Mention Recommendation for Twitter with End-to-end Memory Network

Haoran Huang
School of Computer Science
Fudan University
Shanghai, P.R.China
huanghr15@fudan.edu.cn

Qi Zhang
School of Computer Science
Fudan University
Shanghai, P.R.China
qz@fudan.edu.cn

Xuanjing Huang
School of Computer Science
Fudan University
Shanghai, P.R.China
xjhuang@fudan.edu.cn

Abstract

In this study, we investigated the problem of recommending usernames when people attempt to use the "@" sign to mention other people in twitter-like social media. With the extremely rapid development of social networking services, this problem has received considerable attention in recent years. Previous methods have studied the problem from different aspects. Because most of Twitter-like microblogging services limit the length of posts, statistical learning methods may be affected by the problems of word sparseness and synonyms. Although recent progress in neural word embedding methods have advanced the state-of-the-art in many natural language processing tasks, the benefits of word embedding have not been taken into consideration for this problem. In this work, we proposed a novel end-to-end memory network architecture to perform this task. We incorporated the interests of users with external memory. A hierarchical attention mechanism was also applied to better consider the interests of users. The experimental results on a dataset we collected from Twitter demonstrated that the proposed method could outperform state-of-the-art approaches.

1 Introduction

In Twitter-like social media, a tweet may contain some other users @*username* anywhere in the body of the tweet. If a tweet includes multiple @usernames, all of those people will see it in their notifications tab. According to the definition of Twitter, a tweet that contains @*username* is called a mention. Along with the dramatic increase in Twitter-like microblogging services, a huge number of users frequently use these applications and treat them as their main communication methods. According to a statistic on Twitter, the average time spent on Twitter monthly is almost 170 minutes, and the average number of followers per user was 208[1]. Hence, when people want to mention others in a tweet, a small number of candidates for the specific tweet would benefit many of these users.

Previous works have studied the mentioned recommendation problem from different aspects. Wang et al. [2013] proposed a recommendation scheme using several manually constructed features related to a users interests to expand the diffusion of tweets by recommending proper users to mention. Li et al. [2015] considered this recommendation as probabilistic and proposed a factor graph method to achieve it. Instead of trying to expand the diffusion of tweets, some works have focused on targeting the right person for the mention behavior. Tang et al. [2015] employed a ranking support vector machine model to locate target users. Gong et al. [2015] studied a task similar to our study. They treated the recommendation task as a translation problem. Both microblog content and user histories were incorporated into a topical translation model to perform the task.

In recent years, neural network-based methods have been used for a variety of different tasks [Collobert *et al.*, 2011; LeCun *et al.*, 2015; Chen and Manning, 2014; Zeng *et al.*, 2014; Levine *et al.*, 2016]. Because of the capability of naturally integrating word embeddings, the problems of word sparseness and synonyms can be resolved to some degree. More recently, some progress has been made on complex tasks based on incorporating the addition of memory and an attention mechanism into the network architecture. For example, memory networks [Weston *et al.*, 2015b] could reason with inference components combined with a long-term memory component and achieved state-of-the-art performance on a question answering task. Sukhbaatar et al. [2015] introduced an end-to-end memory network with a recurrent attention model over a possibly large external memory.

Since mentioned users are usually highly related to the tweet, how to measure the similarity between the interests of the candidate users and the tweet is an important factor for performing this task. Motivated by the advantages and progress of memory networks, in this paper, we propose a novel end-to-end memory network [Sukhbaatar *et al.*, 2015] architecture to perform this task. The proposed network architecture adopted the end-to-end neural memory network to incorporate the content of a tweet, history of its author, and interests of candidate users into consideration. It consists of three main components to model the tweet, interests of the author, and interests of a candidate user. The interests of the author and candidate user are incorporated into the external memory parts. The proposed method iterates multiple hops

