

A Link Prediction Strategy for Personalized Tweet Recommendation through Doc2Vec Approach

Mojtaba Zahedi Amiri^{1*} & Abdullah Shobi¹

¹ Department of Information Technology, Computer Science College, Mazandaran University of Science and Technology, Babol, Iran

* Mojtaba Zahedi Amiri, E-mail: mojtaba.zahedi.amiri@gmail.com

Received: June 16, 2017

Accepted: July 8, 2017

Online Published: July 13, 2017

doi:10.22158/rem.v2n4p63

URL: <http://dx.doi.org/10.22158/rem.v2n4p63>

Abstract

Nowadays with growth of using Internet as a principle way of communication, likes different social medias channels (Twitter, Facebook, etc.) and also access to huge amount of information like News, there appear a main research subject to help users to find his/her interests among vast amount of relevant and irrelevant information. Recommender systems are helped to handle information overload problem and in this paper we introduce our Tweet Recommendation System that implement user's Twitter information (Tweets, Retweet, Like,...) as a source of user's information. In this work the semantic of tweets that regard as a User's Explicit Interests (e.g., person, events, product mentioned in user's tweets) are identified with the Doc2vec approach and recommend similar tweets through link-prediction strategy. The experiment results show that Doc2Vec approach is a better approach than the other previous approaches.

Keywords

personalization, Doc2Vec, semantic relatedness, link-predictio

1. Introduction

Every day number of users, using social media have been increased and they create huge amount of data about their interests, everyday events, their friends and..., which are so precious as a source of raw data to explore knowledge about them. The available data on these social networks is of great importance when mined and used for such purposes as analysis and prediction. Nowadays recommender systems are help users to find their own interest between this vast of information and items and also solve information overload. Recommender systems are a means of personalization providing their users with personalized recommendations of items that would possibly suit the users' needs. The main purpose of a recommendation system is to estimate the user's preferences and present him with some items that he doesn't know yet.

In general recommender systems are programs which attempt to predict items that users may be interested in.

Recommender systems are worked almost in a same way through different domains, which by using users historical interests or ratings, predict the items that user might like them. But in some specific domains like News, the story is different.

News are so timed depended and after short period of time their freshness are gone. So the news recommender system should be able to recommend fresh news as well as related to user interests.

People were followed news from different sources like old traditional way but nowadays the most common way that people read news are social networks channels that provide with people most recent and fresh news. One of the most famous social networks that focused in news is Twitter. Every user has profile in twitter and will follow different Channels, Celebrities or his/her friends. Since every user has many following pages, his/her twitter timeline has many relevant or irrelevant subjects which forced users to finding his interested news.

As a consequence, the role of user modeling and personalized information access is becoming crucial: users need a personalized support in sifting through large amounts of available information, according to their interests and tastes.

For this reason in this paper we study different users tweets for modeling users and introduce a framework for our tweet recommender system that semantically enriched each user tweets and detecting his/her interest and recommend to users some fresh and relevant news.

In this propose approach and in the first step we find user's explicit interests, User's profile build with implementing Doc2vec method on user's tweets After build each user's explicit profile based on Doc2vec models, similar user's are fined with similarity of theirs vectors and also we can find each similar semantic tweets.

2. Literature Review

With the growing impact of the Social Web, or Web 2.0, on our everyday life, people start to use more and more different web based services like Facebook, Twitter, Flickr or blogs. They use these services to express their opinion, communicate with others and share pictures with friends. Thereby, they generate and distribute personal and social information like interests, social contacts, preferences and personal goals.

Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but those who are unregistered can only read them. Users access Twitter through the website interface, SMS or mobile device app.

As mention before, Twitter is one of the famous Content-Centric social network, which enables users to send and read short 140-character messages called "tweets". Due to the extensive usage of twitter, a large volume of text is being generated on a daily activity of users. Such a huge volume of user generated data had to be processed to utilized them effectively.

These data could be used in a variety of applications to enhance human life. For processing such huge amount of textual data, more advanced algorithms are required to learn the hidden patterns in the data. Text analytics is the method to process this huge corpus of unstructured text to get high quality data. Text Analytics is defined in Wikipedia as follows:

“Text Analytics describes a set of linguistic, statistical, and machine learning techniques that model and structure the information content of textual sources for business intelligence, exploratory data analysis, research, or investigation”.

