Predicting Which Topics You Will Join in the Future on Social Media

Haoran Huang, Qi Zhang, Jindou Wu, Xuanjing Huang
School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing
Fudan University
Shanghai, P.R.China 200433
{huanghr15,qz,jdwu15,xjhuang}@fudan.edu.cn

ABSTRACT

Every day, social media users send millions of microblogs on every imaginable topics. If we could predict which topics a user will join in the future, it would be easy to determine what topics will become popular and what kinds of users a topic may attract. It also can be of great interest for many applications. In this study, we investigate the problem of predicting whether a user will join a topic based on his posting history. We introduce a novel deep convolutional neural network with external neural memory and attention mechanism to perform this problem. User's posting history and topics were modeled with an external neural memory architecture. The convolutional neural network based matching methods were used to construct the relations between users and topics. Final decisions were made based on these matching results. To train and evaluate the proposed method, we collected a large-scale dataset from Twitter. The experimental results demonstrated that the proposed method could perform significantly better than other methods. Comparing to the state-of-the-art deep neural networks, our approach achieves a relative improvement of 18.2% in F1-score and 28.9% in MAP@10.

CCS CONCEPTS

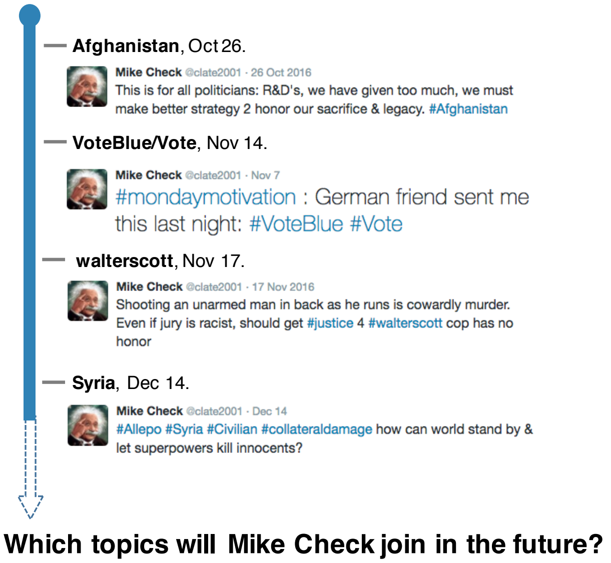
•**Information systems** → **Social recommendation;** *Social tagging;*

KEYWORDS

Topic Prediction, Social Medias, Convolutional Neural Network

1 INTRODUCTION

In recent years, social media sites (e.g., Twitter, Facebook, YouTube) have continuously improved. According to the statistics given by Twitter, more than 313 million monthly active users posted and visited on sites with embedded Tweets[1]. Users talk about every topic imaginable by posting microblogs. To help users quickly find popular topics, both Twitter and Facebook also provide trends (trending lists) for users. To join the discussion of a topic, a user can post a tweet that includes hashtags or phrase as it appears in the topic or retweet the tweet he or she likes that contains this hashtags. Because users produce more content about real-world events almost in real time, topics become accurate sensors of real-world events. Hence, topics provide valuable information for a variety of applications such as stock prediction [3], public health analysis [23, 33] and real-time event detection [24].



Mike Check @clate2001 · 26 Oct 2016
This is for all politicians: R&D's, we have given too much, we must make better strategy 2 honor our sacrifice & legacy. #Afghanistan

Mike Check @clate2001 · Nov 7
#mondaymotivation : German friend sent me this last night: #VoteBlue #Vote

Mike Check @clate2001 · 17 Nov 2016
Shooting an unarmed man in back as he runs is cowardly murder. Even if jury is racist, should get #justice 4 #walterscott cop has no honor

Mike Check @clate2001 · Dec 14
#Allepo #Syria #Civilian #collateraldamage how can world stand by & let superpowers kill innocents?

Figure 1: An Example of This Task in Twitter: A user may join some discussions of topics that interest him. If we can predict which topics he will join in the future, it may provide valuable information for a variety of applications.

Previous researchers have studied the problem of detecting which topics will become popular in advance. In [20], Nikolov and Shah introduced the problem and proposed a nonparametric method to perform the task. Zhao et al. [35] introduced a temporal sequence analysis model to model topic spreading and predict the short-term trends for topics. Mukherjee et al. [19] introduced an approach to identify trending concepts based on the hourly page visitation statistics. In [7], a dynamic Bayesian network-based method was proposed to solve the emerging topic detection problem. They selected features from the topology properties of topic diffusion to construct the DBN-based model. Becker et al. [1] studied the problem from another aspect. They proposed the use of an online clustering method to identify groups of topically similar

[1] https://about.twitter.com/company

SIGIR '17, August 7–11, 2017, Shinjuku, Tokyo, Japan
© 2017 ACM. ISBN 978-1-4503-5022-8/17/08…$15.00.
DOI: http://dx.doi.org/10.1145/3077136.3080791

tweets. Then, they computed features for each cluster to determine which clusters corresponded to events. However, these studies focused only on predicting whether a topic would be a trend or not. They could not determine which users would be attracted to or join in the discussion of the topic.

We think that predicting the topics whose discussions a user will join in advance will provide fine grained results about the topics that will become popular and could also produce valuable information for a variety of applications. Liang et al. [16] studied a similar task in an attempt to recommend topics for users. They proposed the use of an implicit information network to find the relevant topics. Since users are usually not join all the topics they are interested in, the ground truth of this task would be a problem. Some other existing studies have also been conducted on the task of recommending tweets based on users' personalized information (e.g., posting history, retweet history, and social network). These existing studies examined different aspects of the problem, including contextual information [5, 22], information about the relationships between users [9, 29], and multimodal information [6]. In contrast to recommending a single tweet, a topic consists of hundreds of tweets posted by different users.

In this work, we investigate the problem of predicting which topics a user will join in advance. This task has several challenges from different perspectives. A topic may contain hundreds or thousands of tweets. Tweets on a topic may also address different sub-topics or aspects. Moreover, because users usually write tweets in a conversational style in Twitter-like services, tweets are usually noisy and may contain misspelled or abbreviated words, as well as symbols. In addition to these challenges, two critical problems are obstacles to achieving this task. First, hundreds of thousands of topics exist in the real services. If we use traditional classification methods, the number of categories will be very large. The space complexity, time consumption, and performance are all challenging in large-scale classification problems. Second, the number of topics is not fixed. The topics on social media sites may change continuously. Hence, the classification module must be retrained to handle newly emerging targets. Therefore, the traditional classification methods cannot easily achieve a sufficient performance on this task.