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

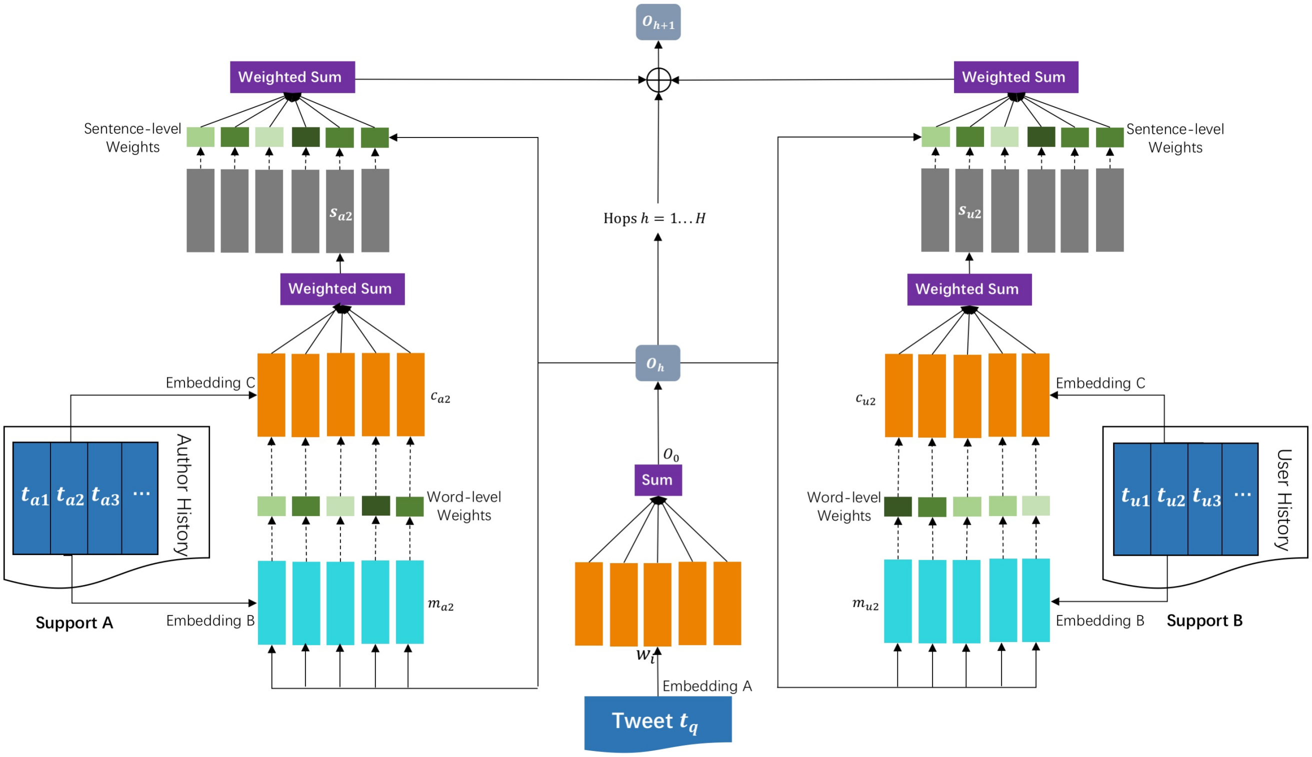

Figure 1: The Architecture of the Proposed End-to-end Memory Network (AU-HMNN).

to generate an internal representation of the tweet, interest of the author, and interest of the candidate user. Finally, a fully connected softmax layer is used to model the prediction.

The three main contributions of our work are as follows: 1) we introduced an end-to-end memory network to perform the user recommendation problem for the mention action in twitter; 2) we proposed a novel network architecture to incorporate the content of a tweet, interests of the user, and interests of the author; 3) through several experimental results, we show that neural networks taking the word embeddings as input can achieve better performance than previous state-of-the-art methods that used BoW-based representations in most cases.

