Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 159 (2019) 352–362

Procedia Computer Science

www.elsevier.com/locate/procedia

23rd International Conference on Knowledge-Based and Intelligent Information & Engineering Systems

A Deep Learning-based Ranking Approach for Microblog Retrieval

Ben Ltaifa Ibtihel[a,*], Hlaoua Lobna[b], Ben Romdhane Lotfi[b]

[a]*ReDCAD Laboratory, ENIS Sokra km 3.5, University of Sfax, Tunisia*
[b]*MARS Research Laboratory, LR 17ES05, University of Sousse, Tunisia*

Abstract

Today, Twitter has become one of the most popular micro-blogging service with a large amount of information on various topics produced by millions of users every day. When searching for useful information in Twitter, users need to assess high quality content that meets their needs. However, the study of effective information retrieval in such microblog is still a challenge because there is a large difference in the quality level of relevant tweets returned in the search results for a given query. Therefore, looking for an effective microblog retrieval requires distinguishing the high-quality tweet content among thousands of results. Existing works are based on hand-crafted features (e.g., number of re-tweets, number of followers, etc.) using hard-hand-engineering to indicate the quality of tweets. In this paper, we focus on the problem of ranking tweets, and particularly on retrieving high quality content. We propose a ranking approach based on k-means clustering to distinguish high quality from low quality tweets. The clustering algorithm is based on learning features from deep learning autoencoder and hand-crafted features from tweets' content and authors' profiles. We used information gain as a feature importance measure to find the optimal set of features having stronger power in clustering the data. By conducting a pilot feature analysis study, we demonstrate the impact of the learned features to identify tweets' quality in the clustering process. Our experimental results show that the integration of learned features has shown significant improvement in the quality of clustering and especially on the ranking performance compared to the use of hand-crafted features only.

Keywords: information retrieval; ranking microblogs; k-means clustering; deep learning; information gain

1. Introduction and Motivation

In recent years, Twitter, as a major microblog service provider, has been an active space of research in many tasks: microblog retrieval, data mining (e.g., sentiment analysis, event detection, etc.), tweet classification and searching for trending topics. Given the increasing number of users, with 336 million monthly active users, according to the Twitter statistics up to 2019, micro-blog content volume also showed an explosive growth trend. Obviously, facing such a

* Corresponding author. Tel.: +216 53 632 258 ; fax: +216 74 666 578
E-mail address: ibtihel.beltaifa@gmail.com

10.1016/j.procs.2019.09.190

large-sized content, it would be interesting to be able to effectively retrieve useful information according to users' needs. However, the work of information retrieval in Twitter is still a challenge. This is mainly in ad-hoc microblog retrieval aiming to find the most relevant and recent tweets in response to queries issued by users [20]. In a similar scenario, TREC (Text REtrieval Conference) microblog introduced an ad-hoc information retrieval track which began in 2011. Several studies have investigated this task. Yet, only a few presented effective retrieval results. For the TREC 2011/2012 microblog track, which goal was to find the most relevant and recent tweets at a specific time in response to a given information need [18], tweets are ranked according to the post time from the most recent to the oldest. However, this proposed ranking method, based on reverse chronological order, provides no guarantee that the most relevant tweets appear on top list. Moreover, retrieved tweets in the search results based on keyword matching may not satisfy the user's information need or real intention. Many of these tweets are not informative. They may include a low quality content. Then, among thousands of results, looking for an effective microblog retrieval requires differentiating the high-quality tweets' content from the other contents.

In this context, many different approaches have been proposed to improve microblog retrieval. They can be summarized as using query expansion techniques to expand the query by adding the terms close to the original query terms [28] [13], performing new ranking strategies and approaches [15] [14] or the combination of both [11]. In the literature, the proposed ranking approaches of microblogs are based on various features and many of them deployed learning to rank algorithms for the ranking process using machine learning algorithms such as support vector machine and random forest [3] [26]. For these ranking approaches, the following issues were addressed: (1) All ranking approaches are based on traditional machine learning algorithms that depend heavily on the features' representation form, yet, it is difficult to choose what features should be extracted. (2) All ranking approaches are based on hand-crafted features, also called hand-designed features, where the designer manually chooses features and their representations or labels and puts them into the algorithm requiring a lot of time and laborious efforts. Specifically, most of these algorithms require that we manually specify the input features using hard-hand-engineering (feature engineering). Once a good feature representation is given, a learning algorithm can do well. (3) Most ranking approaches are only based on feature engineering which utilizes specific features of the microblogs (e.g., the number of followers, number of hashtags, etc.), without deeply studying the ranking techniques or methods. Many features are not used like the description of the author which indicates the author's center of interest or area of expertise. (4) All ranking approaches are based on features that do not reflect the high quality of tweet content: all of them are based on non-text features and do not take in consideration the text of the tweet in terms of well-formedness (correct spelling and grammar) or the description of the author in his/her profile. Moreover, other works have studied the quality of microblog posts by finding features that indicate the quality of tweets. But the impact of these quality-based features on microblog retrieval task is not well-discussed and clear. Then, the study of effective information retrieval in such microblog is still a challenge because there is a large difference in the level of quality of relevant tweets returned in the search results for a given query. However, finding informative tweets needs to consider the tweet content specificities, such as poor syntax, vocabulary mismatch and abbreviations, which makes of retrieving high quality tweets hard.

