Ranking news feed updates on social media: A comparative study of supervised models

Sami Belkacem*, Omar Boussaid**, Kamel Boukhalfa*

*LSI laboratory, USTHB, Algiers, Algeria

{s.belkacem,kboukhalfa}@usthb.dz

***ERIC laboratory, University of Lyon 2, Lyon, France

omar.boussaid@univ-lyon2.fr

Abstract. Social media users are overwhelmed by a large number of updates displayed chronologically in their news feed. Moreover, most updates are irrelevant. Ranking news feed updates by relevance has been proposed to help users catch up with the content they may find interesting. For this matter, supervised learning models have been commonly used to predict relevance. However, no comparative study was made to determine the most suitable models. In this work, we select, analyze, and compare six supervised learning algorithms applied to this case study. Experimental results on *Twitter* highlight that ensemble learning models are the most appropriate to predict the relevance of updates.

1 Introduction

Social media such as *Facebook*, *Twitter*, and *LinkedIn* are used by hundreds of millions of users worldwide. Due to the large number of members and the large amount of data posted and shared, users are overcome by a flow of updates displayed chronologically in their news feed (Ghazimatin et al., 2019). For example, a survey of 587 *Twitter* users showed that 66.3% of them cannot keep up with the large number of updates in their news feed (Bontcheva et al., 2013). Moreover, most of those updates are considered irrelevant. For example, the survey of 587 *Twitter* users revealed that 70.4% of them have trouble finding the relevant updates in their news feed (Bontcheva et al., 2013). Therefore, large data volume and irrelevance make it difficult for users to catch up with the relevant updates in their news feed (Piao and Breslin, 2018).

In several research approaches, ranking news feed updates in descending relevance order has been proposed to help users quickly catch up with the content they may find interesting (Vougioukas et al., 2017). For this matter, supervised learning models have been commonly used and seem suitable to rank news feed updates (Belkacem et al., 2019). Indeed, using labeled training data, these models analyze users' past behaviors to predict whether they will find an update relevant or not in the future (Sammut and Webb, 2011). However, each research work intuitively chooses one supervised learning algorithm, states that other algorithms can be used, and points out that it is out of the scope to compare them (Belkacem et al., 2016). In this work, knowing that the effectiveness of the ranking process depends partly on the chosen model, we conduct a comparative study to determine the most suitable models by selecting, analyzing, and comparing six supervised learning algorithms applied to this case study.

This work focuses on *Twitter* for the followings reasons: (1) the large flow of tweets; (2) the irrelevance of a large part of tweets; (3) the fact that social data are public unlike most other social media; and (4) the availability of API for easy crawling (Berkovsky and Freyne, 2015). However, it would be possible to adapt this work to other social platforms. A tweet has (see Fig. 1): (1) an author; (2) a set of beneficiary users who can read and interact with it; (3) a textual and/or multimedia content; (4) a publication date; (5) mentions which represent links to other users; (6) hashtags which identify tweets on specific topics; and (7) URLs to websites. Users who follow a user ***u*** are called *followers* of ***u*** and users that ***u*** follows are called *followings* of ***u***. If ***u*** follows another user ***u'***, ***u*** will receive in the news feed the tweets of ***u'***.

The paper is structured as follows: section 2 provides a background on ranking news feed updates on *Twitter*, section 3 presents and discusses the experiments we performed to evaluate and compare the supervised models, and section 4 concludes and proposes future work.

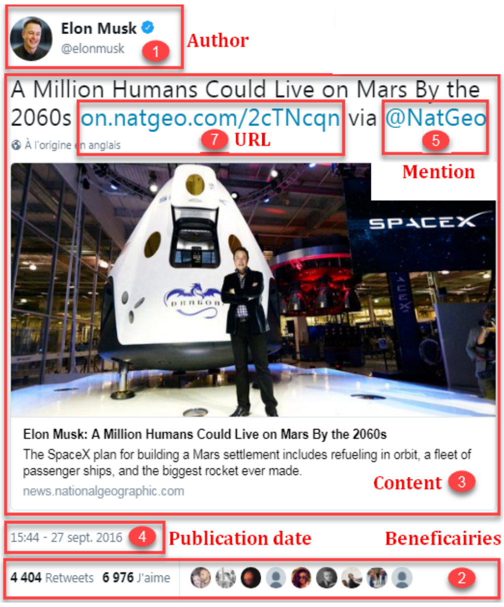
2 Ranking news feed updates on *Twitter*

A user's news feed on *Twitter* is a list of tweets where are displayed from most recent to least recent tweets posted by his followings. Berkovsky and Freyne (2015) propose the following formalization of the problem of ranking news feed updates: "Let ***F(u)*** denotes tweets unread by the beneficiary user ***u*** that can be included in the news feed. Ranking implies selecting and displaying a subset ***K(u)*** $\in$ ***F(u)***, such that $|$***K(u)***$| \ll |$***F(u)***$|$, that corresponds to the most relevant tweets to ***u***". The rest of this paper focuses on the most important step of the ranking, which is predicting a relevance score to each tweet ***t*** $\in$ ***F(u)***. Note that other terms are used to refer to the ranking process, e.g., reordering, recommendation, personalization, etc.

