Post-hoc Explanation for Twitter List Recommendation

Havva Alizadeh Noughabi (h.alizadeh@mail.um.ac.ir)  
Ferdowsi University of Mashhad

Behshid Behkamal (behkamal@um.ac.ir)  
Ferdowsi University of Mashhad

Fattane Zarrinkalam (fzarrink@uoguelph.ca)  
University of Guelph

Mohsen Kahani (kahani@um.ac.ir)  
Ferdowsi University of Mashhad

Research Article

Keywords:

DOI: https://doi.org/

License: This work is licensed under a Creative Commons Attribution 4.0 International License. Read Full License

Additional Declarations: No competing interests reported.
Post-hoc Explanation for Twitter List Recommendation

Havva Alizadeh Noughabi¹, Behshid Behkamal*, Fattane Zarrinkalam² and Mohsen Kahani¹

¹Department of Computer Engineering, Ferdowsi University of Mashhad, Iran.
²School of Engineering, University of Guelph, Canada.

*Corresponding author(s). E-mail(s): behkamal@um.ac.ir;
Contributing authors: h.alizadeh@mail.um.ac.ir; fzarrink@uoguelph.ca; kahani@um.ac.ir;

Abstract

Twitter List recommender systems have the ability to generate accurate recommendations, but since they utilize heterogeneous user and List information on Twitter and usually apply complex hybrid prediction models, they cannot provide user-friendly intrinsic explanations. In this paper, we propose an explanation model to provide post-hoc explanations for recommended Twitter Lists based on the user’s own actions; and consequently benefits to improve recommendation acceptance by end users. The proposed model includes two main components: (1) candidate explanation generation in which the most semantically related actions of a user on Twitter to the recommended List are retrieved as candidate explanations; and (2) explanation ranking to re-rank candidates based on relatedness to the List and their informativeness. Through experiments on a real-world Twitter dataset, we demonstrate that the proposed explanation model can effectively generate related, informative and useful post-hoc explanations for the recommended Lists to users, while maintaining parity in recommendation performance.

Keywords: Explainable recommendation, Post-hoc explanation, Explanation ranking, Twitter List recommendation.
1 Introduction

With the tremendous growth of user generated content on social media, such as Twitter, many social media platforms have provided mechanisms to help users to organize related information into a single bin. For example, Lists were introduced by Twitter in 2009 as a useful feature to cope with the information overload problem [1]. A Twitter List is a curated group of Twitter accounts that can be freely subscribed by a community of users, who are interested in the topics of the List [2, 3].

Since more and more users are creating Lists on Twitter to organize information, it is crucial to develop techniques that can automatically recommend genuine, authoritative and topically related Lists to users. Despite the great success of various List recommender systems by combining several heterogeneous user and List features and applying complex hybrid models [4, 5], their most significant drawback is the lack of explainability, which leaves users with little understanding of how particular decisions (recommended List) are made by these models.

Nowadays, explainability in the recommender systems is a crucial requirement as it benefits users to make more informed decisions and helps them to build and maintain their trust in the recommendations, which, on the other hand, sets the stage for the sustainability of social media [6]. However, one of the fundamental trade-offs in recommender systems is that of accuracy and explainability [7]. To mitigate this trade-off, post-hoc explainability has recently gained considerable interest in the domain of recommender systems, by establishing approaches to provide explanations after a decision has been made by a model [8–11]. Therefore, in order to maintain the predictive accuracy of current List recommendation models whilst yielding explanations, we focus on providing post-hoc explanations for the recommended Lists to users.

To achieve this goal, we only rely on the user’s own action on social media by processing the social posts that the user interacts with (e.g. by liking, publishing or sharing) herself and select some of her most informative and related post to the recommended List as post-hoc explanation. More specifically, our proposed explanation model includes two main components: (1) candidate explanation generation in which we retrieve the most semantically related posts of a user on Twitter to the recommended List and consider them as candidate explanations; and (2) explanation ranking to rank the generated candidates based on their relevance to the List and their informativeness. For example, consider a Twitter List named Crypto Currency that is recommended to a user called ‘Mary’ who has interacted with the following posts on Twitter.

A “Looking forward to Pfizer formally making this request to the New Zealand government. Will be a massive relief to be able to get my younger kids vaccinated.”

B “Barcelona are reportedly planning moves for THREE Chelsea players! Latest gossip! #bbcfourball"

C “Everyone is going to want #Bitcoin for Christmas.”
\textbf{A TEX template}

\textbf{Post-hoc Explanation for Twitter List Recommendation}

D. “\textit{Don’t trip, buy the dip.}”
E. “\textit{Bitcoin’s plan is to use the money to master energy so that we can save the Earth and all life upon it, for billions of years.}”

Based on the keywords explicitly mentioned in her tweets, one could easily infer that the tweets C, D and E are related to \textit{Crypto Currency} and can be considered as candidate explanations to answer why \textit{Crypto Currency} List is recommended to ‘Mary’. However, since tweet E is more informative by providing more information to better explain the recommended List, it is finally selected as a post-hoc explanation.

The key contributions of our work are as follows:

- We propose a post-hoc explanation model for recommended Lists to users on social media by processing the users’ own social actions.
- We propose an explanation ranking model to rank the generated explanations for a recommended List by introducing three categories of features, namely content relevance features, post specific features and publishers’ authority features.
- We show through our experiments on a real-world Twitter dataset that our proposed post-hoc explanation model is able to provide related and informative explanations for the recommended List while maintaining the predictive accuracy of current List recommendation models.

The rest of this paper is organized as follows. In the next section, we discuss related works in two subsections including post-hoc explanations and explainable social recommendation. In Section 3, we formulate the proposed model to provide an explanation for the recommended Twitter List and then introduce our features used in the explanation ranking component. We then evaluate our work from both qualitative and quantitative perspectives in Section 4, followed by Section 5 which concludes the paper and presents some directions for future works.

\section{Related Work}

In this work, we propose a post-hoc explanation model for List recommendations on social media. Thus, our work is related to \textit{post-hoc explainable recommendation} models that develop an explanation model to explain recommendations after the decision has been made by a recommender system and \textit{explainable social recommendations} that provide explainable recommender systems in the social media environment. In this section, we review the related works in these two areas.