A text analytics framework consists of three stages: Text preprocessing, Text representation and Knowledge discovery.

In text preprocessing, textual data that are produced by social media sites could not be analyzed directly because these are raw input texts. Preprocessing makes the text more consistent to facilitate the text representation. For instance removing non-English words, transform all words to lower case, removing links and, etc.

After preprocessing the input text, only significant words are present in the text. These words need to be represented as numeric vectors to make the analyzing easier. Vector space model or term vector model is an algebraic model for representing text documents or any other objects as a vectors of identifiers. There are different methods in text representing such as Bag-Of-Word (BOW), TF-IDF and Paragraph Vector that we used in our approach, called Doc2vec.

The Word2vec and Doc2vec model and application by Mikolov et al. have attracted a great amount of attention in recent two years. The vector representations of words learned by Word2vec and Doc2vec models have been shown to carry semantic meanings and are useful in various NLP tasks. As mentioned before we used Doc2vec model to represent each tweets.

Paragraph Vector is an unsupervised framework that learns continuous distributed vector representations for pieces of texts. The texts can be of variable-length, ranging from sentences to documents. The name Paragraph Vector is to emphasize the fact that the method can be applied to variable-length pieces of texts, anything from a phrase or sentence to a large document.

In Bag of words approach, the text is divided into words. This process is called as tokenization. The structure of the text is not maintained in this approach. Each word is represented as one single variable with different numeric weights. TF-IDF (Term Frequency/Inverse Document frequency) is commonly used as the weighing mechanism. In string of words approach, sequence of the words is maintained. In most applications, Bag of words is used due to its simplicity.

Once the textual data are transformed into numeric vectors, machine learning or data mining algorithms could be used to identify hidden patterns in the text. The most common approaches followed are classification and clustering. Clustering fall under the category of unsupervised learning and classification falls under the category of supervised learning. In unsupervised learning, training data are not required. The documents which contain the textual data are segmented into different partitions such that each partition belongs to a single topic. This process is termed as clustering. In supervised learning,

training data are required to make a machine learning method to learn a classifier to classify unseen data. Classification is used in various applications like news filtering, document organization and retrieval, opinion mining, email classification and spam filtering.

3. Related Work

One of the most viewed social network channel is Twitter. Twitter pose a question to its users “What is happening?” and user can answer to this question in 140 characters.

In twitter user have different opportunity to demonstrate their mine like: post tweet or update their following post or re-tweet them. Users can also using different Tags to show their feeling.

Although tweets may contain precious information, many of them have no relatedness to the users. This can annoy users to find their own interests between big amount of information. To this end, different work have been accomplished to response to this challenge.

In the propose approach classified web pages by calculating the respective weights of terms. The user interest and preference models are generated by analyzing the user’s navigational history. The similarity between Web content and the user’s models is used to determine whether the content will be provided to the user. A user’s navigational data is monitored and analyzed to conduct user modeling. An automatic classification method is utilized to categorize the Web contents browsed by a user.

In the proposed Web page classification method, the terms are determined by the ontology base WordNet (Miller, 2009), and the weights of terms are calculated by the TF-IDF (term frequency—inverse document frequency) method.

Some others researcher work on the hybrid approach of recommendation, like that propose a new methodology for recommending interesting news to users by exploiting the information in their twitter persona which model relevance between users and news articles using a mix of signals drawn from the news stream and from twitter: Profile of social neighborhood of the user, Content of their own tweet stream, Topic popularity in news and in the whole twitter-land.

In the main focus is on the dynamic recommendation system that mentioned to have a successful recommendation system for active users, we should introduce “somewhat novel” articles to users. In this work by combining long-term interest of user with short-time interest, recommending a novel news to users.

The inspiring research is which present a content-based approach to modeling user interests based on Twitter. Personalization techniques are often classified into one of two categories: explicit and implicit. Explicit personalization requires active and conscious data entry from the user, such as through a series of checkboxes or rating devices. In contrast, implicit personalization aims to automatically learn user preferences. Content-based approaches typically monitor the behavior of a user in the scope of an individual site or system and make recommendations based on their historical behavior.

In general, implicit personalization is considered more desirable from a user experience perspective because it does not burden users with data input tasks.

In the other hand some other researcher work on the semantic of the tweets created by the users. Semantic relatedness, which computes the association degree of two objects such as words, entities and texts, is fundamental for many applications. It has long been thought that when human measure the relatedness between a pair of words, a deeper reasoning is triggered to compare the concepts behind the words.