To tackle these challenges, we first constructed a large dataset, which contained more than 14 million tweets. Then, we analyzed the dataset and found that the topics that a user joined were similar to the tweets he posted. Hence, we introduced a novel deep convolutional neural network with an attention mechanism to perform this task. We proposed the use of an external memory architecture to model the interests of users based on their posted tweets. Since the topics in which a user participated are more important and may have been slightly different with his general interests, we separately modeled this part for each user. To calculate the relations between users and topics, we introduced a matching-based method with an attention mechanism to construct similarity features between users and topics. Finally, the prediction results were calculated based on these similarity features.

The main contributions of this work are summarized as follows: 1) we defined the problem of predicting which topics a user will join. In contrast to the tweet recommendation task and early trend detection task, this task focuses on predicting the relations between users and topics; 2) we proposed a novel deep convolutional neural network with an attention mechanism to perform this problem. The user interests and main contents of topics were modeled by an external memory architecture; 3) to train and evaluate the proposed method, we constructed a large dataset of more than 14 million tweets. Experimental results demonstrate that the proposed method achieved better results than the state-of-the-art methods for this task.

2 RELATED WORK

There are three major areas related to this task and the proposed models. The first is the work on the task of topic prediction and recommendation on social media. The second is the matching problem. And the last is memory and attention mechanism. We will mainly introduce works about these areas in the following section.

2.1 Topic Prediction and Recommendation on Social Media

Because of the increasing requirements, various studies have recently been performed on the task of predicting which topics will become popular on social media. The problem was introduced in [20]. A nonparametric method was proposed to achieve the task. Some studies used different methods to perform this problem. Dang et al. [7] proposed a dynamic Bayesian network-based method to solve the problem. In [1], an online clustering methods was used to identify topics about event. However, these works focus only on predicting whether a topic would be a trend or not. They could not determine which users may join in the discussion of the topic.

Liang et al. [16] introduced the topic recommendation problem which studied how to recommend topics for users. They proposed to use the implicit information network formed by the multiple relationships among users, topics and tweets, and the temporal information of tweets to find relevant topics of each topic and to profile user's topic interest. Some other existing studies have also been conducted on the task of recommending tweets based on users' personalized information. Chen et al. [5] used collaborative filtering based on contextual information to solve this problem. Pan et al. [22] built a joint model which takes the advantages of both collaborative filtering and the characteristics of diffusion processes for recommendation. Meanwhile, some other studies were based on different information, such as relationship information among users[9, 29] and multimodal information[6]. Previous studies also evaluated a graph-theoretic model [34] and several different kinds of supervised classification methods [8, 30, 34]. All of these studies worked on the task of tweet recommendation. In our work, we investigate the problem of predicting which topics a user will join.

2.2 Matching Problem

Semantic matching is a critical task for many applications in natural language processing and several methods have been explored [14].

Recently, deep neural networks have been used to solve this problem and have shown outstanding performances. Huang et al. [12] developed a deep structure that projects queries and documents into a common low-dimensional space and matched the query and documents by calculating distance between the

Table 1: Statistics of the corpus we constructed

#User	15,210
#Tweets	33,326,572
#Topics	1,147
Time Period	2015.01.01-2015.12.31

low-dimensional representation. Lu et al. [17] proposed a new deep architecture combining the localness and hierarchy intrinsic for matching short texts. Similarly, Wang et al. [31] gave a deep matching model for using mined dependency tree matching patterns of two short text. Hu et al. [11] devised novel deep convolutional network architectures to match two sentence. Severyn and Moschitti [25] present a convolutional neural network architecture for reranking pairs of short texts. Palangi et al. [21] adopted a LSTM architecture to construct sentence representation.

In this study, we converted the prediction task into a matching problem between users and topics. We proposed an efficient deep convolutional neural network with an external memory to perform the task. The experiments showed that our proposed methods outperformed the others in our task.

2.3 Memory and Attention

Recently, variants of Memory Networks [32] have achieved very good results in various NLP tasks, such as language modeling [28], reading comprehension [10] and question answering [13, 18, 28]. Weston et al. [32] proposed this architecture and applied it on question answering, which have four component: input (I), generalization (G), output (O) and response (R) component. After then, Sukhbaatar et al. [28] introduced an end-to-end neural network with a recurrent attention model over a possibly large external memory. One important contribution of Memory Networks is the idea of storing the information in an external memory architecture and searching the important part from the memory by attention mechanism.

In this work, we proposed the use of an external memory architecture to model the interests of users and topics based on their posted tweets or contents and selected the important parts for each matching.

3 PRELIMINARY

3.1 Data Preparation

To analyze the topic of participation behavior, we crawled a large number of tweets from Twitter. First, we randomly selected 100 users as seeds. Then we crawled the followers and followees of these seed users. Through these steps, we crawled 15,210 users and more than 33 million tweets in total. All data were crawled before March 2016. Then, we restricted the time period used in our experiment to range from January 1, 2015 to December 31, 2015. Any information out of this time period were removed. More specifically, there are many kinds of topics on Twitter, such as a celebrities, hashtags, or keywords. Most users use the hashtag format. Thus, to obtain the topics from this time period, we selected the topics with the hashtag format and filtered out those topics that had a number of occurrences in our dataset larger than 1,000 to ensure the population of topics during the selected time period. The number of topics that satisfied this condition was about 1,147. In our experiments, if the user posted a tweet about the topic or retweeted a tweet about the topic, then we consider the user to have joined the topic. The statistics are listed in Table 1.

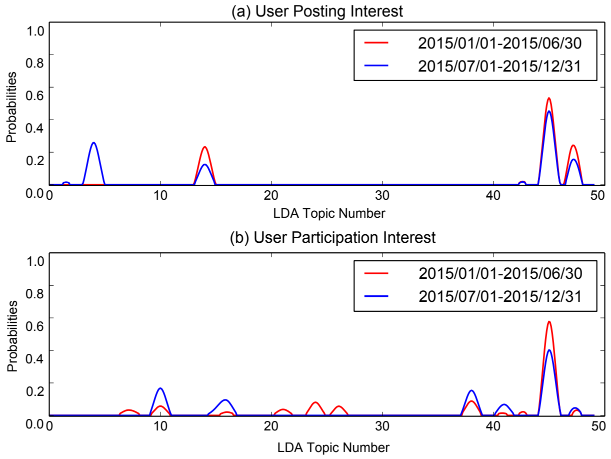


Figure 2: An Example of LDA Topic Distributions of User Interest. There are a few points of interest in user distributions and only minor differences between those distributions in different periods, which demonstrate that users normally focus on specific points and only occasionally join other topics.