\subsection{Post-hoc Explainable Recommendations}

Post-hoc explainable recommendation models provide an explanation for an item after it is recommended by applying different techniques, such as association rule mining [12], reinforcement learning [13] and causal analysis [14]. For
example, Peake and Wang [12] have applied association rule mining to provide post-hoc explanations for the recommendations generated based on the history of each user. They have first considered a user’s history and her recommendations as a transaction for each user and then processed the transactions of all the users extract association rules which are used to explain the recommendations. Xu et al. [14] have developed a post-hoc explanation model by applying causal rule mining in which personalized explanations are generated for the sequential recommendation models. Specifically, the causal explanations are extracted through a perturbation model and a causal rule learning model. Cheng et al. [15] have employed influence analysis for developing a post-hoc explanation model named FIA (Fast Influence Analysis) to explain the prediction results of latent factor models. Specifically, they have calculated the impact of past user-item interactions on the results of latent factor models’ predictions, and generated explanations based on the most influential interactions.

Counterfactual reasoning is another approach which is also adopted in some post-hoc explainable recommender systems to provide counterfactual explanations by investigating what input should be changed and, by how much, to obtain a different recommendation. For example, Ghazimatin et al. [16] have generated counterfactual explanations for recommended items by proposing a searching algorithm on a heterogeneous graph to find a minimal set of users’ historical actions such that by removing them, the model will recommend different items. Similarly, Tan et al. [17] have proposed a framework named Counterfactual Explainable Recommendation (CountER) which generates explanations based on counterfactual changes on item aspects to generate. These altered aspects constitute the explanation of why the original item is recommended. Later, they have extended [17] by proposing a Counterfactual and Factual reasoning framework to generates GNN (Graph Neural Network) explanations by simultaneously considering the necessity and sufficiency of the explanations [18].

Making use of reviews and knowledge graph information have recently attracted considerable attention by researchers to generate post-hoc explanations. For example, [19] have proposed a post-hoc explanation strategy that takes a recommended item as input and a set of reviews and provide as output a post-hoc natural language justification that is independent of the recommender system. Similarly, Wang et al. [13] have designed a reinforcement learning framework that can explain any recommendation and can flexibly control the explanation quality. In their framework, the recommendation model is a part of the environment and supports the explanation model to provide better post-hoc explanations. To leverage knowledge graphs to generate post-hoc explanations, Zhang et al. [20] have proposed a knowledge distillation approach to explain black-box recommendation models. The authors proposed an end-to-end joint learning framework to combine the advantages of embedding-based and path-based recommendation models. Chen and Miyazaki [21] have proposed a post-hoc explanation model that takes a recommendation as input.
and provides an explanation based on the paths in a knowledge graph. They applied third party knowledge bases (Wikidata) to enhance the explainability of recommendation models by providing diverse and high-quality personalized explanations. Similarly, Du et al. [22] have proposed a personalized post-hoc approach by applying algorithmic optimizations based on the extraction of sub-ontology and postorder graph traversal over DBpedia knowledge graph.

Compared with post-hoc explainable models, our work focuses on the user’s own actions to obtain a simple explanation for Twitter List recommendations, generated by an accurate and sophisticated recommender model. In fact, we believe such explanation persuades users to a certain recommendation, without expressing the other user’s actions or interests. To the best of our knowledge, only few studies have focused on user’s own action based explainable recommendations. For example, [16] have generated counterfactual explanations by identifying minimum sets of user actions that their absence would change the top-ranked recommendation to a different item. In our context in a social environment, the users’ own social interactions such as share/like a post will be used to generate an explanation.

2.2 Explainable Social Recommendation

Providing explainability for social recommender systems positively contributes to the user experience with social media, such as trust, understandability, and satisfaction which consequently leads to maintaining the sustainability of social media [6, 23, 24].

For a recommended item to a user in social media, recommender systems have mainly focused on the user’s social network to provide explanations (e.g., “I am recommending this to you because your friend [X] likes it.”). For example, Sharma et al. [25] have investigated the effects of social explanations in music recommendation and for each recommended music to a user provide the number of her friends that liked the music. Similarly, Wang et al. [26] have generated social explanations in the form of “A and B also like the item”. They presented an algorithm to identify an optimal set of users to be put in the explanation in order to generate the most persuasive social explanation. More recently, Park et al. [27] have proposed UniWalk that explains why an item is recommended to a target user based on her social network. UniWalk combines both social network and rating data into a unified graph, and then used network embedding on the unified graph to extract latent features of items and users, and finally recommended items to each user through the features. UniWalk also provides meta-explanations (explanation for explanation) which answer the question, why the other users and items are determined to be similar to the target user and the recommended item, respectively.

There is another line of research in which social media data is represented as a heterogeneous information network (also known as knowledge graph) and then explanations are provided in the form of paths between users and recommended items. For example, Shi et al. [28] have proposed a semantic path-based recommendation method named SemRec to predict the rating scores of users
on items and learn the prioritized and personalized weight representing user preferences on paths to provide explanations for the recommended items. Similarly, a knowledge distillation approach has been proposed by Zhang et al. [20] to explain black-box models for recommendation. To combine the benefits of embedding-based and path-based recommendation models, the authors have suggested an end-to-end joint learning framework. The proposed approach, given an embedding-based model that create black-box recommendations, interprets its recommendation relying on differentiable paths on knowledge graphs; on the other hand, the differentiable paths regularize the black-box model with information in knowledge graph to improve performance.

Recently, the content of existing reviews and posts published by users to express their opinions on social media has been considered as explanations by researchers in the domain of social recommender system [29–31]. For example, in order to predict item ratings based on user opinions and social relations, Ren et al. [29] introduced a social collaborative viewpoint regression (sCVR) model. This model offers viewpoints as explanations, where a viewpoint is a tuple of concept, topic, and sentiment label derived from both user reviews and trusted social relations. Zheng et al. [30] have developed an explainable social relation extraction technique using user interest-based hierarchical semantic specificity matrices. They explain users’ social follow relation with their relevant interest subjects from multi-dimensional and multi-granularity aspects of micro-blog contents. Precisely, they explain the relation between user pairs based on various contributions of words or sentences in the fine-grained interest subjects. More recently, Zheng et al. [31] have proposed an explainable link prediction approach based on fusion embedding of heterogeneous context information. In particular, external knowledge semantics were used to create a fusion user embedding method. The significant finding in the explainable proposed approach is that semantic knowledge of fusion user embedding may more accurately predict and explain the behavioral link relations between users.