Recently, various deep-learning-based methods have been applied in many tasks, espacially in classification tasks of images or text. The deep learning as a subset of machine learning utilizes artificial neural networks that imitate the work of the human brain in extracting and processing data's features [5]. Deep Learning algorithms such as convolutional deep neural networks, recurrent neural networks and stacked autoencoders, are based on neural networks with multilayers that enable learning from large amounts of data [16]. These representation learning algorithms inspired by the human brain allow machines to learn from data sets that can be diverse, unstructured or unlabeled without human intervention. Autoencoder is a type of neural network that applies back propagation to reconstruct its input data. It is a form of feature extraction algorithm that automatically learns features from unlabeled data by forcing the hidden layer (encoding) to compress the data into a low-dimensional representation in order to learn good representations of the inputs [1].

In this paper, we study the problem of ranking tweets from different directions. First, we try to classify tweets into two classes, high-quality and low quality using the k-means clustering algorithm. For the clustering process, we used learning features from deep learning autoencoder with the combination of hand-crafted features based on the tweet's content and the author's profile and behavior. Secondly, we use information gain as a feature importance measure to find the optimal set of features having stronger power in clustering the data. And, by conducting a pilot feature analysis study, we determine the impact of the learned features to identify tweets' quality. Finally, we perform experiments to rank tweets using results from clustering. Also, we demonstrate that the use of deep learning algorithm such as the autoencoder neural network, can automatically learn better and has high-level feature representations than the hand-engineered ones. And, the integration of these learned features can improve the quality of ranking compared to the use of hand-crafted features only. For the same purpose, we demonstrate that our ranking strategy increases the performance of the retrieval effectiveness compared to the reverse-chronological order ranking.

The rest of the paper is organized as follows: in Section 2 we present a review of related work in the area of ranking microblogs. Section 3 presents our ranking approach based k-means clustering. Section 4 provides features' analysis study to determine the impact of the learned features from deep learning autoencoder to identify tweets' quality and an evaluation of the ranking approach, as well as of our k-means clustering. Finally, conclusion and future work plans are given in section 5.

2. Related work

Various ranking approaches of microblogs have been proposed to reorder the results provided by an information retrieval model to find high-quality content. These approaches include the use of the content-based features (e.g.,URLs, number of hashtags, number of retweets) of the microblogs, as well as the social-based features of the bloggers (e.g, the author's authority, the author's popularity). Authors in [3] proposed a new ranking strategy which uses not only the content relevance of a tweet, but also the account authority and the combination of some properties of the tweets, such as the length of the tweet and whether the tweet contains URLs. Authors in [15] proposed a method of ranking tweets considering trustworthiness and content-based popularity. The analysis of trustworthiness and popularity exploits the implicit relationships between the tweets. In [12] the authors considered two metrics to rank tweets: TweetRank of an author based on the number of tweets posted and FollowerRank of an author based on the high number of followers in his social network. Authors in [2] proposed a quality model to estimate the tweet's quality measure using surrogate judgments based on the retweet-set and the set of quality features. However, authors in [9] evaluated the quality of tweet by exploiting the relationships between tweets, authors, and readers. The quality of the tweets is measured by identifying the influence level of their authors using different factors such as the re-tweeting behavior and the topic-specific influence level of users who re-tweet the tweet.

In addition, different hybrid approaches that explore content relevance and tweet-specific information such as temporality, have also been considered to rank tweets [27]. Some studies employ both lexical similarity and temporality (the publishing times of tweets and the timestamps of queries) in ranking tweets [4] [7]. Authors in [14] proposed to re-rank microblogs obtained by Indri[1] search engine using Word2Vec model. In [26], authors proposed a ranking model based on query-sensitive selection and fusion of multiple-learned-ranking model. They showed that the use of different fusion methods by leveraging ranked lists from top rankers for each query, can improve the ranking effectiveness over the single ranker. In [25] a microblog-ranking approach called YouRank is proposed. It is based on the leverage of the user engagement activities (re-tweet/replies engagement) with tweets related to each author.