3.2 Data Analysis

Normally speaking, most of us have the intuition that user interest was concentrated within a limited range, showing only a slightly difference for a long period of time. Each topic also covers one or more points of interest. It is more likely that the user will join the topics close to their interests. Thus, the topics that a user wants to join will be similar to the tweets he or she has posted. To verify these intuitions, we designed the following two experiments.

Hypothesis 1: *User interest is concentrated in a limited range and remains unchanged for a long time.*

There are mainly two types of behaviors that can represent user interest: posting behavior and participation behavior. To verify the hypothesis 1, we analyze the difference in these interests in different time periods. Firstly, we split our dataset into two time periods based on the 50% point-in-time in the timeline. In each unit time period, we randomly selected 50 posted tweets for each user to represent their posting behavior, and 50 tweets from topics they followed to represent their participation behavior. To discover interests from these tweet sets, we chose to directly apply Latent Dirichlet Allocation (LDA) [2003]. LDA can distill the collections of text documents (here, tweets) into distributions of words that tend to co-occur in similar documents. These sets of related words are referred to as "topics" in LDA, which can be also regarded as the points of user interest. We set the number of LDA topics to 50 and modeled our collections in different periods. In order to

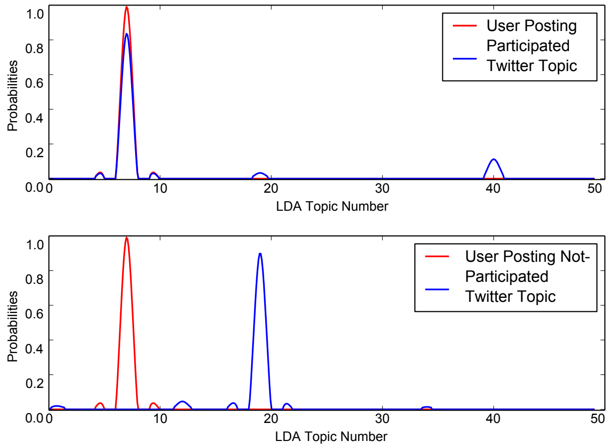


Figure 3: An Example of LDA Topic Distribution between User Postings and Twitter Topics. Topics that a user wants to join have a distribution similar to the tweet sets he or she posted, whereas the topics that a user will not join have a different distribution.

understand the differences between user interest in different times, we introduced the Kullback-Leibler (KL) divergence measure. If the divergence measure is below a certain threshold, meaning that the similarity of LDA topics distribution is high, then user interest remains unchanged. Empirically, we set the a threshold of 2 to measure the similarity between different distributions.

After the experimental analysis, the results revealed that more than 98% of users have similar posting interests between the two periods, because the KL-divergence between their two distributions is below the threshold. For participation interest, the proportion between the two periods is more than 86%. Such a high proportion of users with similar interests in the two different periods proves the consistency of user interest. For more intuitive information, we selected an example whose KL divergence was near the thresholds and plotted its smoothed distributions of LDA topics, shown in Figure 2. From the figure, we can see that there are a few points of interest in user distributions and only minor differences between those distributions. This data demonstrates that users usually focus on specific points and only occasionally join other topics for a special reason, such as important events.

Hypothesis 2: *The topics that a user wants to join will be similar to the tweets he or she has posted.*

To verify the hypothesis 2, we still use LDA to discover user posting interest and Twitter topic interest. For each user, we randomly selected 50 posted tweets as the user's posting documents and 50 tweets from Twitter topic content as the Twitter topic documents. Then, we performed negative sampling to select the topics in which the user will not participate and the sampling number was set to 50. KL divergence was also used to measure the differences between the two distributions. Based on statistics, we found that topics the user will follow have similarities with the user's posts. The KL divergence is low between them and the topics the user will not join have a higher KL divergence, just like the example shown in Figure 3. Certainly, some KL divergence between the negative samples and users is also low and the boundary is unclear, which made our prediction difficult. However, there is a huge difference between the positive samples and the negative samples, where the average KL divergence between the positive samples and users is 1.56, which is significantly lower than the KL divergence between the negative samples and users, which is 2.64. This statistical results demonstrate that the topics that a user wants to join are similar to the tweets he or she posted.

From the above analysis, we can conclude that user interests are concentrated in a limited range for a long period of time, and most of topics the users joined are close to their own posting interests. Thus, for this task, it is feasible to model interests using users' posts and participation history and to construct similarity features between users and topics.

4 APPROACH

In this work, given a user u and a list of topics T, our task is to predict which topics the given user will join. As described above, in this work, we convert this task to a matching problem between topics and users. Two types of information from different aspects are used in this work. One is posting history of users and the contents of topics. Another is the user's topic participation history.

To model the relevance between users and topics, we propose a novel and efficient memory-based convolutional neural network architecture with attention mechanism (MACNN). An overview of the proposed architecture is given in Figure 4. The proposed model has two main inputs, which are the posting history and participation history, as shown in Figure 4. First, two kinds of matching features are captured by well-designed architectures from different aspects. Then, we employ a concatenation layer to combine all these features. Finally, we use a multilayer perceptron to obtain the final preditiction.

4.1 Posting History Modeling

From the description above, we can see that the interests of users can in most cases be represented by the tweets that they post, and a topic also consists of a collection of tweets. Hence, in this work, determining whether the tweet set of a topic is relevant to the tweet set posted by a user is an important problem for our task. In this work, we propose a component to capture the relevance feature between the two posting collections. In this component, the original documents are stored in an external memory. For each user and each topic, a retrieval mechanism is used to select the important part for modeling the relevance. Because a user may pay attention to numerous topics and the tweets in a topic may also address different aspects, only a few tweets can be matched between the two collections. Thus, directly calculating the similarity between the two sets may not work well. A good matching method is necessary. However, it is inefficient and unnecessary to use the whole collections for matching. We select several tweets from the collections as the two tweet sets.