In investigate this question and introduce a framework for user modeling on Twitter which enriches the semantics of Twitter messages (tweets) and identifies topics and entities (e.g., persons, events, products) mentioned in tweets.

In other work investigate semantic user modeling based on Twitter posts which introduce and analyze methods for linking Twitter posts with related news articles in order to contextualize Twitter activities. While many semantic relatedness researches in the past utilized lexical databases such as Word Net and Wikitionary, the recent word embedding approaches have demonstrated their abilities to capture both syntactic and semantic information. In mentioned that among the embedding representations, Word2Vec and GloVe are widely adopted for many researches. However, word senses are not disambiguated in the training phase of both Word2Vec and GloVe. That affects the measurement of semantic relatedness. On the other way round, Word Net and Wikitionary are well-structured ontology that provides senses of each word. Their approach was to combined Word2Vec and GloVe with the lexical database Word Net is proposed for measuring semantic relatedness. Demonstrate that by converting words and phrases into a vector representation, word2vec takes an entirely new approach on text classification. Based on the assumption that word2vec brings extra semantic features that helps in text classification, our work demonstrates the effectiveness of word2vec by showing that tf-idf and word2vec combined can outperform tf-idf because word2vec provides complementary features (e.g., semantics that tf-idf can't capture) to tf-idf.

4. Methods

In Figure 1 architecture of our proposed approach is showed and we describe how this system worked together. Then we describe each part of this respectively.

The Architectural design for the proposed approach consists of the following stages:

- Content Gathering:
 - Use Twitter API.
- User modeling:
 - Tweet Preprocessing,
 - Tweet Representation,
 - Building Explicit user's profile,
 - Building User Explicit Graph,
- Recommendation Task.