To the best of our knowledge, there is no research on applying explainability for Twitter List recommendation which motivated us to provide a persuasive explanation based on the user’s own actions.

3 Proposed Approach

The objective of our work is to provide post-hoc explanations for a Twitter List recommended to a user by a recommendation system. To be able to achieve this goal, we only rely on the posts that the user interacts with on Twitter (e.g. by liking, publishing or sharing) and select the most informative and related tweet to the recommended List as post-hoc explanations for the recommended List to the user. Formally, let $U$ be the set of users and $L$ be the set of Twitter Lists, our problem is defined as follows:

**Problem Definition.** Let $M_u = \{m_1, m_2, \ldots, m_N\}$ is a set of social posts that the user $u$ has interacted with on social media, given a recommended List $l \in L$ to a user $u \in U$, we aim to identify the most informative and relevant post
of user $u$ to the recommended List $l$, i.e., $m \in M_u$ as a post-hoc explanation for the user-List pair $(u, l)$.

Figure 1 illustrates an overview of the proposed approach to address the aforementioned problem. As depicted in Figure 1, we divide this problem into two subproblems in which the output of the first sub-problem becomes the input of the second one: (1) candidate explanation generation to select the most semantically related social posts of a user to the recommended List; and (2) explanation ranking to rank the generated candidates and choose the most relevant, informative and useful explanations. In the following sections, we describe our proposed approach for addressing these two subproblems.

### 3.1 Candidate Explanation Generation

Given a List $l \in \mathbb{L}$ recommended to user $u \in \mathbb{U}$ by a recommender system and the user’s social posts $M_u$, the goal of candidate explanation generation is to
select a set of promising candidates from the user’s posts, denoted by $C_u(l)$, as potential post-hoc explanations for the recommended List $l$. We argue that the user’s social posts which are topically related to the recommended List $l$ can be considered as the potential explanations. Therefore, we first need to identify the dominant topics of each List $l \in \mathbb{L}$ on Twitter based on the posts published in the List $l$, denoted as $M_l$. A topic $z$ has traditionally been defined as a semantically coherent theme which has received substantial attention from the users. Considering $M$, the set of posts published by users $U$ and in the Lists $\mathbb{L}$, it is possible to extract active topics $Z$ using topic modeling methods.

Topic modeling from social posts has already been studied in the literature and therefore is not the focus of our work and we are able to work with any topic modeling. LDA [32] is one of the well-known unsupervised algorithms used for identifying latent topics from a corpus of documents. However, as it is designed for regular documents, it might not perform so well on short, noisy and informal texts like tweets [33–36]. To address this issue, we employ BERTopic which is shown to perform better when modelling short and unstructured texts as in the case of Twitter data [37–39]. BERTopic first generates document embeddings using pre-trained transformer-based language models. Then it clusters these embeddings, and finally generates topic representations with the class-based TF-IDF procedure.

To extract active topics from $M$ by BERTopic, each post $m \in M$ is considered as a single document. By assigning each document to a single topic, BERTopic generates two artifacts: (1) a set of $K$ topic-word distributions, where each topic word distribution associated with a topic $z \in Z$ represents active topics in $M$, (2) The topic of each post $m$, i.e., $z_m \in Z$. Thus, given $Z = \{z_1, z_2, \ldots, z_K\}$ be $K$ active topics on Twitter, to extract the dominant topics of each List $l \in \mathbb{L}$, we define List topic profile as follows:

**List Topic Profile.** Let $M_l = \{m_1, m_2, \ldots, m_N\}$ be a set of posts published in the List $l \in \mathbb{L}$ and $Z$ be a set of $K$ topics, a List profile $P(l)$ is represented by a vector of weights over $K$ topics, i.e., $(f_l(z_1), \ldots, f_l(z_K))$, where $(f_l(z_k))$ is the number of posts of the List $l$ that are related to topic $z_k$, i.e., $|M_{z_k}^l| \subset M_l$. The List representation is normalized using the $L1$-norm.

Next, we select the topically related posts of the user $u$ to the recommended List $l$, as candidate explanations, $C_u(l)$, as follows:

$$C_u(l) = \{m \in M_u \mid P(l).P(m) \geq \alpha\}$$

where $P(m)$ is a $K$ element one-hot vector over the topics $Z$ for post $m$, where the value of all the cells are 0, with the exception of a single 1 in the cell used uniquely to identify the topic of the post $m$, i.e., $z_m \in Z$ identified by BERTopic. Further, $\alpha$ indicates the minimum threshold for the weight of each topic of the List to be considered as a dominant topic for the List $l$. 
3.2 Explanation Ranking

Given \( C_u(l) \), as a set of candidate explanations for the List \( l \) recommended to user \( u \), our goal is to re-rank the retrieved candidates and consider the most relevant and informative post \( m \in C_u(l) \) as the final explanation. As the usual number of candidate explanations is more than one (Our experiments show that the number of such candidates can vary from a few ten to even hundreds), it is imperative that we rank these explanations and present only one, to prevent a cognitive overload.

To address this problem, we represent each post-List pair \( (m \in C_u(l), l \in L) \) by a multi-dimensional feature vector, where each dimension of the vector is a feature indicating how much relevant and informative is \( m \in C_u(l) \) with respect to the dominant topics of \( l \in L \). Given the features, we employ a learning to rank framework to select the most pertinent features and rank the candidate explanations \( C_u(l) \) for each recommended List \( l \).

To define features, we focus on **Informativeness** and **Relevance** of candidate explanations. Adopted from [40], in our work, Informativeness means degree of information acceptability which can explain the recommended List understandable to maximum people and Relevance means the degree of content closeness to the recommended List. Specifically, we propose three categories of features for explanation ranking: (1) Content relevance features, (2) Post specific features and (3) Publishers’ authority features. These features are introduced in Table 1 and in the following, we elaborate more on the definition of these features.