Fig. 2 describes the primary technique used to predict the relevance score ***R(t,u)*** of a tweet ***t*** $\in$ ***F(u)***. This technique is based on a supervised prediction model that analyzes labeled training data of tweets that ***u*** read in the past to predict if he will find ***t*** relevant in the future. Let ***D(u)*** denotes a subset of tweets previously read by ***u***. The training data is a set of input-out pairs such that an input represents the features that may influence the relevance of a tweet ***t'*** $\in$ ***D(u)*** to ***u***, and the output represents the relevance score ***R(t',u)***. The primary technique involves three steps: (1) assign implicit relevance scores to tweets; (2) extract the features that may influence relevance; and (3) train the relevance prediction model. In the rest of this section, we describe each of the steps according to a typical approach (Belkacem et al., 2019).

2.1 Relevance scores

We use the implicit method which has been used by most related work (Belkacem et al., 2017). It assumes that a previously read tweet ***t'*** $\in$ ***D(u)*** is relevant to a user ***u*** $\in$ ***S*** if ***u*** inter-



Elon Musk
@elonmusk

A Million Humans Could Live on Mars By the 2060s on.natgeo.com/2cTNcqn via @NatGeo

À l'origine en anglais

Elon Musk: A Million Humans Could Live on Mars By the 2060s

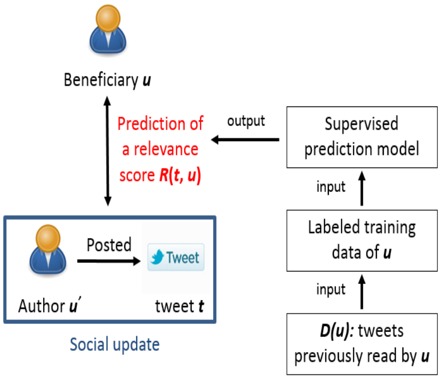
The SpaceX plan for building a Mars settlement includes refueling in orbit, a fleet of passenger ships, and the biggest rocket ever made.

news.nationalgeographic.com

15:44 - 27 sept. 2016

4 404 Retweets 6 976 J'aime

FIG. 1: Tweet posted by *Elon Musk*.



Beneficiary *u*
Prediction of a relevance score *R(t, u)*
output
Supervised prediction model
input
Labeled training data of *u*
input
D(u): tweets previously read by *u*
Posted
Tweet
Author *u′*
tweet *t*
Social update

FIG. 2: Prediction of a relevance score.

acted with ***t'***. Predicting implicit relevance scores results in a binary classification problem:

$$\boldsymbol{R(t',u)} = \begin{cases} 1 & \text{if } \boldsymbol{u} \text{ interacted with } \boldsymbol{t'} \text{ (retweet or reply or like)} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

We use the implicit method because the explicit method used by Kuang et al. (2016) has limitations. It is not related to users' interactions (users' feedbacks were obtained via a survey), and it is binding as it asks users to assign relevance scores to a large number of tweets. We also split relevance scores into two bins because train a finer-grained classifier (e.g., ***t*** is very relevant to ***u*** if he retweeted, liked, and replied to it) would be difficult since users' multiple interactions with the same tweet are not frequent. Out of 569 tweets, we found that only 5% and 0% of tweets get respectively two and three types of interaction from the same user.

2.2 Features that may influence relevance

We use 16 related work features that may influence the relevance ***R(t,u)*** of a tweet ***t***, posted by an author ***u'***, to the beneficiary ***u***. More details are given in (Belkacem et al., 2019). The features are gradually updated, listed in Table 1, and divided into five categories (see Fig. 2):

— Features between ***u*** and ***t*** that match between the content of ***t*** and the interests of ***u***.

— Features between ***u*** and ***u'*** that measure social tie strength between them. The assumption is that ***t*** could be relevant to ***u*** if he has a strong social relationship with ***u'***.