4.1.1 Document Representation. As described above, the original inputs are two documents $\mathbf{D}_u$ and $\mathbf{D}_t$ that contain the selected tweets, which are stored in the external memory. Each tweet in the

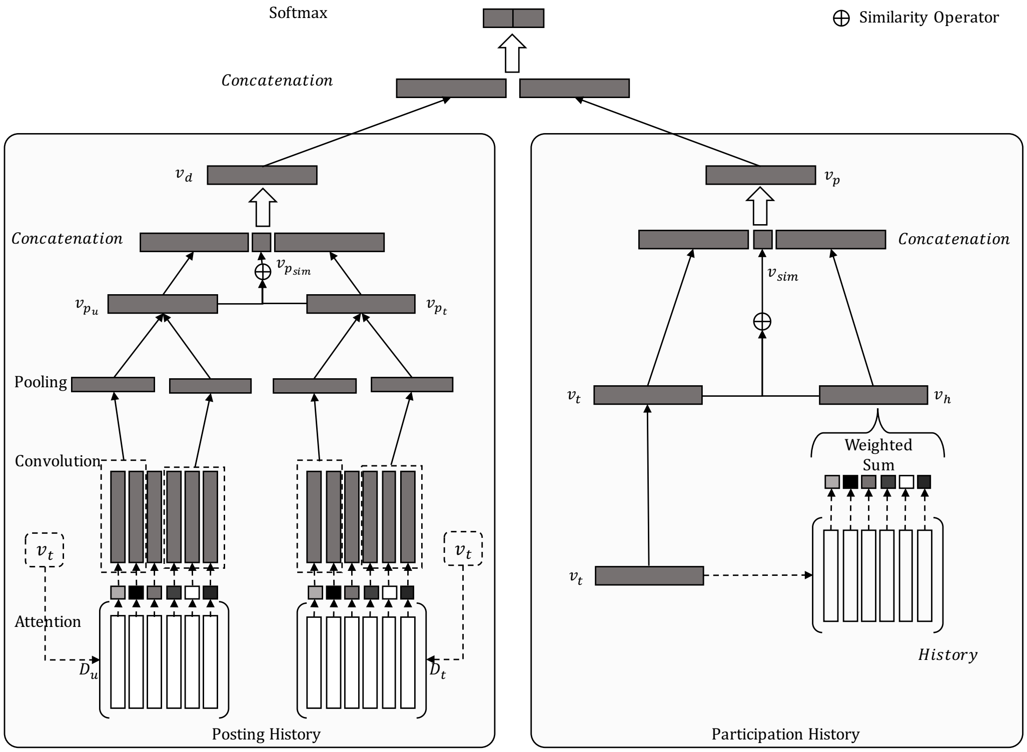


Figure 4: A Memory-based Convolutional Neural Network Architecture with Attention Mechanism

documents is treated as a sequence of words: $[w_1, ..., w_{|d|}]$. Then we embed each word w in the tweet in a continuous space and sum the embedding vectors to get the tweet representation. Specifically, the embedding matrix $\mathbf{E}_w$(of size $dim \times |V|$ where V is the word vocabulary and dim is the word embedding dimension) is used to look up the distributional vectors for words w, and the tweet representation can be calculated: $\mathbf{d} = \sum_i^{|d|} \mathbf{E}_w w_i, \mathbf{d} \in \mathbb{R}^{dim}$.

For each document $\mathbf{D} \in \mathbf{D}_u \bigcup \mathbf{D}_t$, we build a lower-level representation matrix:

$$\mathbf{D} = \begin{bmatrix} - & \mathbf{d}_1 & - \\ - & ... & - \\ - & \mathbf{d}_N & - \end{bmatrix}, \mathbf{D} \in \mathbb{R}^{N \times dim} \tag{1}$$

where each row represents a tweet embedding vector.

4.1.2 Attention Mechanism. However, before modeling the relevance of two sets, we have an underlying intuition that not all the tweets in the posting document are equally relevant for modeling the relevance. Different tweets in a set influence the matching between different topics and users to different extents, and tweets range from important to irrelevant. Based on this viewpoint, we propose the introduction of an attention mechanism to select the important parts of documents.

Given a topic t and user u, we first embed the topic into a vector $\mathbf{v}_t$, where $\mathbf{v}_t \in \mathbb{R}^{dim_t}$. Here, we have a embedding matrix $\mathbf{E}_t$ (of size $dim_t \times |V_t|$, where V_t is the topic vocabulary and dim_t is the topic embedding dimension), which is used to look up the vectors for topic t. After converting the topic into embedding, the next step is to build an attention layer with the help of the topic embedding. For each tweet $\mathbf{d}_i$ in the document, $\mathbf{D}$ should have its own degree of importance.

In the attention operator, we use the following definition to calculate the degree of importance of a single tweet:

$$\mathbf{m}_i = tanh(\mathbf{W}_d \mathbf{d}_i + \mathbf{W}_t \mathbf{v}_t), \tag{2}$$

$$p_i = \frac{exp(\mathbf{W}_m^{\mathrm{T}} \mathbf{m}_i)}{\sum_j^N exp(\mathbf{W}_m^{\mathrm{T}} \mathbf{m}_j)}, \tag{3}$$

where p_i represents the importance degree of the i-th tweet in the history document. N is the size of the posting history document. The parameters in this equation are $\mathbf{W}_d$, $\mathbf{W}_t$ and $\mathbf{W}_m$. Every tweet's degree of importance is calculated using the same parameters to ensure that the feature is learned in the same space.

Considering the impact of probabilities, for each tweet $\mathbf{d}_i \in \mathbf{D}$, we have a new embedding representation for matching between different pairs: $\mathbf{d_i} = p_i \mathbf{d}_i$. Through this step, we can obtain new document representations with their own importance degree values.

4.1.3 Posting Relevance. After adding the attention weights over the input tweets, an obvious solution to find the relevance between $(\mathbf{D}_u, \mathbf{D}_t)$ is to build document-level semantic representations and measure the similarity between them.

To form the posting interest representation, we propose a convolution architecture to aggregate the tweet interests and capture the semantic meaning of texts. Given the embedding set of tweets $\mathbf{D} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_N\}$, we first apply a 1-D convolution operation on the tweet embedding vectors. We use several convolution filters, which have different window sizes c. The i-th convolution output using window size c is given by the following:

$$\mathbf{h}_{c,i} = \sigma(\mathbf{W}_c \mathbf{d}_{i:i+c-1} + \mathbf{b}_c), \tag{4}$$

where $\mathbf{W}_c$ is the convolution weight, and $\mathbf{b}_c$ is the bias. $\sigma(\cdot)$ is an activation function such as the *sigmoid* function or *tanh* function. The filter is applied only to a window $i : i + c - 1$ of size c.