### 3.2.1 Content Relevance Features

Content relevance features measure the relatedness of a candidate explanation to the recommended List. Our intuition is that the more semantically similar an explanation to the recommended List, the more persuasive the explanation will be for the user. The description of two features of this category is given below.

- **Topical Relatedness:** It measures the content relevance of the candidate post \( m \in C_u(l) \) and recommended List \( l \). Given \( z_m \) as the topic of the candidate post assigned by our topic modeling technique as described in Section 3.1, we consider \( f_l(z_m) \) from the Topic Profile of List \( l \) as the degree of topical relevance of \( m \) to \( l \).

- **Relevance to Hashtags:** The candidate post \( m \in C_u(l) \) may include a number of hashtags to emphasize the key words of it. However all those hashtags may not be relevant to the desired List. This feature estimates the count of hashtags in the candidate post \( m \in C_u(l) \) that appeared as hashtags through recommended List history to determine the relevance.

### 3.2.2 Post Specific Features

This category of features measures the quality of the candidate social post regardless of its relatedness to the recommended List. Our hypothesis is that
the more informative and popular a post is, the most useful will be the post to be considered as an explanation. Inspired by the features introduced in the literature for tweet ranking [41–45], we consider the following features in this category:

- **Length**: It is measured by the number of words that a social post contains. Intuitively, a long sentence is apt to contain more information than a short one. We use length of explanation as a measure of the information richness of it.
- **Retweet Count**: It is defined as the number of times a post is retweeted. The basic idea is that the more frequently a post is retweeted, the more informative and useful it is.
- **Favorite Count**: ‘Like’ facility in Twitter determines level of appreciation of the posts’ content. The number of likes may influence how users perceive the explanations. Therefore, the number of the times people have expressed a positive feeling about a post may be a potential indicator of its quality.
- **URL Count**: On Twitter, publishers often include a URL as a supplement in their posts to point to further information on another web page. Therefore, the number of URLs in an explanation may influence its informativeness. Therefore, it can be considered as an additional quality measure for a social post.
- **Hashtag Count**: Intuitively, the publisher spends time on tagging the post because she thinks the post may be useful. The number of hashtags added to a post make an explanation more informative and useful.

3.2.3 Publishers’ Authority Features:

This category of features measure the degree of authority of the publisher of a candidate post. It is noted that, the candidate posts which are selected for a user are not all published by the user herself and they might be among the posts that the user retweets or likes. Therefore, the candidate posts may have different publishers. Our hypothesis is that candidate posts which are published by a more authoritative user on social media may be higher quality and more persuasive for the user to be considered as an explanation for the recommended List. Adopted from [41–44], we considered the following features as some indicators to estimate the authority of a target user:

- **Follower Count**: This feature records the number of followers a user has.
- **Status Count**: This feature measures the number of tweets ever posted by a user.
- **Mention Count**: This feature estimates the number of times the user is mentioned in tweets.
Table 1 The set of features for a given tweet-List pair \((m, l)\) for explanation ranking.

<table>
<thead>
<tr>
<th>Category</th>
<th>Feature Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>Topical Relatedness</td>
<td>The relevance between an explanation tweet and the posting history of the List</td>
</tr>
<tr>
<td>Relevance</td>
<td>Relevance to Hashtags</td>
<td>The count of hashtags in an explanation tweet that appeared as hashtags through List history</td>
</tr>
<tr>
<td></td>
<td>Length</td>
<td>The number of words contained in an explanation tweet</td>
</tr>
<tr>
<td>Post Specific</td>
<td>Retweet Count</td>
<td>The number of times an explanation tweet (m) has been retweeted</td>
</tr>
<tr>
<td></td>
<td>Favorite Count</td>
<td>The number of times an explanation tweet (m) has been liked</td>
</tr>
<tr>
<td></td>
<td>URL Count</td>
<td>The number of URLs contained in an explanation tweet (m)</td>
</tr>
<tr>
<td></td>
<td>Hashtag Count</td>
<td>The number of hashtags contained in an explanation tweet (m)</td>
</tr>
<tr>
<td>Publishers’</td>
<td>Follower Count</td>
<td>The number of followers of user (u)</td>
</tr>
<tr>
<td>Authority</td>
<td>Status Count</td>
<td>The number of tweets posted by user (u)</td>
</tr>
<tr>
<td></td>
<td>Mention Count</td>
<td>The number of times user (u) is referred to in tweets</td>
</tr>
</tbody>
</table>

4 Experiments

4.1 Dataset

We collected a dataset from Twitter including users and their subscribed Lists using Tweepy\(^1\) by snowball sampling of users and Lists as follows: Similar to [2], we started crawling from Lists of ‘Ashton Kutcher’, an American actor, producer, entrepreneur, and former model, who is one of the most popular users on Twitter with more than 17M followers. Given his Lists, to expand the users and Lists in our dataset, we first collected all users who have subscribed to these Lists. Then, given the new users added to our dataset, we expanded the set of Lists by collecting all Lists that these users have subscribed to. In each iteration, from each user we randomly selected a maximum of 10 Lists and from each List we randomly selected a maximum of 50 subscribers. We repeated the expansion step for 4 times which resulted in 25,788 users and 17,604 Lists and 66,583 subscribe relations between users and Lists. Then, we collected the last 3-month recent English tweets of each user (up to 2,000 tweets) and the maximum 500 recent tweets of each List. Similar to [46], to focus on the active Twitter users, we only consider Twitter users who have interacted with more than 200 English tweets within the last three months as golden users. It resulted in 1,284 users, 1,535 Lists, 6,059 subscribe relations between users and Lists and 1,581,321 tweets. Table 2 summarizes some basic statistics about our dataset. It is noted that, for each user, we collected her

\(^1\)https://www.tweepy.org/
Table 2 Basic statistics of our dataset.

<table>
<thead>
<tr>
<th></th>
<th># of users</th>
<th>#of Lists</th>
<th>#of tweets</th>
<th># of subscribe relations</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>per user</td>
<td>min:max:std</td>
<td>total</td>
<td>min:max:std</td>
</tr>
<tr>
<td># of users</td>
<td>1,284</td>
<td>5</td>
<td>1,535</td>
<td>200::1,985::446</td>
</tr>
<tr>
<td># of Lists</td>
<td>1::10::2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3 Some statistics about different types of tweets in our dataset.