— Features between ***u'*** and ***t*** that measure the expertise of ***u'*** in the topics of ***t***. The assumption is that ***t*** could be relevant to ***u*** if ***u'*** is an expert in the topics of ***t***.

TAB. 1: Features that may influence relevance

Features that may influence relevance		**Type**	**N°**
Relevance of the content of ***t***, its hashtags, and mentions to ***u***	Relevance of the keywords of ***t*** to ***u***	Int	*f1*
	Relevance of the hashtags of ***t*** to ***u***	Int	*f2*
	Presence of ***u*** in the mentions of ***t***	Bool	*f3*
Social tie strength between ***u*** and ***u'***	Interaction rate of ***u*** with tweets of ***u'***	Float	*f4*
	Number of times ***u*** mentioned ***u'***	Int	*f5*
Expertise of ***u'*** in the topics of ***t***	Publishing rate of ***u'*** for keywords of ***t***	Float	*f6*
	Interaction rate with keywords of ***t*** posted by ***u'***	Float	*f7*
	Keywords of ***u'*** biography and keywords of ***t***	Bool	*f8*
Authority of ***u'***	Followers count / Followings count	Int	*f9*
	Seniority in years	Int	*f10*
	Listed (group) count	Int	*f11*
Quality of ***t***	Length (# characters)	Int	*f12*
	Presence of hashtags	Bool	*f13*
	Presence of a URL	Bool	*f14*
	Presence of an image or a video	Bool	*f15*
	Popularity (# retweets, replies, likes)	Int	*f16*

- Features of ***u'*** that measure his authority. The assumption is that ***t*** could be relevant to ***u*** if ***u'*** has authority. Indeed, if a user is important, then his tweets are also important.
- Features of ***t*** that measure its quality: length, popularity, the presence of a multimedia content, etc. The assumption is that ***t*** could be relevant to ***u*** if it is of high quality.

2.3 Relevance prediction model

Let ***S*** denotes the set of beneficiary users. First, we generate training data instances for each user ***u*** $\in$ ***S*** in the form of input-output pairs considering each previously read tweet ***t'*** $\in$ ***D(u)***. An input represents the features that may influence the relevance of ***t'*** to ***u***, and the output represents the implicit relevance score ***R(t',u)***. Second, we divide the training data of each user ***u*** $\in$ ***S*** into two sets: a training set of the prediction model for 70% of the first instances (the least recent ones) and a test set for 30% of the remaining instances (the most recent ones).

Finally, we use the training set of each user ***u*** $\in$ ***S*** to train a supervised prediction model. Using a binary classifier learned from previously read tweets in the training set, the purpose is to map new input features of a tweet unread by ***u*** to a relevance score. In the next section, we describe the experiments we performed to evaluate and compare six supervised models.

3 Experiments and comparison results

To evaluate and compare the supervised models, we describe in this section: (1) the dataset used in the experiments we performed; (2) the measures used to evaluate the performance; and (3) the methodology we use in the comparison as well as the obtained results.

3.1 Dataset

First, we randomly selected a set ***S*** of 46 beneficiary users. Then, we collected data over 10 months using *Twitter Rest API*[1]. To simulate the news feed of each user ***u*** $\in$ ***S***, we used the principle proposed by Feng and Wang (2013) to select, ***D(u)***, the subset of tweets posted by the followings of ***u*** that he may have read. The variant is as follows: (1) sort all the tweets posted by the followings of ***u*** from least recent to most recent; (2) for each tweet ***t'*** with which ***u*** interacted, keep the chronological session defined by the tweet ***t'***, the tweet before ***t'*** and the tweet after ***t'***. This resulted in an interaction rate with tweets of approximately 35% and an average number of instances of 569 tweets in the training data of each beneficiary user.

3.2 Measures

First, we train a binary classifier model for each user ***u*** $\in$ ***S*** using the corresponding training set (70% of the least recent instances). Then, we define the following concepts to evaluate the model using the corresponding test set (30% of the most recent instances):

— TP (True Positive): number of relevant tweets correctly predicted relevant to ***u***
— TN (True Negative): number of irrelevant tweets correctly predicted irrelevant to ***u***
— FP (False Positive): number of irrelevant tweets incorrectly predicted relevant to ***u***
— FN (False Negative): number of relevant tweets incorrectly predicted irrelevant to ***u***

After that, we use the weighted *F1 score* given by Equation 2 (Sammut and Webb, 2011). This measure is suitable to evaluate the performance since classes are slightly unbalanced with an interaction rate with tweets of approximately 35% for each beneficiary user.