Then, we use a max-overtime pooling layer to aggregate the information that has passed through the convolution layer. The representation is reduced, and some useless information is filtered out using the pooling layer. The output of the pooling layer over the feature map using window size c is denoted as follows:

$$\mathbf{h}_{c,pooled} = \begin{bmatrix} pooling(\mathbf{h}_{c,1}) \\ \cdot \\ \cdot \\ \cdot \\ pooling(\mathbf{h}_{c,N-c+1}) \end{bmatrix}. \tag{5}$$

For the feature output from different convolution filters, we concatenate them to form the interest representation vector:

$$\mathbf{v}_p = [\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_c], \tag{6}$$

Through the steps described above, a user's posting interest and topic are denoted as two semantic vectors, $\mathbf{v}_{p_u}$ and $\mathbf{v}_{p_t}$, respectively. Next, we adopt the approach of [4] to calculate the similarity score between two vectors. This method considers interactions between different dimensions. Thus, it can capture complicated interactions. Specifically, this method defines the similarity score as follows:

$$v_{p_{sim}} = sim(\mathbf{v}_{p_u}, \mathbf{v}_{p_t}) = \mathbf{v}_{p_u}^T \mathbf{M}_p \mathbf{v}_{p_t}, \tag{7}$$

where $\mathbf{M}_p \in \mathbb{R}^{dim \times dim}$ is a similarity matrix used to reweight the interactions between different dimensions. The parameter matrix $\mathbf{M}_p$ will be optimized in the training process.

We employ a concatenation layer and hidden layer to combine the user posting interest vector, topic posting interest vector and similarity score to a fixed-size posting matching feature vector $\mathbf{v}_d$.

4.2 Participation History Modeling

The task in this work is predicting which topics the users will join in the future. Thus, the topic participation history should significantly improve the performance. Actually, it is feasible to obtain the users' topic participation history from social media websites and introduce an efficient architecture to utilize them. We denote the participation history as H_t, which contains many topics the user has joined. To measure the relevance between the given topic and the user participation interest, we proposed a brief method in our model.

Given a topic t and a user u, we find a corresponding participation history H_t and store it in an external memory architecture. Then, we embed each topic in history $t_i \in H_t$ into a continuous space:

$$\mathbf{v}_{t_i} = \mathbf{E}_t t_i, \mathbf{v}_{t_i} \in \mathbb{R}^{dim_t}, \tag{8}$$

where $\mathbf{E}_t$ is the same topic embedding matrix as that denoted in Section 4.1.2.

Next, we propose to use an attention mechanism to retrieve the topics that are relevant to the given topic t and aggregate the representation of this information to form the user participation interest representation. The weight p_{t_i} of topic t_i is calculated as follows:

$$\mathbf{m}_{t_i} = tanh(\mathbf{W}_t \mathbf{v}_t + \mathbf{W}_p \mathbf{v}_{t_i}) \tag{9}$$

$$p_{t_i} = \frac{exp(\mathbf{W}_{m_t}^{\mathrm{T}} \mathbf{m}_{t_i})}{\sum_j^M exp(\mathbf{W}_{m_t}^{\mathrm{T}} \mathbf{m}_{t_j})} \tag{10}$$

Then, the interest representation $\mathbf{v}_h$ of the user is constructed by summing the topic embedding $\mathbf{v}_{t_i}$, weighted by the probability p_{t_i}, as follows:

$$\mathbf{v}_h = \sum_i^M p_{t_i} \mathbf{v}_{t_i}. \tag{11}$$

To capture meaningful interactions between topic $\mathbf{v}_t$ and user interest $\mathbf{v}_h$, we apply the similarity operation introduced in Eq.(7) as follows:

$$v_{sim} = sim(\mathbf{v}_t, \mathbf{v}_h). \tag{12}$$

A concatenation layer and a hidden layer are applied to combine the topic embedding, user participation interest vector, and similarity score to a fixed-length participation history relevance feature vector $\mathbf{v}_p$.

4.3 Prediction

We use a multi-layer perceptron (MLP) and a softmax layer to determine whether or not u_i will join the discussion of topic t_j. The feature vectors are passed into the full connection hidden layer to obtain a higher-level representation:

$$\sigma(\mathbf{W}_h [\mathbf{v}_d, \mathbf{v}_p] + \mathbf{b}_h), \tag{13}$$

where $\mathbf{W}_h$ is the weight of the hidden layer, $\mathbf{b}_h$ is a bias, and $\mathbf{v}_d$ is the posting matching feature vector. $\mathbf{v}_p$ is the participation history relevance feature, and $\sigma(\cdot)$ is the non-linear activation function.

A softmax layer is used to predict the labels:

$$p(y = j|\mathbf{x}) = \frac{e^{\mathbf{x}^T \theta_s^j}}{\sum_{k=0}^K e^{\mathbf{x}^T \theta_s^k}}, \tag{14}$$

where θ_s^k is a weight vector of the k-th class and $j \in \{0, 1\}$ is the label of a pair. Here, $j = 0$ means that u_i will not join t_j, and $j = 1$ means that u_i will join t_j.

4.4 Training

Our training objective function is as follows:

$$J = \sum_{\mathbf{D}_{u_i} \in \mathbf{D}_u} \sum_{\mathbf{D}_{t_j} \in \mathbf{D}_t} -logp(y_{ij}|(\mathbf{D}_{u_i}, \mathbf{D}_{t_j})), \tag{15}$$

where $\mathbf{D}_u$ is the tweet set of users in the training corpus, $\mathbf{D}_t$ is the tweet set of the topics, and y_{ij} is a label that represents whether user u_i will join topic t_j.

The parameters θ are learned during the training as follows:

$$\theta = \{\mathbf{E}_w; \mathbf{E}_t; \mathbf{M}; \mathbf{W}_{m_d}; \mathbf{W}_{m_h}; \mathbf{W}_h; \mathbf{b}_h; \theta_s\}, \tag{16}$$

where $\mathbf{E}_w$ are the word embeddings. $\mathbf{E}_t$ are the topic embeddings. $\mathbf{M}$ are the similarity matrices. $\mathbf{W}_{m_d}$ are the parameters in the posting history matching component, and $\mathbf{W}_{m_h}$ are the parameters in the participation history component. The other parameters belong to the MLP and softmax layers.

In this study, stochastic gradient descent (SGD) with the adagrad update rule is used to optimize our model. Dropout regularization has proved to be an effective method for reducing the overfitting in deep neural networks with millions of parameters[26]. In this work, we also used it to improve the regularization of the hidden layer. Dropout regularization sets a portion of the hidden units to zero with probability p during the forward phase so that they will not contribute to the output of the softmax layer. $l2$-norm regularization terms are added to the parameters of the network to augment the cost function.