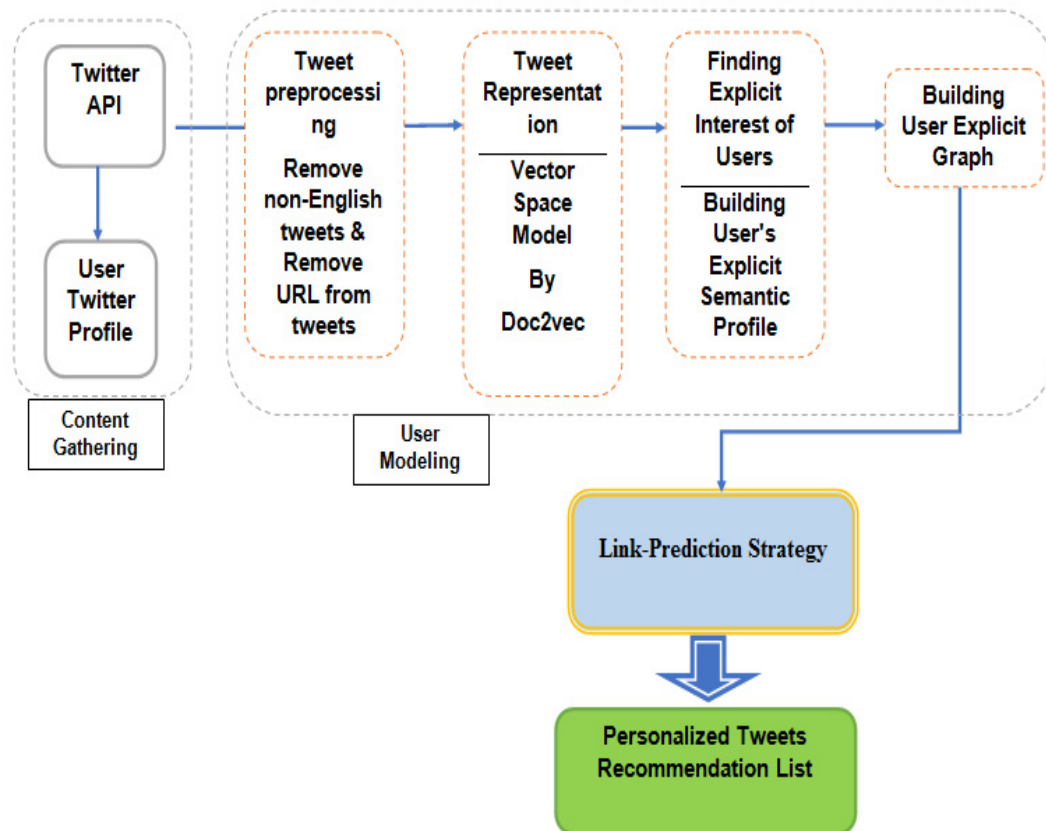


Figure 1. Architecture of Proposed Approach

In Content Gathering step, Twitter provides many REST APIs to acquire data from Twitter using screen name. Using the “user_timeline” API, tweets from a particular user is acquired. This API takes many parameters as input. Screen name and number of tweets are two important parameters required to acquire a certain number of tweets from a particular user. Using this API, tweets for all the users in all the categories are acquired and stored in a file. Tweets, which are collected in the previous step cannot be used to train the classifier directly. Because these textual data contain some unwanted text. In preprocessing, these unwanted symbols and meaningless words are removed from the original tweet. Most Tweets contain URLs, links, special symbols, abbreviations, hashtags, mentions and incorrect spellings. The following rules have been followed to preprocess the tweets:

Rule 1: Remove all special characters except “#” and “@”;

Tweets express emotions. So people use special characters to express their emotions. So all these special characters are replaced with null characters. “Hashtags” are the keywords in the tweets followed by the “#” symbol (e.g., #SuperBowl). Many users would be using the same hashtag for a particular event. So these hashtags are retained in the tweets. @ Symbol is used to specify the username. This is being handled by Rule 2.

Rule 2: Remove all URLs and @mentions;

Shortened URLs are used in tweets. These URLs do not provide much information for us. For example, consider this shortened URL “bit.ly/12Jkw6U”. These URL strings do not contain much text to predict the category. So these shortened URLs are removed from the text during preprocessing. “@” symbol is used to specify a screen name of a user in the tweets (e.g., @BarackObama). The words prefixed with “@” symbol is called as “mentions”. These words cannot be used to predict the category of the tweet. Because these words usually contain only user name.

Rule 3: Convert all words to lower case;

Tweets are written in an inconsistent format. All the characters in the tweets can be either capital or small or mixed. To make the training data more efficient, all the words in the tweets are converted to lower cases. In the Representation step, the vector representations of words learned by Word2vec and Doc2vec models have been shown to carry semantic meanings and are useful in various NLP tasks. As mentioned before we used Doc2vec model to represent each tweets.

Paragraph Vector is an unsupervised framework that learns continuous distributed vector representations for pieces of texts. The texts can be of variable-length, ranging from sentences to documents. The name Paragraph Vector is to emphasize the fact that the method can be applied to variable-length pieces of texts, anything from a phrase or sentence to a large document.

Paragraph2Vec, which can be called in many names such as Doc2vec, paragraph vector or sentence embedding, is the algorithm that was modified from Word2Vec. The main purpose of Doc2Vec is associating arbitrary documents with labels, so labels are required. Doc2vec is an extension of word2vec that learns to correlate labels and words, rather than words with other words. In Figure 2, the abstract of Doc2vec is demonstrated.

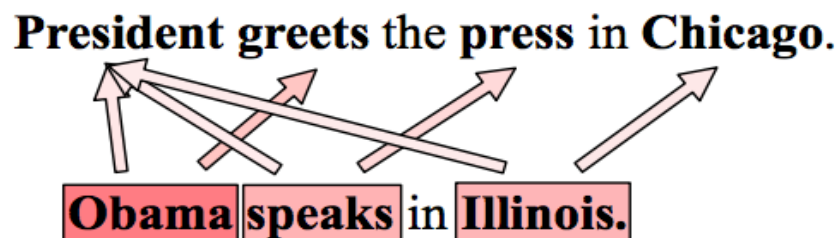


Figure 2. Doc2Vec

After preprocessing and represent the tweet as vector, we can find explicit interest of user's through the tweet's he/she were post or retweet or like. User's have their own tweets and similarity of user's can find through Doc2vec similarity function, which Doc2vec similarity function is cosine-similarity by default. Cosine-similarity computes each user's vector with each other and find most-similar of each user.