Other works focused on both filtering and ranking steps to find the most interesting microblogs with high-quality content. Authors in [23] modeled the quality of a tweet based on content-based features and link-based features which include the reputation of the URL domain. In [22], authors proposed a user-oriented tweet ranking method based on the use of the retweet behavior. Other approches are based on incorporating the author's authority in the ranking process to filter the quality of tweets [27]. This method is based on two topical and conventional features : the author's topical follower signal and the author's conventional popularity signal (e.g.,number of followers, number of friends, etc.).

[1] https://www.lemurproject.org/indri/

3. Our proposed ranking approach based on k-means clustering

In this section, we present our ranking approach based on k-means clustering to distinguish high quality and low quality tweets to improve the performance of the retrieval effectiveness. For the clustering process, we used learning features from deep learning autoencoder with the combination of hand-crafted features based on the tweet's content and the author's profile and behavior. The overall process of our ranking approach is highlighted in the following steps, as illustrated in Figure 1.

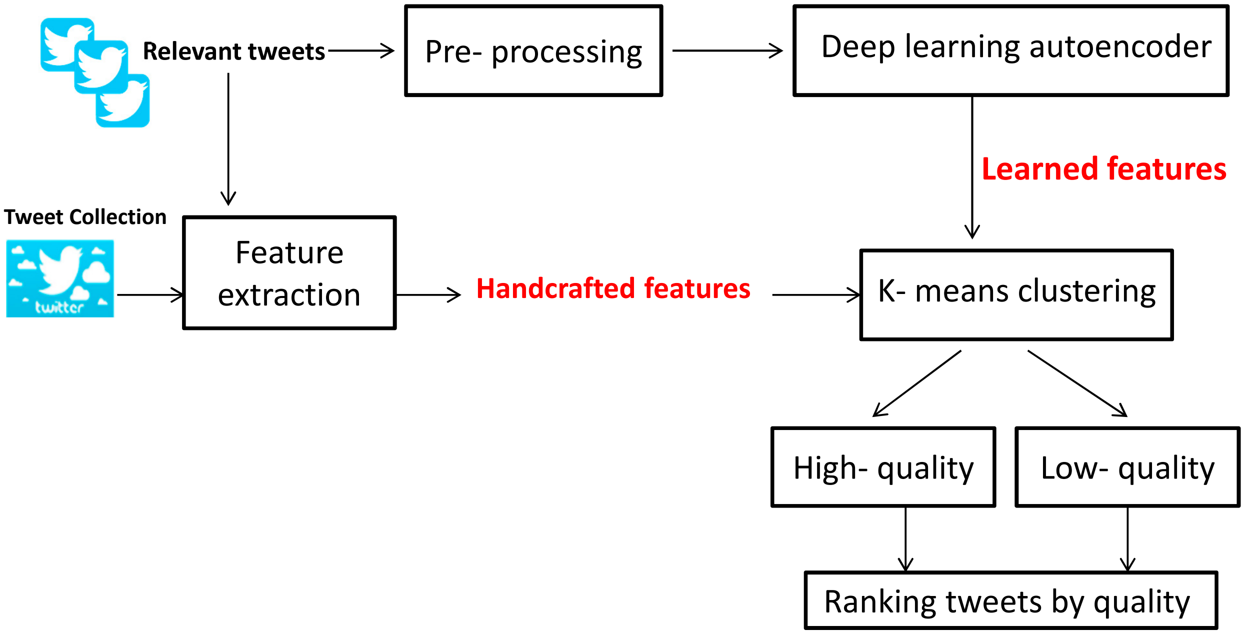

Fig. 1. Overall process of ranking tweets

3.1. Pre-processing

Given a set of data, we begin with the pre-processing step to clean the input data (removing punctuation, stop words, numerical characters, symbols and special characters ('RT','@','#', emoticons)) and perform word embeddings with neural network. Word embedding is a representation method to capture the context of a word in the text (text in the tweet or in the description of the author) and represent each word in the text as vector.

3.2. Learning features from deep learning autoencoder

In order to learn the most important and salient features present in the input data (word embeddings), we used the autoencoder neural network. An autoencoder neural network architecture is based on three main components to learn efficient representations: encoder, code and decoder [10], as illustrated in Figure 2.