2 Approach

In this paper, we formulate the task of user recommendation for mention action as a matching problem. Given a tweet t_q, its author a and a list of candidate users U, our task is to determine whether a user $u \in U$ should be recommended for the author's mention action in the tweet t_q. We introduce a novel memory network architecture AU-HMNN to solve this matching problem, as illustrated in Figure 1.

Our proposed model utilizes the tweet t_q and the supporting memory which contains two parts: author history and candidate user history. There are three main components in this memory network architecture. First, we use a tweet encoder to represent the tweet. Second, we encode the history interests of the author and the history interests of the candidate user with the help of t_q. In this step, we introduce a hierarchical attention mechanism to help the encoder capture high-quality history interests information. Then, our model can find a continuous internal representation for the tweet t_q, the author a and the user u and the representation can be processed via multiple hops to the output layer, where the hops are denoted as $h = \{0, 1, 2, ..., H\}$. Finally, we use a fully connected softmax layer for the matching prediction. We will describe the details of our proposed model in the following sections.

2.1 Supporting Memory

In this work, user interests can be represented by a set of tweets that users post. Hence, we use the tweet sets D_a and D_u to represent the history interests of the author and the candidate user, respectively. Each tweet set D contains many tweets denoted by $D = \{t_1, t_2, ..., t_N\}$ and each tweet t contains many words denoted by $t = \{w_1, w_2, ..., w_M\}$, where N is the size of the tweet set and M is the length of the tweet.

As described above, we split the supporting memory into two parts: author history and user history, store these sets of tweets in the supporting memory and read from the memory multiple times to capture their history interests representation.

2.2 Tweet Modelling

To find representation of the query tweet t_q, we use a simple encoder to embed it. The tweet t_q is treated as a bag-of-words representation and the embedding weight A is used to look up the vectors for words $w \in t_q$ here. The size of A is $d \times |V|$, where d is the embedding dimension and $|V|$ is the size of vocabulary. Each word w_q in t_q is embedded in a continuous space and then we sum these embedding vectors to obtain the representation of the tweet: $t_q = \sum_i Aw_i$. The representation of t_q is also treated as the initial internal input o_0, which helps the model read information from the supporting memory for the next hops.

2.3 History Interests Modelling

Based on the description above, we can see that the history document stored in the memory has a hierarchical structure. Each document has many tweets: $D = \{t_1, t_2, ..., t_N\}$, and a sentence-level structure. Each tweet has words: $t = \{w_1, w_2, ..., w_M\}$, and a word-level structure. Hence, we propose a two-level encoder architecture to model the history interests. Meanwhile, with an underlying intuition that not all tweets in the history document are equally relevant for modelling the interests and not all words in tweets are equally important, we introduce a hierarchical attention mechanism in the encoder.

Word-level encoder

Given an input set $\{t_1, t_2, ..., t_i\}$, first, each word $w_{ij} \in t_i$ is embedded into a memory vector m_{ij} (of dimension d) using a embedding matrix B (of size $d \times |V|$), giving $m_{ij} = Bw_{ij}$. For each tweet, we obtain a matrix representation of size $M \times d$, where M is the length of the tweet. The memory vector m_{ij} in this process step we called is input memory, which projects the input into a same space. The next step addresses the attention layer. The attention layer makes it possible to pay attention to the interests and for different interests to operate with different weights. In this work, we use the input memory to lookup the important input positions with the help of the internal information. The match between input memory vector m_{ij} and o_h is then computed by taking the inner product followed by a softmax:

$$p_{ij} = \frac{\exp(o_h^{\mathrm{T}} m_{ij})}{\sum_m^M \exp(o_h^{\mathrm{T}} m_{im})}, \tag{1}$$

where o_h is the internal state in hop h, m_{ij} is the input memory vector of w_{ij}, and p is the probability over the input position.