5. Experiment

We use a graph based link prediction to build user explicit graph and infer explicit interest of user's to recommend similar tweets. Link prediction is the problem of predicting the presence or absence of edges between nodes of a graph. There are two types of link prediction: (i) structural, where the input is a partially observed graph, and we wish to predict the status of edges for unobserved pairs of nodes, and (ii) temporal, where we have a sequence of fully observed graphs at various time steps as input, and our goal is to predict the graph state at the next time step.

The underlying graph of our proposed approach use three type of information: user's relationship with each other, user's relationship with tweets and tweet's relationship with each other.

Base on our underlying representation for user model can be formalized as follow:

The representation model $G=(G_U, G_T, G_{UT})$ is a heterogeneous graph composed of three sub graphs, G_U , G_T and G_{UT} . $G_U=(V_U, E_U)$ is unweighted and undirected, which represents similarity between users base on the Doc2vec model. $G_T=(V_T, E_T)$ denotes potential similarity between all tweets base on the Doc2vec model. $G_{U,T}=(V_{U,T}, E_{U,T})$ represents similarity between each user and other tweets base on the Doc2vec model.

The proposed approach in this paper is implemented in Python and run under Windows platform. We perform our experiment to answer this question: Is the hybrid approach has a better performance than implementing the explicit approach and implicit approach independently.

6. Dataset

In this proposed approach we collected our tweets dataset from Twitter using Twitter's API. The first step to using Twitter's API is to be authenticated by Twitter. After registered the application in Twitter, the following parameters are provided to access the Twitter to collect tweets: Consumer token, Consumer secret key, access token and access secret key.

We used this parameters to get tweets through using Python Twitter library called tweepy. We collect our data set from timelines of followers of most visited pages in Twitter (such as bbctech, bloomberg, espn, fl, microsoft, newyork times, washington post, cnni, euronews and, etc.). For each channel we collect the follower of that channel user's id and take that user's id to extracted his/her timelines tweets. We crawled over 900 user's and 160,000 tweets. Twitter constrains that for each user, we can only crawl his/her last 3200 tweets. However this is sufficient for our experiment.

To generate the dataset, we use Sqlite as our data warehouse. All extracted tweets are went to the preprocessing step, which remove all URL, Non-English words, remove all @(mention) and transform all words to lower case. In the next step, we illustrate how to implemented Doc2Vec on tweets and user's profile.

After preprocessing step, to represent each tweets as a vector, we implemented Doc2Vec model on tweets dataset. We build a JSON file which contain all tweets and in each line the file look like this ("Tweet-Id", "Tweet-text"). Doc2Vec model was trained in this data set which the label was tweet-id.

7. Tweet-Tweet Graph

In our proposed approach the Doc2Vec features are as follow, size of 35 and windows of 5. The result is similar semantic tweets with their tweet-id and similarity degree. This tweet's Doc2vec model is used to build *Tweet-Tweet graph* as mentioned in recommending task. We collect 30 most similar tweets for each tweet, to build the link between tweets in graph.

8. User-User Graph

After training all tweets through Doc2Vec model, to find similarity between users we should trained all user. To evaluate our proposed approach, we collected 100 active user, whom at least posted 400 tweets. To predict our proposed model accuracy, we implement Cross-Validation. Cross validation is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of *known data* on which training is run (*training dataset*), and a dataset of *unknown data* (or *first seen data*) against which the model is tested (*testing dataset*). Cross validation has different types but in our approach we implement Leave-p-cross-validation. Through used Cross-validation, remove randomly 30% of tweets from all collected user's tweets, which they were liked or retweet by user to test our approach and the other user's tweets are used to trained our user's explicit model.

The Doc2Vec model build on this 100 users. The user's Doc2Vec model input is a JSON file, that the format look like this ("User-Id", "All user's tweet"). Doc2Vec model was trained in this data set which the label was user-id.

The result is similar semantic users with their user-id and similarity degree. This user's Doc2vec model is used to build *User-User graph* as mentioned in recommending task. We collect 30 most similar users for each user, to build the link between users in graph.

9. User-Tweet Graph

After building *user-user* and *tweet-tweet* graphs, it's time to build *user-tweet* graph. To this end, we model all tweets and collected user with each other through Doc2Vec. The result is similar semantic tweets for each user. We collect 30 most similar tweets for each user, to build the link between user and tweets in graph.

