريم وكل عام وانتم بالف خير ان شاء الله	0.7
I've slept less than 2 hours in the past 24 hours and I'm feeling restless. I feel like I'm gonna end up crashing like crazy very soon... ☹	- 0.204
Et ça alors ? Donald Trump propose d'injecter les désinfectants aux malades de COVID-19. https://t.co/dnip7fEJXv	0.0

We created three Dataframes for the three categories: **positive** for the tweets with polarity > 0, **negative** for the tweets with polarity < 0, and **neutral** for tweets with polarity $= 0$. Then, we counted the number of tweets within each category and extracted the percentages as shown in Table 3. Finally, we classified them by places as shown in Fig. 7.

Table 3. Percentages of tweets in the three categories

# of tweets	% Positive	% Negative	% Neutral
25 146	35.85%	12.73%	51.41%

Topic Modeling

In order to discover the hidden topics that occur in our preprocessed tweets, we implemented two modeling algorithms; LDA and NMF. Firstly, we created a dictionary from the document collection using gensim package [23]. The dictionary created is a collection of unique terms in the document collection. This dictionary was then used to create a document-term matrix. This document-term matrix is used by each of the models.

For NMF, the same pre-processed corpus documents were transformed to log-based Term Frequency-Inverse Document Frequency (TF-IDF) vectors, after we used gensim to get the best number of topics with the coherence score and then use that number of topics for the sklearn [24] implementation of NMF. In the case of LDA, the MALLET [25] implementation was applied to the sets of document feature sequences.

Using the coherence score, we can run the model for different numbers of topics and then use the one with the highest coherence score.

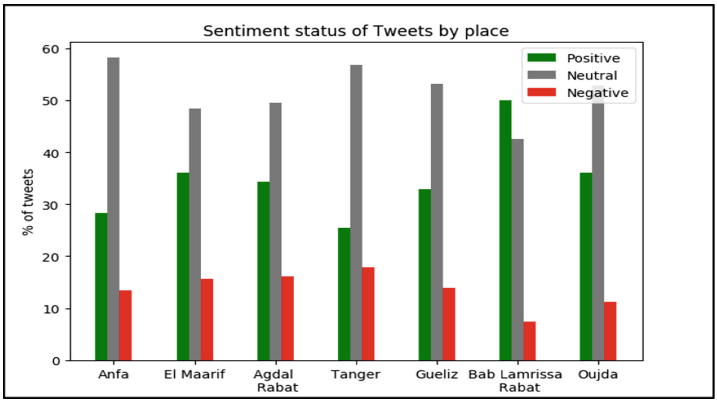

Fig. 7. .Sentiment analysis of tweets by places.

In two cases, topics were discovered for values of k ϵ [5; 70] (intervals of 5) as number of topics, and we calculated the coherence using CV (UCI) and UMass to compare NMF and LDA algorithms:

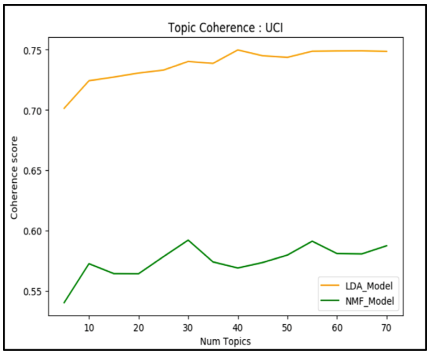

Fig. 8. Comparison in UCI score between LDA and NMF

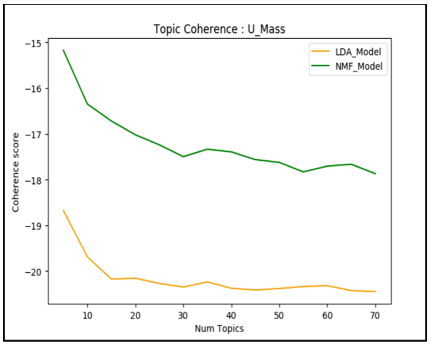

Fig. 9. Comparison in UMass score between LDA and NMF

We observed in Figs. 8 and 9, that LDA achieves higher topic-coherence scores across all of number of topics.