5 EXPERIMENT

5.1 Dateset and Setup

For training and evaluating the proposed method, we randomly selected 1500 users from the prepared dataset. Then, we filtered out the users whose language was not English or whose posting amount was smaller than 100 in the time period. As a result shown in Table 2, we ultimately had 1,183 users and 1,147 topics. To split and label the dataset, we set the split time point to 2015.07.01. We assume that we have known all the information before this point-in-time and the information after that is unknown. Thus, we can obtain a label list that shows whether the user will join the topic and the topic the user have joined before the point-in-time has been removed from the list. If the user posts a tweet about the topic or retweets a tweet about the topic during the period 2015.07.01-2015.12.31, the label is equal to one; otherwise, it is zero. In our experiment, we split the dataset into a training set, a development set and a test set. There were 946 users in the training set and 119 in the validation set. The remaining 118 users were in the test set. All of the tweets were processed by removing stopwords and special characters.

In the experiments, the embeddings for words were randomly initialized with each component sampled from the uniform distribution and all of the embeddings for topics were also randomly initialized from the uniform distribution with 200 dimension unless otherwise noted. The word embedding matrix E_w and the topic embedding matrix E_t were both non-static, and were tuned to the task-at-hand. Empirically, the selected architecture had two convolutional filters as well as two pooling layers to model the posting history interest. The widths c of the two convolution filters were set to $(1, 2)$, respectively. The two convolutional layers had 32 and 64 filter maps. In this work, the number of tweets randomly selected for each user or topic was set to 20 and the maximum number of topics in a user's participation history was set to 300. This configuration was also used in the other methods described in the following paragraphs. The network was used for training for 30 epochs with early stopping. The learning rate was set to $l = 0.01$, and the dropout rate was 0.2.

To evaluate the prediction correctness of our model, we used three metrics: Precision, Recall, and F1 score. In this work, we proposed a matching-based method and calculated the prediction scores between users and topics. The rank of the correct result based on these scores is also very important for many applications. Thus, we examined the ranks of the correct result using the mean average precision and precision at K results (denoted as MAP@K and P@K, respectively). The MAP equation is as follows:

$$\frac{1}{|U|}\sum_{u_i \in U} AveP(u_i), \tag{17}$$

where U is the user set and $|U|$ is the size of the user set.

Table 2: Statistics of the evalution collection

#User	1,183
#Tweets	1,800,442
#Topics	1,147
History Period	2015.01.01-2015.06.30
Prediction Period	2015.07.01-2015.12.31

We selected some effective methods to compare to our model, as follows:

- **Random**: Random method is implemented with a uniform random prediction.
- **NB**: Naive Bayes is implemented with the bag-of-word features transformed from the posting history and participation history.
- **SVM**: The support vector machine is implemented using libsvm with the bag-of-word features transformed from the posting history and participation history.
- **Collaborative Filtering**: Collaborative filtering algorithm is used to find top 50 related topics for each user. We used the standard user based collaborative filtering approach [15]. The similarity of two users was calculated based on the overlap of their topics.
- **UKNN**: UKNN is a topic recommendation method [16]. It proposed a user based KNN model to making use of the implicit information network formed by the multiple relationships among users and topics. We re-implemented it on our dataset to recommend top 50 related topics for each user.
- **ARC**: ARC is a convolutional matching model proposed by [11]. It uses a convolutional network to construct the document representation and thereby match the text. In this experiment, we use it to model the posting similarity between users and topics.
- **LSTM-RNN**: LSTM-RNN is used to construct posting history representations and obtains the final matching scores between them using cosine similarity [21].
- **LRCNN**: LRCNN is proposed by [25] and used here to model the posting similarity, which is a convolutional neural network architecture learning the representation of text pairs and a similarity function to relate them.
- **MemN2N**: MemN2N [27] is a neural network architecture with a recurrent attention model over a large external memory, which is used to model the participation history in this task. The hops of MemN2N is set to 3.

5.2 Results and Discussion

In Table 3, we list the prediction performances on our dataset using the different methods. In the experiments, we also introduced two variants of our proposed model: MACNN-Posting and MACNN-Participation. MACNN-Posting only uses the information about the posting history to predict the relationship between users and topics. MACNN-Participation is the right component shown in Figure 4, and uses the information about the participation history. Our proposed model, MACNN-All, combines these two types of

Table 3: Performances of different methods on the evaluation dataset

Methods	P@1	P@3	P@5	P@10	MAP@10	Precision	Recall	F1-Score
Random	0.008	0.024	0.020	0.024	0.006	0.025	**0.500**	0.048
NB	0.297	0.215	0.208	0.181	0.100	0.128	0.150	0.138
SVM	0.280	0.198	0.198	0.170	0.089	0.104	0.211	0.139
Collaborative Filtering [15]	0.313	0.254	0.217	0.198	0.107	0.123	0.243	0.163
UKNN [16]	0.364	0.266	0.239	0.192	0.120	0.109	0.224	0.146
ARC [11]	0.288	0.212	0.207	0.186	0.101	0.117	0.181	0.142
LSTM-RNN [21]	0.254	0.215	0.178	0.171	0.091	0.110	0.172	0.135
LRCNN [25]	0.280	0.234	0.224	0.182	0.103	0.120	0.183	0.145
MemN2N [27]	0.347	0.266	0.236	0.208	0.121	0.138	0.223	0.170
MACNN-Posting	0.297	0.215	0.219	0.195	0.110	0.150	0.192	0.168
MACNN-Participation	0.314	0.297	0.256	0.227	0.132	0.143	0.243	0.180
MACNN-All	**0.390**	**0.342**	**0.285**	**0.238**	**0.156**	**0.175**	0.236	**0.201**

information. For random results, we implemented a random guess method with random prediction. The precision was about 0.024 because the average number of topics in which a user participated was about 25. The MAP@10 was 0.006, which gave an indication of the difficulty of the task. From the tables, we can see that the proposed model is significantly better than the other methods because it obtains the best results using these important metrics.

From the results shown in Table 3, we can observe that our approach achieves a relative improvement of 18.2% in F1 score, 14.4% in P@10, and 28.9% in MAP@10 over MemN2N. MACNN-All also provides improvements of 0.021 for F1 score, 0.011 for P@10, and 0.024 for MAP@10 over MACNN-Participation. It shows that the proposed model makes effective use of the two types of information. Observing the results for ARC, LSTM-RNN, and LRCNN, we see that these neural network architectures for text matching are also effective in our task. All of them predicted the result based on the posting history information, which proves that a matching-based method for posting history information can achieve the task. However, because in this task we want to match two tweet sets that contain numerous tweets where only a part of the tweets are important for interest modeling and these architectures are designed for the sentence-level semantic matching task, the results of these methods are worse than our models at performing this task. Considering the comparison between MACNN-Posting and these methods, it is clear that our model can model a better similarity feature for the posting history, and the degree of importance of the tweets is a key ingredient of prediction. Our model successfully captures the different influences of tweets and thus achieves a strong performance.