Then, each word w_{ij} is embedded into another memory vector c_{ij}(of dimension d and called output memory) using another embedding matrix C (of the same size with B). Finally, the representation of the tweet is obtained by summing the output memory weighted by the probability:

$$s_i = \sum_j^N p_{ij} c_{ij}, \tag{2}$$

where s_i is the embedding of the tweet t_i, c_{ij} is the output memory of jth word w_{ij} in t_i and p_{ij} is the probability over c_{ij}.

From the above procedure, each tweet in the history document is converted into a fixed-length vector which represents the interest embedding of the tweet.

Sentence-level encoder

To form the user history interests representation, we propose a sentence-level encoder to aggregate the tweet interests and extract the important parts. Given the embedding set of tweets $s = \{s_1, s_2, ..., s_i\}$, the interest representation of the history document is formed by a weighted sum of these tweet embeddings. The weights over the input tweets are interpreted as the degree of importance of a particular tweet in the document; the equation of this operation is as follows:

$$m_{s_i} = tanh(W_o o_h + W_s s_i), \tag{3}$$

$$p_{s_i} = \frac{exp(W_{ms}^{\mathrm{T}} m_{s_i})}{\sum_j^N exp(W_{ms}^{\mathrm{T}} m_{s_j})}, \tag{4}$$

$$r = \sum_i^N p_{s_i} s_i, \tag{5}$$

where p_{s_i} is the normalized attention at tweet t_i, N is size of the history document and r is the history interests embedding. The parameters in these equations are W_o, W_s and W_{ms}.

In our proposed model, the supporting memory stores author history and user history, both of which can be modeled by the proposed encoder. In each hop h, the history interests embedding of author a and that of user u can be combined with the last internal state to update the state: $o_{h+1} = a + o_h + u$. The hops operator allows the model to recurrently accumulate information from the supporting memory, ultimately producing a final joint representation for the prediction.

2.4 Final Prediction

Based on the representation obtained from the above process, we introduce a multi-layer perceptron(MLP) and a softmax layer to determine whether or not user u should be recommended for the author's "@" action in the tweet t_q. The feature representation is passed into the full connection hidden layer:

$$f = \sigma(W_m o_H + b_m), \tag{6}$$

where W_m is the weight vector of the hidden layer, b_m is a bias, o_H is the final representation obtained from the last hop and $\sigma(\cdot)$ is the non-linear activation function.

Finally, we use a softmax layer to predict:

$$p(y = i|f; \theta_s) = \frac{exp(\theta_s^i f)}{\sum_j exp(\theta_s^j f)}. \tag{7}$$

According to the scores output from the softmax layer, we can list the top-ranked recommended users for "@" action.

2.5 Training

In this work, the training objective function is formulated as follows:

$$J = \sum_{(t_q,i,a,u)\in D} -\log p(i|t_q; a, u), \tag{8}$$

where D is the training corpus. $i \in \{0, 1\}$ is the label of triple (t_q, a, u), when $i = 0$ means the user u should not be recommended for the author a's "@" action in the tweet t_q and $i = 1$ represents the user u should be recommended.

The parameters of our model are listed as follows:

$$\theta = \{A, B, C, W_o, W_s, W_{ms}, W_m, b_m, \theta_s\}, \qquad (9)$$

where A, B and C are embedding matrix. W_o, W_s and W_{ms} are the parameters in sentence-level encoder. W_m and b_m are weights and bias of MLP. θ_s is the parameter of softmax layer.

To minimize the objective function, we use stochastic gradient descent (SGD) with the adagrad update rule. Then, we use the dropout and add l_2-norm terms for the regularization.

3 Experiment

3.1 Dataset and Setup

To evaluate the effectiveness of our proposed model, we constructed a dataset from Twitter.