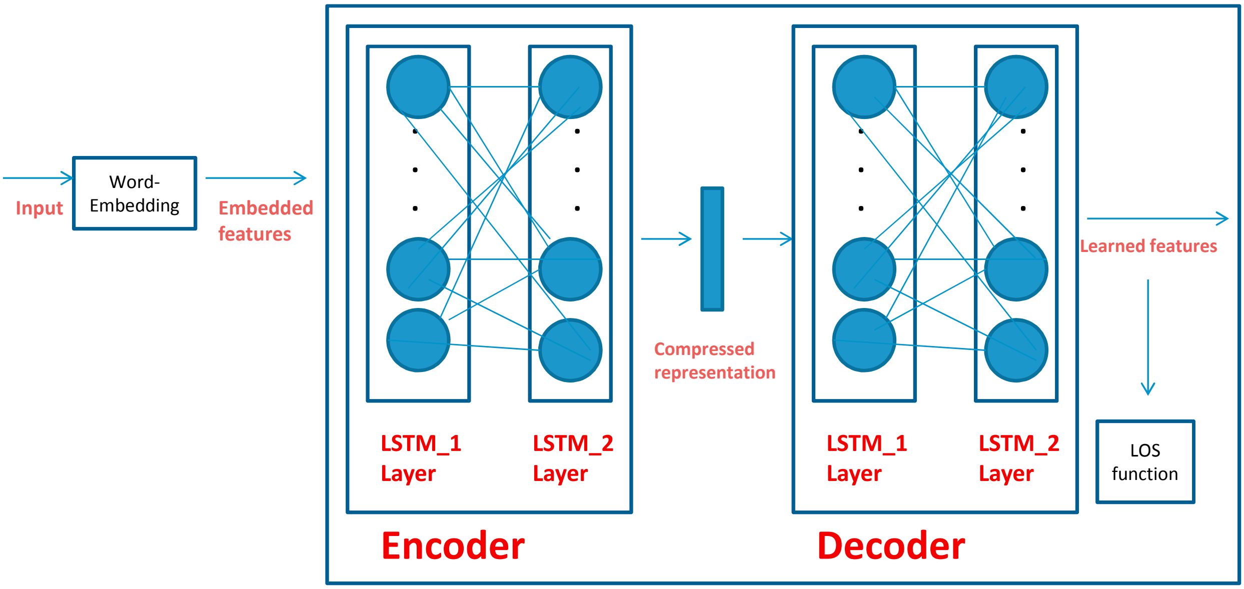

Fig. 2. Autoencoder neural network architecture

The encoder compresses the input vectors (embedding words) into a lower-dimensional and produces the code (called the latent-space representation or compressed representation). The decoder then reconstructs the output from this representation (The vector representation) using only this code that can be deployed by the decoder to reproduce the original input. In our work, we used LSTM (Long Short-Term Memory Networks) autoencoder for automatic features extraction from text tweet based on Encoder-Decoder LSTM architecture where both encoder and decoder layers are LSTMs. Finaly, the Autoencoder used the loss function as optimization function which learns to reduce the error in prediction. In the context of a regression problem (regression losses) dealing with predicting continuous values, we used Mean Squared Error (MSE) as loss function. MSE is measured as the average of squared difference between predictions and actual observations (the ground truth) [19]. Then the formula for MSE is given in equation 1:

$$MSE = \left(\frac{1}{n}\right)\sum_{i=1}^{n}(\hat{V}_i - V_i)^2 \quad (1)$$

Where $\hat{V}_i$ be the vector denoting actual values of n number of predictions and V_i be a vector representing n number of true values.

Finally, autoencoder produces a learning feature representation of the tweet data that will be used in the k-means clustering algorithm.

3.3. Defining Handcrafted features

Four types of hand-crafted features were used for distinguishing the tweet's quality content: (1) representational and structural features (2) Well-formedness features (3) author profile features(4) interaction and behavioral features.

(1) Representational and structural features: based on how the tweet content is presented including

- Tweet length: computed by counting the number of words in the tweet;
- Existence of hyperlinks: this aims to identify if the tweet contains hyperlinks and to count the number of hyperlinks per tweet. Providing referencing information is a useful component to support the information contained in the tweet and to confirm the credibility of the tweet content;
- Existence of hashtags: to identify if the tweet contains hashtags and to count the number of hashtags per tweet;
- Named entity presence: to identify named entities such as names of persons, locations and organizations using DBpedia Spotlight [6] and count their numbers. Named entities indicate the discussed topics specially in news articles by enrishing the news content;
- Question mark presence : to identify if the tweet text contains exclamation or question marks in order to measure the degree of certainty of tweet author (if this is a question or not).