<table>
<thead>
<tr>
<th></th>
<th># of original tweets</th>
<th># of retweets</th>
<th># of replies</th>
<th># liked tweets</th>
<th># of quote tweets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>per user</td>
<td>total</td>
<td>per user</td>
<td>total</td>
<td>per user</td>
</tr>
<tr>
<td># of original tweets</td>
<td>60</td>
<td>77,600</td>
<td>212</td>
<td>271,573</td>
<td>83</td>
</tr>
<tr>
<td># of retweets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>347</td>
</tr>
<tr>
<td># of replies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td># liked tweets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

original tweets, retweets, replies, liked tweets, and quote tweets. Some statistics about different each type of tweets in our dataset are reported in Table 3.

4.2 Experimental Setup

**Topic modeling.** As explained in Section 3.1, we apply BERTopic to extract topics of tweets. Before applying BERTopic, as suggested in [38], we slightly preprocessed the tweets by lowercasing its content and removing URLs, mentions, punctuations and special characters. In our work, we employ the default embedding model in BERTopic (i.e., ”all-MiniLM-L6-v2”) for encoding the tweets to dense vector embeddings. Further, we run BERTopic in auto setting which results in 539 topics for our dataset. For example, Table 4 reports 10 frequent topics extracted from our dataset and their associated top-10 words.

As defined in section 3.1, the List Topic Profile is built based on the topics of its constituent tweets and the weight of each topic is based on the number of its tweets labeled by that topic. For some Lists in which the users are talking about some diverse topics, to consider only the dominant topics to selecting the candidate explanations based on Equation 1, we set the $\alpha$ to the average value of the weights of its topics as follows:

$$\alpha = \left( \frac{1}{|S|} \right) \sum_{k=1}^{K} (f_l(z_k)) \quad , \quad S = \{ f_l(z_k) \in P(l) \mid f_l(z_k) > 0 \}$$  

where $f_l(z_k)$ is the weight of topic $z_k$ in the topic profile of List $l$. In fact, each List usually has a main subject which is equivalent to several topics in our topic modeling. In our dataset, the average number of dominant topics per List is 11 (min:1, max:38, stddev:6).

**Explanation Ranking.** We used the RankLib\(^2\) library for learning to rank in all our experiments. We adopt three machine learning methods, including one linear (Coordinate Ascent) and two non-linear (MART, LambdaMART) to train the explanation ranking model. To collect the train data, we randomly selected 100 recommended Lists and randomly selected a maximum of 10 candidate explanations for each. Finally, after annotating explanations

\(^2\)https://sourceforge.net/p/lemur/wiki/RankLib/
Table 4 Some frequent topics extracted from our dataset by BERTopic.

<table>
<thead>
<tr>
<th>Topic Theme</th>
<th>Topic Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bitcoin</td>
<td>'bitcoin', 'crypto', 'btc', 'blockchain', 'defi', 'ethereum', 'cryptocurrency', 'dip', 'market', 'buy'</td>
</tr>
<tr>
<td>Covid</td>
<td>'covid', 'deaths', 'cases', 'virus', 'pandemic', 'new cases', 'covid cases', 'covid deaths', 'coronavirus', 'hospitalizations'</td>
</tr>
<tr>
<td>AI</td>
<td>'ai', 'machinelearning', 'artificialintelligence', 'datascience', 'robot', 'bigdata', 'ml', 'learning', 'machine learning', 'artificial'</td>
</tr>
<tr>
<td>Cancer</td>
<td>'cancer', 'breast', 'breast cancer', 'heart', 'prostate', 'patients', 'prostate cancer', 'treatment', 'disease', 'surgery'</td>
</tr>
<tr>
<td>Space</td>
<td>'moon', 'space', 'nasa', 'mars', 'earth', 'launch', 'solar', 'mission', 'telescope', 'spacecraft'</td>
</tr>
<tr>
<td>Climate</td>
<td>'climate', 'cop', 'energy', 'climate change', 'fossil', 'emissions', 'change', 'gas', 'oil', 'carbon'</td>
</tr>
<tr>
<td>Music</td>
<td>'music', 'song', 'album', 'kanye', 'songs', 'wizkid', 'spotify', 'tickets', 'dj', 'dance'</td>
</tr>
<tr>
<td>Jewelry</td>
<td>'earrings', 'necklace', 'silver', 'jewelry', 'sterling', 'sterling silver', 'bracelet', 'vintage', 'pendant', 'charm'</td>
</tr>
<tr>
<td>Cybersecurity</td>
<td>'cybersecurity', 'cyber', 'security', 'infosec', 'cyber security', 'hacking', 'cyberwarrior', 'cybercrime', 'privacy', 'malware'</td>
</tr>
<tr>
<td>Github</td>
<td>'github', 'git', 'git commands', 'git github', 'learn git', 'repositories', 'commands', 'github repositories', 'github profile', 'developer'</td>
</tr>
</tbody>
</table>

by annotators according to manual annotation instruction in section 4.6, we generated the annotation pool with 935 annotated explanations. Given the train dataset, to select the best-performing model, we have applied a 5-fold cross-validation approach to evaluate different ranking models. The results are reported in Table 5 in terms of Normalized Discounted Cumulative Gain (NDCG) [47] and Expected Reciprocal Rank (ERR) [48]. Based on the results, we selected LamdaMART to report our results for the rest of the experiments to report the results of our explanation model.

Table 5 NDCG and ERR on test data reported by various models.

<table>
<thead>
<tr>
<th>Model</th>
<th>NDCG@5</th>
<th>ERR@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coordinate Ascent</td>
<td>0.945</td>
<td>0.809</td>
</tr>
<tr>
<td>MART</td>
<td>0.936</td>
<td>0.797</td>
</tr>
<tr>
<td>LamdaMART</td>
<td>0.953</td>
<td>0.823</td>
</tr>
</tbody>
</table>

4.3 List Recommender System Baselines

The focus of our work is to provide post-hoc explanations for Lists recommended to a user by a recommender system. Therefore, we are able to work with any List recommender systems. Without loss of generality, to show the model agnostic nature of our explanation model, we applied two collaborative filtering (CF) based recommender systems and two content-based recommender systems to generate List recommendations for each user. It is
noted that, since our main goal is not to propose a List recommender system, four simple recommender algorithms are used in our experiments which are explained in the following.