10. Explicit Recommendation

To build explicit profile of user we should build $G=(G_U, G_T, G_{UT})$ which contains of *user-user* graph, *tweet-tweet* graph and *user-tweet* graph. After build the G graph, the recommendation will build based on the link-prediction strategy. Our problem is to infer whether a user u is explicitly interested in tweet t . In other words, we are going to find missing links by adopting an unsupervised link prediction strategy over links in G Most of unsupervised link prediction strategies either generate scores based on vertex neighborhoods or path information. Vertex neighborhood methods are based on the idea that two

vertices are more likely to have a link if they have many neighbors in common. Path-based approaches consider all paths between two vertices. All these approaches are based on a predictive score function for ranking links that are likely to occur. There is no single superior approach and the structure of the specific graph indicates their quality. In our approach we used Jaccard's Coefficient strategy for inferring explicit interests of a user.

The Jaccard's Coefficient is defined as follows:

$$\text{score}(x,y) = \frac{|\Gamma(x) \cap \Gamma(y)|}{|\Gamma(x) \cup \Gamma(y)|}$$

The explicit profile of a user is a link between tweet t and user u , which the link is computed through the link-prediction approach (L):

If $U = \{u_1, u_2, \dots, u_n\}$, $u_i = \{\text{tweet1}_{ui}, \text{tweet2}_{ui}, \dots\}$ and $T = \{t_1, t_2, \dots, t_n\}$:

The explicit profile of a user is: $E(u) = \{u, L(u,t)/t \in T, u \in U\}$.

11. Evaluation and Metrics

To handle information overload and help users find items based on their interests, some kind of personalization techniques are used in personalized recommender systems. To figure out how much recommended items are suited and relevant to users, we should test and evaluate our proposed recommender system.

The main question is: is a recommender system efficient with respect to specific criteria like accuracy, user satisfaction, response time, serendipity or in some other domain, do customers like/buy our recommended items?

Three typical measures used for evaluating the performance of recommender systems are Precision, Recall, and F-measure. In information retrieval contexts, precision and recall are defined in terms of sets of retrieved documents and a set of relevant documents.

For classification and recommendation tasks, the terms true positives, true negatives, false positives, and false negatives compare the results of the classifier or recommender under test. These four outcomes can be defined in a Contingency matrix as follows in Figure 3:



		Proposed by recommender: 	
		Yes	No
Liked by user: 	Yes	Correct predictions	False negatives
	No	False positives	Correct omissions

Figure 3. Contingency Matrix

The three performance measures are defined in Equation 1, 2 and 3:

$$(1) \text{ precision}(p) = \frac{TP}{TP + FP}$$

$$(2) \text{ recall}(R) = \frac{TP}{TP + FN}$$

$$(3) f - \text{measure}(F1) = 2 * \frac{P * R}{P + R}$$

In our proposed approach, we evaluate the results through these three measurements and compare each model based on Precision, Recall and F-measure. But the point is that in most situations, the system outputs a ranked list of recommendations rather than an unordered set. To this end, in modern information retrieval, precision and recall are not longer a meaningful metric, as many queries have thousands of relevant documents. Precision at k documents (P@K) and Recall at k documents (R@K) are meaningful and useful metrics (e.g., p@10 corresponds to the number of relevant results on the first search results page).

P@K proportion of top-k documents that are relevant and R@K proportion of relevant documents that are in top-k. If we don't know what value of K to chose, we can compute and report several: {5, 10, 15, 20, 15, 30}.

In our evaluation we test our recommendations in different K and find out that in K=30, the recommendation results are better based on precision and recall @K=30. The results of recommendation @k shown in Table 1.

Table 1. Recommendation Evaluation

Link-Prediction	Precision	Recall	F1
K=5	25.6%	9.3%	13.6%
K=10	29.1%	15.3%	20%
K=15	32.7%	28%	30.1%
K=20	38%	39.3%	38.6%
K=25	46%	51%	48.3%
K=30	53 %	56 %	54.4%
K=35	48%	54%	50.8%

The evaluation results show that in k=30 the proposed personalized recommendation system has a better performance. Detail of three recommendation evaluation metrics diagrams are show in Figures 4, 5 and 6.