(2) Well-formedness features: based on well-written, grammatically correct and understandable tweets. We detect spelling mistakes and the use of punctuation excessively in ill-formed tweets. These features were measured using features that include the following criteria that can be applied to any given tweet:

- Spelling check : percentage of misspelled words in the tweet compared to the total number of words in the tweet. A tweet with spelling errors decreases in quality. The score is computed using Hunspell [2]spell checker and morphological analyzer library;
- Grammatical check: the score is computed based on the number of grammatical errors or ambiguities in the tweet identified by Queequeg [3]grammar checker;
- Capitalize beginning of tweet: this aims to identify if the first terms begin with capital letters;
- Number of repeated characters.

(3) Author profile features: characteristics of the author's profile:

- Presence of author profile description;
- Author's profile description length.

(4) Interaction-features: based on author's behavioral interaction with others. These features were incorporated using:

- The number of re-tweets of the author's tweet;
- The number of replies to the author's tweet;
- Percentages of tweets that include a mention of the author.

3.4. K-means Clustering

Given the unlabelled data (i.e., data without any class-labels) available on our dataset, we choose clustering that performed well in unsupervised settings. Clustering is a type of unsupervised machine learning aiming to group a given set of unlabelled data instances (objects) into disjoint clusters (meaningful groups) according to some notion of similarity [24]. We used k-means clustering algorithm to classify a given data set (set of features) into k-groups (k-clusters fixed a priori) [17]. The k-means algorithm assigns each data point only to one of K-groups (to the closest cluster and to the nearest centroid) by minimizing the sum of squared distance between the data points and all the cluster's centroids in each cluster (33). For our clustering process, we need to group our set of features that are provided in two clusters one for high quality tweets and the other for low quality tweets.

[2] https://cran.r-project.org/web/packages/hunspell/vignettes/intro.html
[3] https://github.com/f-frhs/queequeg

Our K-means clustering algorithm is implemented using euclidean distance metric [8] which measures the similarities between data points (features' similarity) as shown in equation 2.

$$Dist(x,y) = \sqrt{\sum_{i=1}^{n}(x-y)^2} \tag{2}$$

3.5. Features' ranking

After the clustering process, we need to select the best set of features having the big contribution to distinguish clusters. In the literature, many feature selection methods based on importance measures have been proposed to evaluate the importance of features for clustering or classification, such as information gain-based, correlation-based and association-rule-mining-based feature selection. We chose information gain (IGain) as a feature selection method which returns a ranking score (weight) for each feature in the cluster. The objective here is to deduct the optimal and relevant set of features for each cluster by measuring how much each feature relates to the cluster or class label. The information gain-based method selects features by measuring entropy, which is the measurement of its unpredictability or impurity of the data with the same class (cluster) [21]. Let data set D with k class labels, the Shannon entropy of the given data set is shown in equation 3:

$$H(D) = -\sum_{i=1}^{k} P(i)logP(i) \tag{3}$$

Where p(i) is the probability of class i which calculates the proportion of each class i in the data set D. Let feature set $F = \{F_1, F_2, ..., F_m\}$, data set D is divided into k parts $D = \{d_1, d_2, ..., d_k\}$ according to feature F_j. The IGain of the feature F_j for the data set D is given by this formula 4 :

$$IGain(F_j) = H(D) - \sum_{i=1}^{k}\left(\frac{d_i}{D}\right)H(di) \tag{4}$$

Where H(d_i) is the entropy of the i^{th} subset generated by dividing D based on feature F_j. In the clustering process, if the IGain score of the feature F_j is higher, the potential of this feature to the clustering results is greater.

3.6. Ranking process

For our task the goal is to find the most relevant tweets in response to a given query (topic). So, for each topic, we aim to rank tweets in each cluster by measuring the separation distance between the data points and the cluster's centroid. For the ranking process, we estimate that the level of the quality of features increases when approaching the cluster's centroid. Then, we compute the average distance from all data points in the same cluster C_i and finally we rank tweets by the distance of inter-cluster. Based on the set of features relative to a specific tweet belonging to a cluster C_i, we rank tweets having the minimum distance in the top list. Given a set of data points $\{x_1, x_2, ..., x_n\}$, assigned in a set of k clusters $C = \{C_1, C_2, ..., C_k\}$ $(k \leq n)$, we calculate the distance ($minD$) between the data points and the cluster's centroid ($cent$) for each cluster. We chose the minimum distance of inter-cluster based on this formula 5:

$$minD = argmin \sum_{\forall\{X_j \in C_i\}} \|(X_j - cent)^2\| \tag{5}$$

4. Evaluation

To evaluate the performance of our ranking approach, we realized a series of experiments. The two main objectives of this evaluation consist in studying firstly, by features analysis from k-means clustering results, how the use of learned features from deep learning autoencoder shows high-level representations of features than the hand-engineered ones. And secondly, the impact of the clustering process to distinguish tweet's quality content and especially to improve ranking effectiveness. The evaluation of the results of the clustering process is incorporated in the evaluation of the ranking approach.