Finally, we used sklearn and gensim packages in Python to run NMF and LDA topic models, respectively.

As shown in Table 4 and 5, we selected the four most frequent topics detected by our models, each topic comprising of 10 words.

For example: Topic 01 in NMF model is about events in Morocco, topic 02 is about weather, topic 03 about feelings, and topic 04 about an event in USA concerning George Floyd death. Concerning LDA, topic 01 is about Corona virus, Topic 02 is about quarantine, topic 3 is about George Floyd death in USA, and Topic 04 is about holidays.

After inspecting the above topics, we noted that results generated by LDA are more meaningful than the ones modeled by NMF.

Table 4. Topics detected by applying NMF

Topic #01	Topic #02	Topic #03	Topic #04
sahara	wind_kmh	beautiful	trump
laayoune_western	clouds_humidiy	happy	update
true	humidity	really	think
story	temperature	good	forever
morocco	marrakech	nature	speedy
meaning	overcast	good_morne	strategy
people	rabat	shining	black
quarantine	cloud	love	racism
ramadan	Agadir	come	wrong
episode	humidity_wind	day	murder

Table 5. Topics detected by applying LDA

Topic #01	Topic #02	Topic #03	Topic #04
family	world	black_live	friend
virus	year	guy	work
end	still	police	thing
moment	quarantine	racism	nature
thought	home	attack	well
second	job	urgent	picture
together	covid_19	love	back
stay	feeling	trump	shining
corona	darkness	wrong	cool
chance	morocco	black	happy

Next to the quality, it is important to know the costs of topic modeling algorithms. Therefore, we have analyzed their runtimes; during the experiment, we used a dataset limited on English tweets and number of topics (k = 10) to analyze the runtimes of our models. As result, we observed that the time taken by LDA was 01 min and 30.33 s, while the one taken by NMF was 6.01 s, so NMF was faster than LDA.

5 Conclusion and Future Work

In this paper, we investigated the tweets posted by Moroccan users, calculating some useful statistics and visualizing the results using different graphs. We focused in particular, on the distribution of used languages, the most active places in terms of published tweets, the type of accounts (particular or organization), and the most frequently #Hashtags used in these tweets (WordCloud). Moreover, we performed a sentiment classification of tweets into three categories: positive, negative and neutral.

We also compared two of most popular topic models; LDA and NMF in order to dig out topics distribution under Moroccan tweets, which will lead to better understanding of

the Moroccan mood. In our experiment, we evaluated the models using topic coherence (UCI and UMass), meaning of topics and runtime of algorithms, and we deduced that LDA outperforms NMF if runtime is not a constraint.

In brief, the results shows that Twitter users in Morocco post more neutral tweets (51.41%), Casablanca is the most active place in terms of published tweets, and the most used language in tweets is Arabic. Moreover, the different discussed topics are diverse, but some topics are more prevalent than others, and the most remarkable topics concerned covid_19, George Floyd death and different events during studied period like Ramadan and holidays.

Future research efforts will be devoted to using Big Data plateforms (e.g., Hadoop, Spark) in order to store more tweets in a distributed way and accelerate their processing using different deep learning algorithms.