From Table 3, we can also observe that MACNN-Participation achieves a better performance than MACNN-Posting, and MemN2N also obtains better results than the other models using the posting history, which demonstrates that the participation history is more important in this task. In the comparison between MemN2N and MACNN-Participation, our model is also better. We analyzed the difference between the two architectures and found that the similarity function introduced in our model could better capture the relevance feature between users and topics. We also evaluated some classical methods such as Naive Bayes, SVM. Their results were worse than the others.

Considering the results of collaborative filtering and UKNN, we can observe that the recommendation-based methods are more suitable than classification-based methods. We think that the huge number of labels is one of the main reasons why supervised classification methods do not work well on this task. Based on the relationship between users and topics, these two methods can find the most likely topics for each user. UKNN achieved a high performance for P@1, which agrees with the original implementation in [16]. The results indicated that our prediction results for which topics a user will join may be helpful in constructing topic recommendation systems. Our model also showed its advantages in this situation, as shown by the performances listed in Table 3.

As we known, there are many types of Twitter users based on the number of topics in which they participate. Intuitively, since heavy social media users are willing to participate in discussions and post a variety of tweets, they are involved in more topics. It should be easier to predict these users' participation behaviors because of the richer information about their interests and a higher proportion of positive results; for other users, it must be more difficult. In Figure 5, we split the users into three groups based on how many topics they participated in. The results of our proposed model (MACNN-All) for these sets are shown on the figure. Users who participated in less than 20 topics are contained in the first group, and those who participated in more than 20 but less than 50 compose the second set. The third group is composed of heavy social media users.

From Figure 5, we can observe that the results for the heavy users were the best, proving our hypothesis. Because of the low activity of light users, it is difficult for us to obtain useful information from their history and its proportion of positive results was lower than the others, which led to the worst results in the figure. Even though we introduced the posting history in our model to help solve the cold-start problem to some extent, it is still a great challenge for future work. The model achieved much better performances for normal users and heavy users than light users.

Table 4: Performances of MACNN with different parameters

Methods	Window Sizes	Word Embedding Dim.	Topic Embedding Dim.	Precision	Recall	F1-Score
MACNN-Posting	(1,2)	200	-	0.150	0.192	0.168
MACNN-Participation	-	-	200	0.143	0.243	0.180
MACNN-All	(1,2)	50	200	0.158	0.231	0.187
	(1,2)	100	200	0.167	0.237	0.196
	(1,2)	300	200	0.169	0.240	0.199
	(1)	200	200	0.171	0.227	0.195
	(1,2)	200	200	**0.175**	0.236	**0.201**
	(1,2, 3)	200	200	0.169	0.247	0.200
	(1,2)	200	300	0.166	0.244	0.198
	(1,2)	200	100	0.167	**0.251**	0.200
	(1,2)	200	50	0.150	0.217	0.177

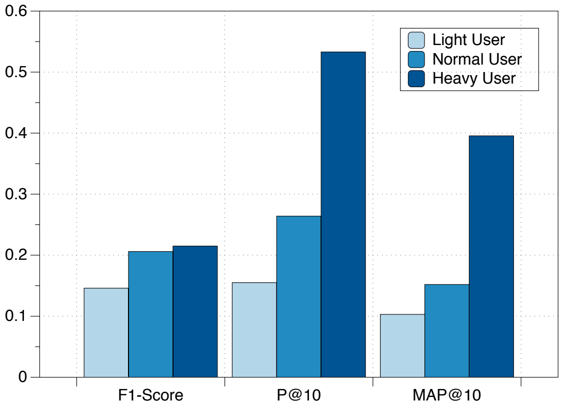


Figure 5: The results between different types of users.

5.3 Parameter Influence

There are several hyper-parameters that could influence the performances of our proposed models. This section explains how we changed these parameters to evaluate their influences. In the experiment, we investigated how filters with different window sizes c in a convolution architecture affected our results. Then, we experimented with different word embedding dimensions and topic embedding dimensions. We varied one parameter at a time, while fixing the other two. The results are listed in Table 4.

First, we changed the window size c of the filters and tested our model with three different size sets. More types of filters can capture richer information from the posting history and model better similarity features between users and topics. However, the experiment showed that the performance did not improve as the number of filters with different window sizes increased. As can be seen, we obtained the best result using MACNN-All with two types of filters with sizes of $(1, 2)$. The model with only one filter with a size of (1) obtained the worst result, while the results for the model with sizes of $(1, 2, 3)$ were comparable to those of the best one. To achieve a good performance, it was better to use filters with various window sizes.

Then, we analyzed the influence of the dimensions chosen for two types of embedding vectors: word embedding vectors and topic embedding vectors. A suitable dimension for the embedding vectors could enhance the feature expression ability, whereas a lower dimension achieved a lower performance. In this experiment, we changed the word and topic embedding dimensions separately and randomly initialized the embedding vectors with dimensions ranging from 50 to 300, with the results listed in Table 4. For the two types of embedding, a higher embedding dimension resulted in a better performance. When dimensions are both larger than 100, our proposed model performed well. When one of dimensions is equal to 50, the performance is bad. Our proposed model performed well with a high embedding dimension. When the two dimensions were equal to 200, we obtained the best results, which showed that our model needs a high dimension to store more feature information. However, there were some differences between the two embeddings. The dimension of the topic embeddings had a larger impact on the prediction, which met our expectations because the participation history was more important in this task. For a better prediction and to ensure the robustness of the proposed model, it is suggested that high embedding dimensions should be selected.

6 CONCLUSION

In this work, we studied the problem of predicting which topics a user will join and collected a large evaluation dataset consisting of more than 14 million tweets. We proposed a novel deep convolutional neural network with an attention mechanism to solve this problem. Specifically, we modeled the user interest and main contents of topics with an external memory architecture. The posting similarity features between users and topics were captured by the deep convolutional neural network. Because tweets have different degrees of importance, we also proposed to incorporate an attention mechanism to select the important ones. Participation features were also modeled using an attention network. Then, we passed the feature vectors into a multi-layer perceptron to obtain the final matching score. The experimental results on the evaluation dataset showed that the proposed methods outperformed other methods.

ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 61532011, 61473092, and 61472088) and STCSM (No.16JC1420401).