$$F = \frac{(F_r \times (TP + FN)) + (F_i \times (TN + FP))}{TP + TN + FP + FN} \tag{2}$$

Where:
— F_r is the standard *F1 score* for the class of relevant tweets
— F_i is the standard *F1 score* for the class of irrelevant tweets

[1] https://dev.twitter.com/rest/public

3.3 Results

First, we selected six supervised algorithms that were used in several related works (Belkacem et al., 2019): Naive Bayes (NB), Logistic Regression (LR), Decision Trees (DT), Gradient Boosting (GB), Random Forest (RF), and Support Vector Machine (SVM). Further details about the algorithms are available in (Sammut and Webb, 2011). Then, since SVM requires data scaling, we normalized all feature values in the range [0,1] using the Min-max scale. Moreover, for a fair comparison, we selected the best parameters of each algorithm with a 5-fold time-series cross-validation performed on the train set. We thus ran *Randomized Search* over different parameter values, which is a widely used strategy for algorithm hyper-parameter optimization. Finally, to study the algorithmic stability with small changes to training data, we retrained and iterated each model over 60 random state [2] values, then evaluated it on the test set. We end up with 60 *F score* values for each algorithm and plot the corresponding boxplot.

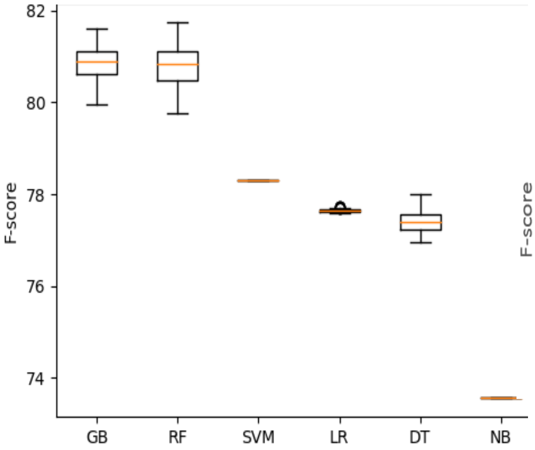


FIG. 3: *F scores* over 60 random state values.

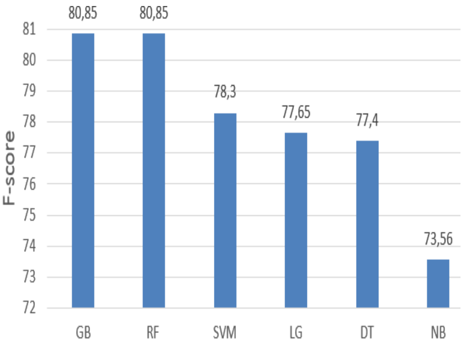


FIG. 4: Average *F scores*.

The experimental results were obtained using the *Python* environment and the *scikit-learn* library [3]. Fig. 3 presents the boxplot of the comparison between the algorithms trained and evaluated over 60 random state values, and Fig. 4 shows the comparison between the average *F scores*. First, both figures show that the scores differ greatly depending on the algorithm, from 80.85% on average with GB and RF to 73.56% with NB. This confirms that comparing and choosing the most suitable supervised model is critical to ensure the effectiveness of the ranking process. Moreover, we point out from Fig. 3 that SVM, LR, and NB remain stable even if retrained and iterated over different random state values. Indeed, the learning process of these models does not imply random sampling from data, unlike GB, RF, and DT.

However, although GB and RF are less stable, we notice from Fig. 3 and Fig. 4 that they outperform the other algorithms, with an average *F score* of 80.85%. This highlights that ensemble learning models such as GB and RF, which combine multiple learning algorithms to

[2] Variable used in randomized algorithms to determine the random seed of the pseudo-random number generator
[3] https://scikit-learn.org/stable/

improve the overall performance, are the most appropriate to predict the relevance of updates. We point out that both GB and RF combine multiple decision trees. The results also reveal that SVM performs well with an average *F score* of 78.3% but not as well as GB and RF. Indeed, as in the case study of ranking news feed updates, it has been proved that ensemble methods such as GB and RF tend to perform better than SVM in clearly defined problems with small to intermediate datasets and a manageable number of features (Sammut and Webb, 2011).