4.1. Dataset Description

For our evaluation, we used data set from TREC'11 microblog track which released 16 million tweets and 50 topics. By the guidelines of this track, the final ranked results of the relevant tweets must be ordered chronologically. For our work, we used only the relevant tweets based on relevance score with BM25 and ordered by their publishing times (baseline). We used the judgment file of TREC (qrels) for the evaluation of results. For the k-means clustering process, we used all TREC data set presented in JSON format to extract different sets of features about the authors and their tweets.

4.2. Results and Analysis

In this section, we will present the experimental results of our proposed ranking approach based on k-means clustering. The evaluation measures are precision and mean average precision (MAP) for the full set of topics. To show the ability of our proposed ranking approach to return the top high quality tweets in retrieval results, we therefore use the P@5 (precision at 5 tweets), p@10, p@20 and p@30 precision which correspond to the set of top relevant tweets with high quality content respectively over the top 5, 10, 20 and 30 retrieved tweets.

Features' analysis We begin with features' analysis to study the importance of each feature for the clustering task. We aim to determine, in detail, which attributes in the set of the feature vectors are most important and relevant to differentiate between clusters using the IGain method. We calculate IGain of each feature (feature weight) with respect to the cluster label. Figure 3 lists all features for each cluster (learned features and handcrafted features) with their IG scores.

We observe that the learned features based on text tweet significantly outperform hand-crafted features in terms of IGain in both clusters, showing that this is the optimal feature in the clustering. In addition, we observe that the feature based on author's description is an important feature to distinguish clusters. Generally not all authors put a description in their Twitter profile. Only names of public figures or experts of a particular domain or organizations, are supposed to put descriptions in their profiles. We can see that this feature has a higher score (IGain=0.46) in cluster 1, so, this feature can be considered as a characteristic feature of 'cluster 1'. Moreover, well-formedness based features including percentage of spelling errors (% of misspelled words) and percentage of grammar errors (% of grammatical errors) contribute most to the clustering with a high IGain in both clusters. The feature of hyperlink presence (Contains hyperlinks) has a good score (IGain=0.24) in 'cluster 1'. A tweet with URL generally contains useful information. However, hashtag features (Contains hashtags) present in tweets have almost the same IGain in both clusters and provide no useful information for the clustering (all users uses hashtags to indicate the discussed topic). Structural features including tweet length have a slight influence in clustering. But, the feature of the number of repeated characters (N.RepeatedChar) has more influence in 'cluster 2'. In addition, following our observation of the features' ranking results, some of the hand-crafted features, however, do not provide useful indication to clustering. We notice that the interaction based features such as number of retweets (N.retweets), number of replies (N.replies) and number of mentions (N.mentions) are not important features in the clustering performance. We can see that these features are not useful for selecting clusters because all users in Twitter (e.g., experts, organizations, influences, ordinary users, etc.) can use these behaviors to interact with others. Also, capitalize beginning of tweet