References

1. «Twitter Usage Statistics - Internet Live Stats». https://www.internetlivestats.com/twitter-statistics/. (consultéle févr. 19, 2020)
2. «DataReportal – Global Digital Insights». [En ligne]. Disponible sur: https://datareportal.com. [Consulté le: le mars 25, 2020]
3. Tripathi, P., Vishwakarma, S., Lala, A.: Sentiment analysis of English tweets using rapid miner. In : 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, pp. 668–672 (2015). https://doi.org/10.1109/CICN.2015.137
4. Al-Thubaity, A., Alqahtani, Q., Aljandal, A.: Sentiment lexicon for sentiment analysis of Saudi dialect tweets. Proc. Comput. Sci. **142**, 301–307 (2018). https://doi.org/10.1016/j.procs.2018.10.494
5. Wang, X., Yu, Y., Lin, L.: Tweeting the United Nations climate change conference in Paris (COP21): an analysis of a social network and factors determining the network influence. Online Soc. Netw. Med. **15**, 100059 (2020). https://doi.org/10.1016/j.osnem.2019.100059
6. Cvetojevic, S., Hochmair, H.H.: Analyzing the spread of tweets in response to Paris attacks. Comput. Environ. Urban Syst. **71**, 14–26 (2018). https://doi.org/10.1016/j.compenvurbsys.2018.03.010
7. Ray, S.K., Ahmad, A., Kumar, C.A.: Review and implementation of topic modeling in Hindi. Appl. Artif. Intell. **33**(11), 979–1007 (2019).https://doi.org/10.1080/08839514.2019.1661576
8. Greene, D., Cross, J.P.: Exploring the Political Agenda of the European Parliament Using a Dynamic Topic Modeling Approach. ArXiv160703055 Cs, juill. 2016, Consulté le: mai 30, 2020. [En ligne]. Disponible sur: https://arxiv.org/abs/1607.03055
9. Pasquali, A.R.: Automatic coherence evaluation applied to Topic Models (2016)
10. Ilyas, S.H.W., Soomro, Z.T., Anwar, A., Shahzad, H., Yaqub, U.: «Analyzing Brexit's impact using sentiment analysis and topic modeling on Twitter discussion», p. 7 (2020)
11. Dahal, B., Kumar, S.A.P., Li, Z.: Topic modeling and sentiment analysis of global climate change tweets. Soc. Netw. Anal. Min. **9**(1), 24 (2019). https://doi.org/10.1007/s13278-019-0568-8
12. «Tweepy». [En ligne]. Disponible sur: https://www.tweepy.org/. [Consulté le: 25-nov-2019]
13. «The most popular database for modern apps», MongoDB. [En ligne]. Disponible sur: https://www.mongodb.com. [Consulté le: 25-nov-2019]

14. Siddharth, S., Darsini, R., Sujithra, D.M.: Sentiment Analysis On Twitter Data Using Machine Learning Algorithms in Python, p. 15
15. «seaborn: statistical data visualization—seaborn 0.10.0 documentation». [En ligne]. Disponible sur: https://seaborn.pydata.org/. [Consulté le: 12-févr-2020]
16. «Matplotlib: Python plotting—Matplotlib 3.1.3 documentation». [En ligne]. Disponible sur: https://matplotlib.org/. [Consulté le: 12-févr-2020]
17. Blei, D.M. : Latent Dirichlet Allocation, p. 30
18. Chen, Y., et al.: Experimental explorations on short text topic mining between LDA and NMF based schemes. Knowl.-Based Syst. (2018). https://doi.org/10.1016/j.knosys.2018.08.011
19. Mclachlan, G.J., Krishnan, T.: The EM Algorithm and Extensions. John Wiley (2007)
20. Nugraha, P., Rifky Yusdiansyah, M., Murfi, H.: Fuzzy C-means in lower dimensional space for topics detection on Indonesian online news. In: Tan, Y., Shi, Y. (eds.) Data Mining and Big Data, vol. 1071, pp. 269–276. Springer, Singapore (2019)
21. «Natural Language Toolkit—NLTK 3.4.5 documentation». [En ligne]. Disponible sur: https://www.nltk.org/. [Consulté le: 17-févr-2020]
22. Loria, S.: textblob Documentation, pp. 1–73 (2018)
23. Rehurek, R.: «gensim: Python framework for fast Vector Space Modelling». [En ligne]. Disponible sur: https://pypi.org/project/gensim/. [Consulté le: 26-févr-2020]
24. «scikit-learn: machine learning in Python—scikit-learn 0.23.1 documentation». https://scikit-learn.org/stable/ (consulté le juin 06, 2020)
25. McCallum, A.: Mallet: A Machine Learning for Language Toolkit (2002)