**CF-based recommender systems:** A user that subscribes to a List probably likes the topic of the List. However, if she doesn’t subscribe, we cannot interpret it as dislike. Therefore, since relation between users and Lists is a type of implicit feedback, we choose the following two CF based implicit recommender systems to recommend some Lists to each user. We used the implementation provided in Implicit Library\(^3\) for these approaches.

- **Logistic Matrix Factorization (LMF):** This method takes a probabilistic approach that models the probability of a user’s preference on an item by a logistic function \(^{49}\).
- **Alternating Least Squares (ALS):** This method is a point-wise matrix factorization model that assigns confidence levels to different implicit feedback instances and then factorizes the weighted matrix. It works based on the Alternating Least Squares (ALS) algorithm described in \(^{50}\) with performance optimizations described in \(^{51}\).

**Content-based recommender systems:** To develop a content based List recommender system for Twitter, we need to first model the users’ profiles and Lists’ profiles and then recommend some Lists to each user based on the similarity between their profiles. In our work, each user’s preference profile is built based on the content of the tweets that she has interacted with. Similarly, each List profile is created by the content of tweets that a List contains. More details about how these profiles are constructed based on the tweets are explained in the following two methods which are considered as our content-based recommender systems.

- **TFIDF-based:** In this method, both users and Lists are represented as TF-IDF weighted vectors over terms extracted from their corresponding tweets. Then, we used cosine similarity to measure similarity between the user and List profiles \(^{52}\).
- **Topic-based:** In this method, both users and Lists are represented over the \(K\) topics extracted from our dataset by BERTopic as explained in Section 3.1. Then two matrices are created: user-topic matrix that is represented as \(A = |U \times |Z|\) where \(A_{ij}\) denotes weights of interest user \(u_i\) over topics \(z_j\) and topic-List matrix that defines a relation between the set of Lists and the topics that are spanned by these Lists and represented as \(B = |Z \times |L|\). Thus, the interest of users towards the Lists is a matrix that is obtained from \(H = A \times B\) that \(H\) is \(|U \times |L|\) matrix. Therefore, we can generate recommendations for each user based on weightage \(H[4]\).

To evaluate the accuracy of the List recommender system baselines, similar to \(^{49}\) and \(^{50}\), we adopted Mean Percentage Ranking (MPR), a recall based evaluation metric, that evaluates a user’s satisfaction with an ordered list of

recommendations. For each user $u$ we generated a ranked list of the Twitter Lists in $L$ sorted by preference. Let $rank_{ul}$ denote the percentile ranking of List $l$ for user $u$. $rank_{ul} = 0\%$ signifies that $l$ is predicted as the highest recommended List for $u$. Similarly, $rank_{ul} = 100\%$ signifies that $l$ is predicted as the lowest recommended List for $u$. MPR is defined as follows:

$$MPR = \frac{\sum_{u,l} r_{ul} \times rank_{ul}}{\sum_{u,l} r_{ul}}$$

(3)

where $r_{ul}$ is a binary variable indicating whether a user $u$ subscribes List $l$.

It should be noted that the lower values of MPR are more desirable. We split the data as 10% test and 90% train for CF-based algorithms and we used the set of all data as test set for content-based algorithms. The evaluation result is shown in Table 6. The content-based and CF-based algorithms achieved an MPR about (avg) 17.65% and 24.6%, which is already a good improvement over a purely random model that has an expected MPR of 50%.

### 4.4 Evaluation Methodology and Metrics

In general, there are two approaches for evaluating the explanations generated for recommendations offline: (1) evaluate Fidelity of the explanation model by computing the percentage of recommendations that can be explained by the explanation model regardless of the explanations quality; and (2) evaluate the quality of explanations by user study [7].

For the first approach, adopted from [12], we used Model Fidelity as a measure to evaluate post-hoc explanation model, which is defined as the percentage of explainable Lists in the recommended Lists by a recommender system:

$$\text{Model Fidelity} = \frac{|\text{explainable Lists} \cap \text{recommended Lists}|}{|\text{recommended Lists}|}$$

(4)

where an explainable List $l$ for a user $u$ is defined as a List for which at least a candidate explanation exists among the user’s tweets based on the candidate explanation generation method explained in Section 3.1, i.e., $|C_u(l)| \geq 1$.

For the second approach, adopted from [53], we evaluated the following two aspects of our system: (1) the quality of the generated explanations in terms of their relatedness and informativeness to the recommended Lists by performing a user study. (2) the importance of ranking step by conducting a pairwise evaluation to compare the ranked explanations with candidate explanations. The results of these two aspects are given in section 4.6 and 4.7 respectively.
4.5 Model Fidelity Analysis

As described in Section 3, given the top-$T$ Lists recommended by a recommender system to a user, our explanation model generates an explanation for each recommended List. In this section, we evaluate our proposed explanation model in terms of Model Fidelity based on Equation 4 when the recommendations are generated by different recommendation baselines introduced in Section 4.3. The results are reported in Table 7 for different values of $T \in 1, 5, 10$. Based on the results, one can observe that the model fidelity of our proposed explanation model is more than 0.96 for all the recommendation baselines and different values of $T$. It shows our explanation model is able to generate explanations for at least 96% of recommended Lists to a user. Therefore, regardless of the type and accuracy of the underlying recommender system, our explanation model is able to provide post-hoc explanation for the majority of recommended Lists.

As another observation, by comparing the results of different recommender system baselines together, it can be observed that the fidelity of our explanation model is higher for the content-based recommender systems (i.e., TFIDF-based and Topic-based) compared to the CF-based methods (i.e., ALS and LMF). The content-based recommender systems generate recommendations based on the textual content of users and Lists on Twitter which is more aligned with the idea of our approach to select candidate explanations compared to the CF-based recommender systems. In other words, since the CF-based recommender system may recommend some Lists to a user which are not topically related to the tweets of the user herself (e.g., it may be related to the interests of the user’s friends), there is more possibility for our explanation model not to be able to generate explanations for them.