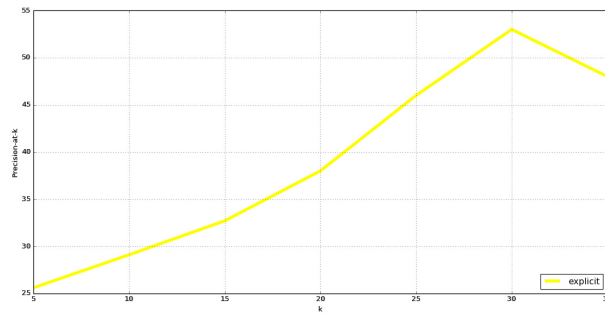


Figure 4. Precision@k

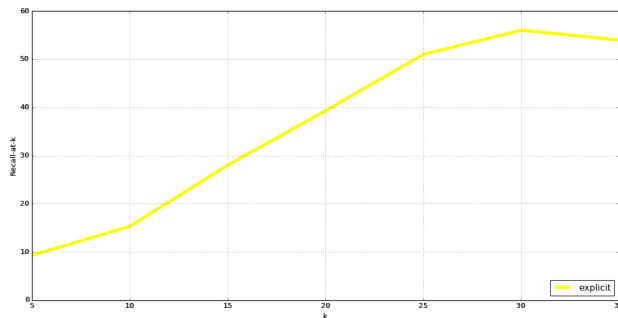


Figure 5. Recall@k

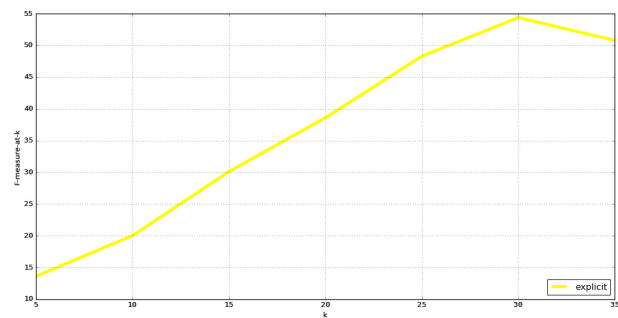


Figure 6. F1@k

12. Conclusion and Future Work

To response the information overloading through internet and help users to find their interested items among these crowd of relevant/irrelevant information, recommender systems have appeared. In this paper, introduce history of recommender systems and different kind of them, like Content-Based recommender system, Collaborative recommender system and Hybrid recommender system. Also, in social media domain, focus on Twitter, which in this research is the main source of information for our proposed personalized recommender system. The next domain, is Text analysis which in text processing step and in representation we used Doc2Vec model. All tweets represent as an words vector. For represent tweets as a vector, in preprocessing step, all links and non-useful symbols and non-English tweets are removed and also convert all words to lower case. We recommend similar

tweets for each user's based on his/her interests through Link-Prediction strategy and the result show that in $k=30$ the proposed approach has a better performances. For future work, to regard in this matter that user's interests are changed through passing time, with adding user's short-term interests and build a dynamic personalized recommender system, have a better personalized recommendation for each user.

References

- Abel, F., Gao, Q., Houben, G. J., & Tao, K. (2011). *Semantic enrichment of twitter posts for user profile construction on the social web in Lecture Notes in Computer Science* (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Vol. 6643 LNCS, No. PART 2, pp. 375-389). https://doi.org/10.1007/978-3-642-21064-8_26
- Abel, F., Gao, Q., Houben, G., & Tao, K. (2011). *Analyzing User Modeling on Twitter for Personalized News Recommendations* (pp. 1-12).
- Abel, F., Henze, N., Herder, E., & Krause, D. (2010). *Interweaving public user profiles on the web* (Vol. 6075, pp. 16-27). Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). https://doi.org/10.1007/978-3-642-13470-8_4
- Amichai-Hamburger, Y., & Vinitzky, G. (2010). Social network use and personality. *Comput. Human Behav.*, 26(6), 1289-1295. <https://doi.org/10.1016/j.chb.2010.03.018>
- Banion, S. O., Birnbaum, L., & Hammond, K. (2012). *Social Media-Driven News Personalization* (pp. 45-51).
- Danah, M. B., & Ellison, N. B. (2007). *Social Network Sites: Definition, History, and Scholarship* (Vol. 13, No. 1, pp. 210-230). *Comput. Commun.*
- De Francisci Morales, G., Gionis, A., & Lucchese, C. (2012). *From Chatter to Headlines: Harnessing the Real-time Web for Personalized News Recommendation* (pp. 153-162). Proc. Fifth ACM Int. Conf. Web Search Data Mining, WSDM'12. <https://doi.org/10.1145/2124295.2124315>
- Ferragina, P., & Informatica, D. (2015). *On Analyzing Hashtags in Twitter* (pp. 110-119).
- Hampton, K. N., Goulet, L. S., Marlow, C., & Rainie, L. (2012). *Why most Facebook users get more than they give* (pp. 1-40). Pew Internet.
- Hogan, B. (1996). *Analysing Social Networks Via the Internet* (Vol. 46, No. 10, pp. 141-160). *Soc. Networks*.
- Kaplan, A. M., & Haenlein, M. (2010). Users of the world, unite! *The challenges and opportunities of Social Media*, 53(1), 59-68. *Horiz.* <https://doi.org/10.1016/j.bushor.2009.09.003>
- Khater, S., Elmongui, H. G., & Gra, D. (2014). *Tweets You Like: Personalized Tweets Recommendation based on Dynamic Users Interests* (pp. 1-10).
- Kim, B. M. (2003). *Clustering approach for hybrid recommender system* (pp. 33-38). Proc. IEEE/WIC Int. Conf. Web Intell.
- Kumar, M. (2015). *Automatic Identification of User Interest From Social Media*.