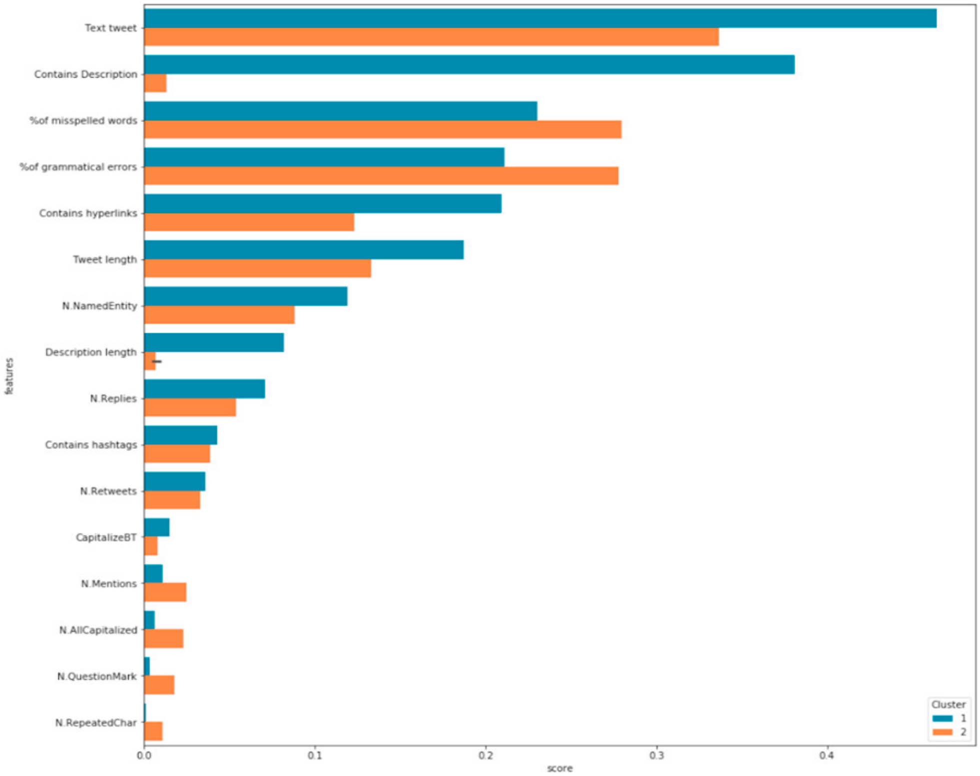

Fig. 3. Feature ranking by informatin gain

feature (CapitalizeBT) present a low IGain in both clusters, which proves that it has no influence on the cluster label. Also, By feature analysis, we conclude that the textual content of the tweet is a stronger indicator of the quality of tweets compared to non-textual indicators such as interaction features. A conclusion is also derived from the scoring features' results. We can say that the use of deep learning algorithm such as autoencoder neural network can automatically learn better and high-level feature representation than the hand-crafted ones. These representations strongly affect the quality of clustering. Therefore, based on the features' ranking, high scoring features are most relevant and significant for the clustering process.

Ranking results We evaluate the ranking results using the results of k-means clustering. This evaluation aims to distinguish high-quality and low-quality tweets in clusters. According to the ranking results in Table 1, 'Cluster 1' performs the best among all the measures, achieving the highest precision with high-quality ranking. It is clear how the results based on 'Cluster 1' outperform in all measures the classic ranking based on the reverse chronological order. The results of the ranking prove that 'Cluster 1' yields high precision indicating the presence of high-quality tweets. Ranking results indicate that 'Cluster 2' has low quality tweets with less precision. Our experimental results show that the proposed ranking approach based on clustering with learned features (LF) is effective and promising. Our ranking approach significantly increases the performance of tweet ranking compared to the ranking result based on reverse chronological order (run baseline). Especially, this proves that ranking tweets in reverse chronological order does not guarantee that the most interesting tweets appear on top of the list of relevant tweets.

Table 1. Ranking results based on k-means clustering with learned features

Run	P@5	P@10	P@20	P@30	MAP	MAP-Gain
run- baseline	0.1288	0.1337	0.1321	0.1293	0.1034	-
Cluster 1 (with LF)	0.2310	0.2267	0.2226	0.1968	0.1881	81%
Cluster 2 (with LF)	0.1288	0.1337	0.1371	0.1352	0.1108	7%

In Table 2, the results of ranking in the two clusters do not present a big difference in terms of precision (almost similar) and we can not deduce which cluster contains high quality. This proves the usefulness of the learned features based on deep learning autoencoder to distinguish high quality and low quality tweets in the clustering process. Also, we conclude that the quality of clustering strongly affects the quality of the ranking.

Table 2. Ranking results based on k-means clustering without learned features

Run	P@5	P@10	P@20	P@30	MAP	MAP-Gain
run- baseline	0.1288	0.1337	0.1321	0.1293	0.1034	-
Cluster 1 (without LF)	0.2145	0.2145	0.2059	0.1882	0.1510	46%
Cluster 2 (without LF)	0.1928	0.1928	0.1870	0.1611	0.1356	31%

5. Conclusion and Future Work

In this paper, we focus on the problem of ranking tweets, and particularly on retrieving high quality content. We proposed a ranking approach based on k-means clustering to distinguish high quality and low quality tweets. The clustering algorithm is based on learning features from deep learning autoencoder and hand-crafted features from tweets' content and authors' profiles. Then, we demonstrate that the use of deep learning algorithm such as auto encoder neural network can automatically learn better high-level feature representations than the hand-engineered ones. Also, our proposed ranking approach increases the performance of the retrieval effectiveness compared to the reverse-chronological order ranking. We note that the results of this work can be deployed on other microblogging services or other social networks. In the future, we plan to use a deep learning algorithm such as auto encoder neural network to see how the quality of queries given by users in Twitter to express their information need affects the search results effectiveness.

References

[1] Chicco, D., Sadowski, P., Baldi, P., 2014. Deep autoencoder neural networks for gene ontology annotation predictions, in: Proceedings of the 5th ACM conference on bioinformatics, computational biology, and health informatics, ACM. pp. 533–540.