4.6 Evaluate the quality of explanations by user study

In the previous section, we evaluated the results of our proposed model in terms of Model Fidelity which only evaluates the ability of the model to generate explanations for the recommended Lists to a user and overlook the quality of the explanations. Therefore, in this section, we have conducted a user study to evaluate the quality of the generated post-hoc explanations for each
List recommended to a user (i.e., (user, List, explanation) triples). Since our explanation model is a tweet-based explanation generation model, we evaluate explanations from two different perspectives for each explanation, relatedness and informativeness.

To do so, we have first randomly selected 100 (user, List, explanation) triples generated by our explanation model for the Lists recommended by a recommender system. Then, we asked two human annotators who are active Twitter users to annotate each triple by one of the following three points (The links to the user profile and List on Twitter are provided for the annotators to get extra information):

0 explanation is not related to the List (unrelated)
1 explanation is related to the List but is not informative (related and non-informative)
2 explanation is related to the List and is informative (related and informative).

To resolve the annotation conflicts, for each (user, List, explanation) triple, similar to [40], if the difference between the points assigned by the two annotators is equal to 1, the lower point is selected as the final label. Otherwise, we asked the annotators to re-annotate those triples after discussion on disparities.

The results of user study for our explanation model for each recommender system baseline are reported in Figure 2 which shows the number of (user, List, explanation) triples annotated by each label (i.e., unrelated, related and non-informative, related and informative). Based on the results, for all the recommender system baselines, the number of related and informative explanations are more than other types of explanations. On average for all four recommender systems, 70.25% are related and informative, 13% of generated explanations are unrelated and 16% are related and non-informative. It illustrates the model agnostic nature of our explanation model which is able to generate related and informative explanations for the majority of samples regardless of the underlying recommender system.

Further, similar to our observation in terms of model Fidelity, by comparing the results of different recommender systems together, one can see that our model works better on providing explanations for the Lists recommended by the content-based recommender systems. Therefore, although our approach is model agnostic and is able to generate explanations for different types of recommender systems, since it is selecting the tweets as explanations based on the topical similarity, it is more effective when the Lists are also recommended based on the similar idea.

In Table 4.6, two sample explanations for each annotation category are reported. For example, for a given user, the explanation provided by our system for the recommended List "scientists who do climate" is labeled as "unrelated" by our evaluators. After digging into our data, we found that the reason why our explanation model couldn’t provide related explanation to this recommended List is that, the members of this List has published many tweets
topically unrelated to the main topics of the List (e.g., since COVID-19 pandemic was a trending topic in dataset creation time, members of this List have also published some tweets about the pandemic); thus the topic modeling method has detected COVID-19 as one of the dominant topics of the List and as a result selected one of the user’s tweets which is related to COVID-19. To alleviate this issue, we intend to filter some off-topic tweets from the Lists and then extract their dominant topics. As another example, the explanation provided for the recommended list “Anti-Racism in Medicine” to a user is also rated as “unrelated”. As it is shown in Table 4.6, the content of explanation for this List is about a black man that is relevant to Anti-Racism, but since it is not related to Medicine, it is considered as unrelated by the annotators. The reason why our model couldn’t generate related explanation in this example is that the user have not published a fully related tweet to the List herself and the potential reason that this List is recommended to the use is probably her friends activities. In Table 4.6, we have also reported two sample explanations for the ”related and non-informative” and ”related and informative” labels. ”related and informative” explanations include appropriate related information to the recommended List. However, ”related and non-informative” explanations are semantically related to the main topic of the recommended List but its content cannot provide useful information. For example, ”Everyone is going to want #Bitcoin for Christmas” and ”The world doesn’t need to collapse for bitcoin to be successful. In fact, it’s likely bitcoin will serve as a release valve for the global economy. We should be extremely thankful and hopeful for its existence and survival. #Bitcoin is hope.” are both related explanations for “Bitcoin” List. However, the latter is include more information about bitcoin and labeled as informative as well.
Table 8 Examples for explanation annotations categories.

<table>
<thead>
<tr>
<th>Human annotation</th>
<th>List name</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>unrelated</td>
<td>scientists who do climate</td>
<td>&quot;One of my 90 year old homebound dementia patients just got hospitalized with COVID. (Full code per husband) Last visit I showed aide and husband that the CDC website does NOT have lots of reported cases of death by vaccine. I am so sick of this shit. #COVIDIOTS&quot;</td>
</tr>
<tr>
<td>related and non-informative</td>
<td>Anti-Racism in Medicine</td>
<td>&quot;I just went to grab lunch in the café wearing my fight suit and the Black man behind the counter started cheering and clapping. He was so excited, had never seen a Black woman wearing one and went on about how proud he is off me. I’m just over here like, how am I so blessed?&quot;</td>
</tr>
<tr>
<td>related and informative</td>
<td>Covid</td>
<td>&quot;I don’t know about you guys but I love wearing masks.&quot;</td>
</tr>
<tr>
<td></td>
<td>Bitcoin</td>
<td>&quot;Everyone is going to want #Bitcoin for Christmas.&quot;</td>
</tr>
<tr>
<td></td>
<td>Covid</td>
<td>&quot;More than 99.99% of people fully vaccinated against Covid-19 have NOT had a breakthrough case resulting in hospitalization or death. I’ll take those odds! #GetVaccinated&quot;</td>
</tr>
<tr>
<td></td>
<td>Bitcoin</td>
<td>&quot;The world doesn’t need to collapse for bitcoin to be successful. In fact, it’s likely bitcoin will serve as a release valve for the global economy. We should be extremely thankful and hopeful for its existence and survival. #Bitcoin is hope.&quot;</td>
</tr>
</tbody>
</table>

4.7 Analysis on the Explanation Ranking component

As explained in Section 3, our explanation model includes two main components, candidate explanation generation and explanation ranking. The explanation ranking component re-ranks the retrieved candidate explanations for each (user, List) pair based on their relatedness to the recommended List and their informativeness. The boxplots in Figure 3 depicts the number of retrieved candidates by each of our recommender system baselines. Based on the results, the number of retrieved candidate explanations can vary from a few ten to even hundreds. Therefore, it is imperative to have a ranking component to provide the most informative and related candidates as a final explanation.