REFERENCES

[1] Hila Becker, Mor Naaman, and Luis Gravano. 2011. Beyond Trending Topics: Real-World Event Identification on Twitter. *ICWSM* 11 (2011), 438–441.

[2] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[3] Johan Bollen, Huina Mao, and Xiaojun Zeng. 2011. Twitter mood predicts the stock market. *Journal of Computational Science* 2, 1 (2011), 1–8.

[4] Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open Question Answering with Weakly Supervised Embedding Models. In *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I.* 165–180. DOI:http://dx.doi.org/10.1007/978-3-662-44848-9_11

[5] Jilin Chen, Rowan Nairn, Les Nelson, Michael Bernstein, and Ed Chi. 2010. Short and Tweet: Experiments on Recommending Content from Information Streams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '10).* ACM, New York, NY, USA, 1185–1194. DOI:http://dx.doi.org/10.1145/1753326.1753503

[6] Kailong Chen, Tianqi Chen, Guoqing Zheng, Ou Jin, Enpeng Yao, and Yong Yu. 2012. Collaborative Personalized Tweet Recommendation *(SIGIR '12).* ACM, 661–670. DOI:http://dx.doi.org/10.1145/2348283.2348372

[7] Qi Dang, Feng Gao, and Yadong Zhou. 2016. Early detection method for emerging topics based on dynamic bayesian networks in micro-blogging networks. *Expert Systems with Applications* 57 (2016), 285–295.

[8] Ernesto Diaz-Aviles, Lucas Drumond, Lars Schmidt-Thieme, and Wolfgang Nejdl. 2012. Real-time Top-n Recommendation in Social Streams. In *Proceedings of the Sixth ACM Conference on Recommender Systems (RecSys '12).* ACM, 59–66. DOI:http://dx.doi.org/10.1145/2365952.2365968

[9] Ido Guy, Naama Zwerdling, Inbal Ronen, David Carmel, and Erel Uziel. 2010. Social Media Recommendation Based on People and Tags. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10).* ACM, 194–201. DOI:http://dx.doi.org/10.1145/1835449.1835484

[10] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv preprint arXiv:1511.02301* (2015).

[11] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional Neural Network Architectures for Matching Natural Language Sentences. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada.* 2042–2050.

[12] Po-Sen Huang, Xiaodong He, Jianfeng Gao, Li Deng, Alex Acero, and Larry P. Heck. 2013. Learning deep structured semantic models for web search using clickthrough data. In *CIKM'13, San Francisco, CA, USA, October 27 - November 1, 2013.* 2333–2338.

[13] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. *CoRR, abs/1506.07285* (2015).

[14] Hang Li and Jun Xu. 2014. Semantic Matching in Search. *Foundations and Trends in Information Retrieval* 7, 5 (2014), 343–469. DOI:http://dx.doi.org/10.1561/1500000035

[15] Huizhi Liang, Yue Xu, Yuefeng Li, Richi Nayak, and Xiaohui Tao. 2010. Connecting users and items with weighted tags for personalized item recommendations. In *Proceedings of the 21st ACM conference on Hypertext and hypermedia.* ACM, 51–60.

[16] Huizhi Liang, Yue Xu, Dian Tjondronegoro, and Peter Christen. 2012. Time-aware topic recommendation based on micro-blogs. In *Proceedings of the 21st ACM international conference on Information and knowledge management.* ACM, 1657–1661.

[17] Zhengdong Lu and Hang Li. 2013. A Deep Architecture for Matching Short Texts. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.* 1367–1375. http://papers.nips.cc/paper/5019-a-deep-architecture-for-matching-short-texts

[18] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* (2016).

[19] Sayandev Mukherjee, Ronald Sujithan, and Pero Subasic. 2014. Detecting Trending Topics Using Page Visitation Statistics. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion).* ACM, New York, NY, USA, 347–348. DOI:http://dx.doi.org/10.1145/2567948.2577303

[20] Stanislav Nikolov and Devavrat Shah. 2012. A nonparametric method for early detection of trending topics. In *Proceedings of the Interdisciplinary Workshop on Information and Decision in Social Networks (WIDS 2012). MIT.*

[21] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward. 2016. Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. *IEEE/ACM Trans. Audio, Speech & Language Processing* 24, 4 (2016), 694–707. DOI:http://dx.doi.org/10.1109/TASLP.2016.2520371

[22] Ye Pan, Feng Cong, Kailong Chen, and Yong Yu. 2013. Diffusion-aware personalized social update recommendation. In *Proceedings of the 7th ACM conference on Recommender systems.* ACM, 69–76.

[23] Michael J Paul and Mark Dredze. 2011. You are what you Tweet: Analyzing Twitter for public health. *ICWSM* 20 (2011), 265–272.

[24] Takeshi Sakaki, Makoto Okazaki, and Yutaka Matsuo. 2010. Earthquake shakes Twitter users: real-time event detection by social sensors. In *Proceedings of the 19th international conference on World wide web.* ACM, 851–860.

[25] Aliaksei Severyn and Alessandro Moschitti. 2015. Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15).* ACM, New York, NY, USA, 373–382. DOI:http://dx.doi.org/10.1145/2766462.2767738

[26] Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958. http://dl.acm.org/citation.cfm?id=2670313

[27] Sainbayar Sukhbaatar, arthur szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 2440–2448. http://papers.nips.cc/paper/5846-end-to-end-memory-networks.pdf

[28] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, and others. 2015. End-to-end memory networks. In *Advances in neural information processing systems.* 2440–2448.

[29] Jing Sun and Yangyong Zhu. 2013. Microblogging personalized recommendation based on ego networks. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, Vol. 1. IEEE, 165–170.

[30] Ibrahim Uysal and W Bruce Croft. 2011. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM international conference on Information and knowledge management.* ACM, 2261–2264.

[31] Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2015. Syntax-Based Deep Matching of Short Texts. In *IJCAI 2015.* 1354–1361. http://ijcai.org/papers15/Abstracts/IJCAI15-195.html

[32] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[33] Kumanan Wilson and John S Brownstein. 2009. Early detection of disease outbreaks using the Internet. *Canadian Medical Association Journal* 180, 8 (2009), 829–831.

[34] Rui Yan, Mirella Lapata, and Xiaoming Li. 2012. Tweet recommendation with graph co-ranking. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1.* Association for Computational Linguistics, 516–525.

[35] Juanjuan Zhao, Weili Wu, Xiaolong Zhang, Yan Qiang, Tao Liu, and Lidong Wu. 2013. A short-term prediction model of topic popularity on microblogs. In *International Computing and Combinatorics Conference.* Springer, 759–769.