[2] Choi, J., Croft, W.B., Kim, J.Y., 2012. Quality models for microblog retrieval, in: Proceedings of the 21st ACM international conference on Information and knowledge management, ACM. pp. 1834–1838.

[3] Duan, Y., Jiang, L., Qin, T., Zhou, M., Shum, H.Y., 2010. An empirical study on learning to rank of tweets, in: Proceedings of the 23rd International Conference on Computational Linguistics, Association for Computational Linguistics. pp. 295–303.

[4] Efron, M., Lin, J., He, J., De Vries, A., 2014. Temporal feedback for tweet search with non-parametric density estimation, in: Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval, ACM. pp. 33–42.

[5] Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning. MIT press.

[6] Ibtihel, B.L., Lobna, H., Maher, B.J., 2018. A semantic approach for tweet categorization. Procedia Computer Science 126, 335–344.

[7] Jia, L., Yu, C., Meng, W., 2013. The impacts of structural difference and temporality of tweets on retrieval effectiveness. ACM Transactions on Information Systems (TOIS) 31, 21.

[8] Kaoungku, N., Kerdprasop, K., Kerdprasop, N., 2017. A method to clustering the feature ranking on data classification using an ensemble feature selection. International Journal of Future Computer and Communication 6, 81–85.

[9] Kong, S., Feng, L., 2011. A tweet-centric approach for topic-specific author ranking in micro-blog, in: International Conference on Advanced Data Mining and Applications, Springer. pp. 138–151.

[10] LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. nature 521, 436.

[11] Massoudi, K., Tsagkias, M., De Rijke, M., Weerkamp, W., 2011. Incorporating query expansion and quality indicators in searching microblog posts, in: European Conference on Information Retrieval, Springer. pp. 362–367.
[12] Nagmoti, R., Teredesai, A., De Cock, M., 2010. Ranking approaches for microblog search, in: Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology-Volume 01, IEEE Computer Society. pp. 153–157.
[13] Pal, D., Mitra, M., Bhattacharya, S., 2015. Exploring query categorisation for query expansion: A study. arXiv preprint arXiv:1509.05567 .
[14] Quillot, M., Delorme, A., 2017. Re-ranking microblogs using word2vec in microblog search task, university of avignon .
[15] Ravikumar, S., Balakrishnan, R., Kambhampati, S., 2012. Ranking tweets considering trust and relevance, in: Proceedings of the Ninth International Workshop on Information Integration on the Web, ACM. p. 4.
[16] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. Neural networks 61, 85–117.
[17] Singh, A., Yadav, A., Rana, A., 2013. K-means with three different distance metrics. International Journal of Computer Applications 67.
[18] Soboroff, I., Ounis, I., Macdonald, C., Lin, J.J., 2012. Overview of the trec-2012 microblog track., in: TREC, Citeseer. p. 20.
[19] Strub, F., Mary, J., 2015. Collaborative filtering with stacked denoising autoencoders and sparse inputs, in: NIPS workshop on machine learning for eCommerce.
[20] Teevan, J., Ramage, D., Morris, M.R., 2011. # twittersearch: a comparison of microblog search and web search, in: Proceedings of the fourth ACM international conference on Web search and data mining, ACM. pp. 35–44.
[21] Uğuz, H., 2011. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems 24, 1024–1032.
[22] Uysal, I., Croft, W.B., 2011. User oriented tweet ranking: a filtering approach to microblogs, in: Proceedings of the 20th ACM international conference on Information and knowledge management, ACM. pp. 2261–2264.
[23] Vosecky, J., Leung, K.W.T., Ng, W., 2012. Searching for quality microblog posts: Filtering and ranking based on content analysis and implicit links, in: International Conference on Database Systems for Advanced Applications, Springer. pp. 397–413.
[24] Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al., 2001. Constrained k-means clustering with background knowledge, in: Icml, pp. 577–584.
[25] Wang, W., Duan, L., Koul, A., Sheth, A.P., 2014. Yourank: Let user engagement rank microblog search results, in: Eighth International AAAI Conference on Weblogs and Social Media.
[26] Wei, Z., Gao, W., El-Ganainy, T., Magdy, W., Wong, K.F., 2014. Ranking model selection and fusion for effective microblog search, in: Proceedings of the first international workshop on Social media retrieval and analysis, ACM. pp. 21–26.
[27] Zhai, Y., Li, X., Chen, J., Fan, X., Cheung, W.K., 2014. A novel topical authority-based microblog ranking, in: Asia-Pacific Web Conference, Springer. pp. 105–116.
[28] Zingla, M.A., Chiraz, L., Slimani, Y., 2016. Short query expansion for microblog retrieval. Procedia Computer Science 96, 225–234.