In Section 4.6, we evaluated the quality of our explanation model including both components. In this section, we study the contribution of the explanation ranking component on the quality of selected explanations. To do so, we design a pairwise evaluation to compare the quality of top-1 ranked explanations (i.e., the final output of our explanation model) compared to the top-1 explanation selected by the candidate explanation generation component before applying the ranking component. Given the top-1 ranked explanation, denoted by $A$ and
The number of candidate explanations with different recommender approaches.

top-1 candidate explanation, denoted by B, the annotators have annotated pairwise evaluation between two explanations with one of the following point:

1) A is more related and informative than B.
2) B is more related and informative than A.
3) A and B are almost the same, both related and equally informative.
4) A and B are almost the same, both unrelated.

Pairwise evaluation results are shown in Figure 4. On average for all four recommender systems, for 49.5% of recommended List, top-1 ranked explanations are more related and informative than the top-1 candidate ones. In addition, for 35.75% of recommended List, the annotators find that both explanations are useful and it is hard to judge which one is better. Therefore we can conclude that in most cases, top-1 ranked explanations achieves equal or even better performance than candidate ones which shows the positive contribution of the explanation ranking component to improve the quality of explanations.

To zoom in on the relative effectiveness of each of the feature groups (i.e., content-relevance, post-specific and publisher’s authority) used in the explanation ranking component explained in Section 3.2, we have done an ablation study by removing the features of each group at a time. The results of the ablation study on the performance of LambdaMART are presented in Table 9. Based on the results, all three categories of the features are effective on the performance of the explanation ranking model and by removing them the performance is decreased in terms of both metrics (i.e., NDCG@5 and ERR@5). However, among different category of features, it is shown that content relevance features and post specific features play more important role on the performance of the ranking component.
Table 9 Ablation study results. Statistically significant differences (paired t-test, $p < 0.05$) from the first row are marked by *.

<table>
<thead>
<tr>
<th></th>
<th>NDCG@5</th>
<th>▼</th>
<th>ERR@5</th>
<th>▼</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>0.953</td>
<td>0.823</td>
<td></td>
<td></td>
</tr>
<tr>
<td>- Content Relevance</td>
<td>0.858</td>
<td>9.97% *</td>
<td>0.753</td>
<td>8.51% *</td>
</tr>
<tr>
<td>- Post Specific</td>
<td>0.838</td>
<td>12.07% *</td>
<td>0.736</td>
<td>10.57% *</td>
</tr>
<tr>
<td>- Publishers’ Authority</td>
<td>0.951</td>
<td>0.21%</td>
<td>0.821</td>
<td>0.24%</td>
</tr>
</tbody>
</table>

4.8 Analysis on Recommendations without Explanations

The proposed explanation model is based on the user’s actions that are semantically related to the main subject of the recommended List. Therefore, as it is shown in Section 4.5 in terms of Model Fidelity, for some Lists recommended to a user by a recommender system, our explanation model is not able to provide explanations. In this section we propose to provide non-personalized explanations for such cases based on the dominant topics of the recommended List. In detail, as BERTopic creates easily interpretable topics whilst keeping important words in the topic descriptions using c-TF-IDF, we have the top-10 most representative words of each topic, based on their scores in c-TF-IDF. The main idea is to provide a set of representative words for each List based on its dominant topics and present it as a non-personalized explanation. Algorithm 1 shows how to create a set of words for a List by having its dominant topics. Then similar to [54] and [55], we presented an explanation as a word-cloud that
Post-hoc Explanation for Twitter List Recommendation

Fig. 5 non-personalized explanation on word-cloud for recommended Lists

the word size reflects the weight of each word in List representation. Figure 5 shows two examples that the target user can be informed about the recommended List based on its representative words on the word-cloud. For example, the word-cloud of List “NHL” shows the words such as hockey, olympic, skating, medal, winter and league that can describe the List and inform about its content. Therefore, the target user can obtain some information about the recommended List easily by observing this word-cloud.

Algorithm 1 non-personalized explanation for the recommended List

\begin{algorithm}
\caption{non-personalized explanation for the recommended List}
\begin{algorithmic}[1]
\State \textbf{Input} a recommended List $l$, a set of dominant topics $Z_l$ of List $l$, a topic model $tm$
\State \textbf{Output} a set of representative words for List $l$ as explanation

\State List_words = EmptyDictionary
\For {tp in $Z_l$}
\State tp_words = $tm$.get_topic(tp)
\For {word, score in tp_words}
\State List_words[word] = List_words.get(key= word,0) + score
\EndFor
\EndFor
\State \textbf{return} List_words
\end{algorithmic}
\end{algorithm}

5 Conclusion and Future Work

In this paper, we studied post-hoc explainability in List recommendation on Twitter. We proposed an explanation model that focuses on the user’s own action to provide explanations for users to make more informed decisions on social media. More specifically, given a (user, recommended List) pair, our model first retrieves the user’s tweets which are topically similar to the recommended List as candidate post-hoc explanations and then rank the candidates based on their relatedness and informativeness. In our experiments, we first evaluated the explanation model in terms of model Fidelity and then conducted a user study to evaluate the quality of generated explanations. We also investigated the importance of different components in our explanation model.
and showed the ranking component helps selecting more related and informative tweets as explanations. As future work, we intend to extract more features from the candidate tweets and utilize them in the explanation ranking process such as the recency of microposts, sentiment of tweets and their POS tags. In current work, we select only one tweet that is semantically related to the recommended List as a final post-hoc explanation. We intend to examine how many tweets could be more accurate indicators of persuasiveness explanation. As another future work, we are going to investigate the effect of the different topic modeling approaches on the performance of the proposed explanation model.

References


Post-hoc Explanation for Twitter List Recommendation


Post-hoc Explanation for Twitter List Recommendation


[46] Lu, C., Lam, W., Zhang, Y.: Twitter user modeling and tweets recommendation based on wikipedia concept graph. In: Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence (2012)


Post-hoc Explanation for Twitter List Recommendation