The results also indicate that LR and DT perform moderately with average *F scores* of 77.65% and 77.4%, respectively. Being parametric algorithms makes them simpler, faster to train, require less data but not as powerful as GB, RF, and SVM, which are nonparametric algorithms. Nonparametric methods are in fact more flexible as they can learn any functional form from the training data, but have a higher model complexity and require more data and training time. Finally, we observe that NB performs poorly with an average score of 73.56%. This is probably due to the strong assumption this parametric algorithm makes about the independence of the features which do not hold in the case study of ranking news feed updates. Indeed, the value of some features is dependent on the value of other features, e.g., the more the followers a user has on *Twitter* (feature f_9), the more he is listed in groups (feature f_{11}).

Despite the effectiveness and the excellent performance of GB and RF, these algorithms remain shallow learning models whose performance depend largely on the manually extracted features. However, to the best of our knowledge, deep neural network models that automatically execute feature extraction have not yet been used in ranking news feed updates. Therefore, for any further improvement, it would be interesting to consider the feasibility of deep learning models and compare their performance with ensemble methods such as GB and RF.

4 Conclusion

In this work, we first presented a background on ranking news feed updates on *Twitter*. Then, we selected, analyzed, and compared six supervised models applied to this case study. To the best of our knowledge, this paper is the first to make such a comparative study. Following experiments on *Twitter* with a rigorous methodology, the comparison results highlight that choosing the most suitable supervised model is critical to ensure the effectiveness of the ranking process. Moreover, the results show that ensemble learning models such as Gradient Boosting and Random Forest are the most appropriate to predict the relevance of updates.

For now, we only evaluated and compared the supervised models implicitly using users' interactions. For further work, we intend to get explicit users' feedback by asking their opinion on the predicted relevance scores. We also plan to make a scalable and detailed analysis and include in the comparison, deep learning models that automatically execute feature extraction.

References

Belkacem, S., K. Boukhalfa, and O. Boussaid (2016). News feeds triage on social networks: A survey. In *Proceedings of The 2nd International Conference on Computing Systems and*

Applications (CSA), pp. 34–43.

Belkacem, S., K. Boukhalfa, and O. Boussaid (2017). Tri des actualités sociales: Etat de l'art et Pistes de recherche. In *Journées francophones sur les Entrepôts de Données et l'Analyse en ligne (EDA)*, Volume 13, pp. 85–100. Revue des Nouvelles Technologies de l'Information.

Belkacem, S., K. Boukhalfa, and O. Boussaid (2019). Expertise-aware news feed updates recommendation: a random forest approach. *Cluster Computing*.

Berkovsky, S. and J. Freyne (2015). Personalised Network Activity Feeds: Finding Needles in the Haystacks. In *Mining, Modeling, and Recommending'Things' in Social Media*, pp. 21–34. Springer.

Bontcheva, K., G. Gorrell, and B. Wessels (2013). Social media and information overload: Survey results. *arXiv preprint arXiv:1306.0813*.

Feng, W. and J. Wang (2013). Retweet or not?: personalized tweet re-ranking. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pp. 577–586. ACM.

Ghazimatin, A., R. Saha Roy, and G. Weikum (2019). FAIRY: A Framework for Understanding Relationships Between Users' Actions and their Social Feeds. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pp. 240–248. ACM.

Kuang, L., X. Tang, M. Yu, Y. Huang, and K. Guo (2016). A comprehensive ranking model for tweets big data in online social network. *EURASIP Journal on Wireless Communications and Networking 2016*(1), 1.

Piao, G. and J. G. Breslin (2018). Learning to Rank Tweets with Author-Based Long Short-Term Memory Networks. In *International Conference on Web Engineering*, pp. 288–295. Springer.

Sammut, C. and G. I. Webb (2011). *Encyclopedia of machine learning*. Springer Science & Business Media.

Vougioukas, M., I. Androutsopoulos, and G. Paliouras (2017). Identifying Retweetable Tweets with a Personalized Global Classifier. *CoRR abs/1709.06518*.

Résumé

Les utilisateurs de médias sociaux sont submergés par un grand nombre d'actualités affichées chronologiquement dans leur fil d'actualités. De plus, la plupart des actualités sont non pertinentes. Le tri des fils d'actualité par ordre de pertinence a été proposé pour aider les utilisateurs à rattraper le contenu qui pourrait les intéresser. Pour ce faire, les modèles d'apprentissage supervisé ont été couramment utilisés pour prédire la pertinence. Toutefois, aucune étude comparative n'a été effectuée pour déterminer les modèles les plus appropriés. Dans ce travail, nous sélectionnons, analysons et comparons plusieurs algorithmes d'apprentissage supervisé appliqués à ce cas d'étude. Les résultats expérimentaux sur *Twitter* soulignent que les modèles ensemblistes d'apprentissage sont les plus adaptés pour prédire la pertinence des actualités.