- Le, Q., & Mikolov, T. (2014). *Distributed Representations of Sentences and Documents* (Vol. 32, pp. 1188-1196). Int. Conf. Mach. Learn. - ICML 2014.
- Lee, Y., Ke, H., Huang, H., & Chen, H. (2016). *Combining Word Embedding and Lexical Database for Semantic Relatedness Measurement* (pp. 73-74). <https://doi.org/10.1145/2872518.2889395>
- Li, L., Zheng, L., Yang, F., & Li, T. (2014). *Modeling and broadening temporal user interest in personalized news recommendation* (Vol. 41, No. 7, pp. 3168-3177). Expert Syst. Appl. <https://doi.org/10.1016/j.eswa.2013.11.020>
- Lilleberg, J. (2015). *Support Vector Machines and Word2vec for Text Classification with Semantic Features* (pp. 136-140). <https://doi.org/10.1109/icci-cc.2015.7259377>
- M. Van, S. (2005). *Supporting People In Finding Information: Hybrid Recommender Systems and Goal-Based Structuring*.
- Menon, A., & Elkan, C. (2011). *Link prediction via matrix factorization* (Vol. 6912, pp. 437-452). Ecm1 Pkdd. https://doi.org/10.1007/978-3-642-23783-6_28
- Meyer, B., Bryan, K., Santos, Y., & Kim, B. (2011). *Twitter Reporter: Breaking News Detection and Visualization through the Geo-Tagged Twitter Network* (Vol. 10, pp. 84-89). Cata.
- Murphy, J., Keating, M., & Edgar, J. (2013). *Crowdsourcing in the Cognitive Interviewing Process*.
- Murphy, J., Link, M. W., Childs, J. H., Tesfaye, C. L., Dean, E., Stern, M., ... Harwood, P. (2014). *Social Media in Public Opinion Research: Executive Summary of the Aapor Task Force on Emerging Technologies in Public Opinion Research* (Vol. 78, No. 4, pp. 788-794). Public Opin. Q. <https://doi.org/10.1093/poq/nfu053>
- O'Connor, B., Balasubramanian, R., Routledge, B. R., & Smith, N. A. (2010). *From tweets to polls: Linking text sentiment to public opinion time series* (pp. 122-129). From tweets to polls Link. text Sentim. to public Opin. time Ser.
- Popescul, A., Pennock, D. M., & Lawrence, S. (2001). *Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments* (pp. 437-444). Proc. Seventeenth Conf. Uncertain. Artif. Intell..
- Rong, X. (2014). *Word2vec Parameter Learning Explained* (pp. 1-19).
- Safko, L. (2009). *The Social Media Bible* (p. 840).
- The history of social media*. (n.d.).
- Wen, H., Fang, L., & Guan, L. (2012). A hybrid approach for personalized recommendation of news on the Web. *Expert Syst. Appl*, 39(15), 5806-5814. <https://doi.org/10.1016/j.eswa.2011.11.087>
- Zarrinkalam, F., Fani, H., Bagheri, E., & Kahani, M. (2016). *Inferring implicit topical interests on twitter. Lect. Notes Comput. Sci.* (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) (Vol. 9626, pp. 479-491). https://doi.org/10.1007/978-3-319-30671-1_35