In the first step, we randomly selected 4,000 users as the central authors and crawled their post histories. In this step, we collected 9,461,820 tweets. Second, we selected the tweets that contained at least one @username and collected the corresponding mentioned users. We found that 3,150 central authors had the mention behavior, and a total of 133,267 query tweets with at least one "@username" were gathered. The number of mentioned users was 18,782. Finally, we crawled the mentioned users' histories, and 42,205,577 tweets were collected. Based on the statistics shown in Table 1, the average number of mention behaviours per central author was 42.3, and the average number of users that the central authors' mentioned was 17.9. For each query tweet, the list of mentioned users annotated with authors was treated as the ground truth, and the mentioned history of each author was considered as a candidate. Finally, we split the dataset into training and testing sets with an 80/20 ratio.

In this work, the memory capacity was restricted to 5 tweets, and the maximum length of each tweet was 32. For each author and each mentioned user, we randomly extracted 5 tweets from their history matching their history interests and stored them in the supporting memory. The embedding dimension in the experiment was set to 300, and the number of hops was set to 6. The learning rate was set to 0.01, and the dropout rate was set to 0.2.

We evaluated the results using the following standard information retrieval metrics. We used Precision, Recall, and F-Score for the highest ranked result, and then used Hits@3 and Hits@5 to measure the percentage of mentioned users to be correctly recommended from the top n results. To evaluate the rank of the recommended results, in this study, we also used the Mean Reciprocal Rank (MRR) metrics.

3.2 Baselines

For comparison with the proposed model, we evaluated some effective methods as baselines and introduced two degeneration models, which can be described as follows:

- **Frequency Descending (FD)**: The ranked list depends on the frequency ranks of the candidates in the history. A candidate user mentioned with a higher frequency by the author would have a higher rank in the recommendation list.

Table 1: Statistics of the Constructed Dataset

# Tweets	51,800,664
# User	22,782
#Avg.Mention per Author	42.3
#Avg.Mentioned User/Author	17.9

- **PMPR**: Personalized Mention Probabilistic Ranking (PMPR) system is proposed in [Li *et al.*, 2015] to achieve the mention recommendation problem.
- **Ranking**: Ranking is a ranking support vector machine model proposed in [Tang *et al.*, 2015] to locate the target users.
- **A-UUTTM**: A-UUTTM is the translation-based model proposed in [Gong *et al.*, 2015], which considered not only the content of a microblog but also the histories of candidate users, was the state-of-the-art approach used for this task.
- **U-MNN**: U-MNN is a variant of our proposed model, which considered the history interests of the candidate users. The history interest encoder used in this model is a memory network architecture without a hierarchical attention mechanism, which was proposed in [Sukhbaatar *et al.*, 2015].
- **U-HMNN**: U-HMNN uses an encoder with a hierarchical attention mechanism to model the history interests of the candidate users.
- **AU-HMNN**: The model proposed in this paper incorporates the textual information of query tweets and the history interests of the author and candidate users. The history interests encoder is a memory network architecture with a hierarchical attention mechanism.

3.3 Results and Discussion

Table 2 lists the performances of different methods on our dataset. From the table, we can observe that our proposed model (AU-HMNN) consistently achieves a better performance than the comparison methods. In all the metrics, AU-HMNN is significantly better than the other methods. Compared with A-UUTTM, which was the state-of-the-art method, the proposed model (AU-HMNN) achieves a relative improvement of 7.8% in precision, along with a 7.9% increase in recall and 7.9% increase in the F-score. The best results for Hits@3 and Hits@5 are greater than 0.876 and 0.902, respectively, which means 87.6% of the correct users will be found in the top 3 recommendation list and 90.2% of the users can be recommended in the top 5. The MRR result of AU-HMNN is also better than the others, which demonstrates that the rank of the result is related to a better recommendation of candidate users.

For the FD results, we implemented it by ranking the users based on the frequency of its mentioned history. Intuitively, because authors always mention the users they know well, the FD method should be suitable for some authors, which means it should perform well on this task. Experiment results have also proved this conclusion. We can see that the

Table 2: The Performances of Different Methods on the Testing Dataset

Method	Precision	Recall	F-Score	MRR	Hits@3	Hits@5
FD	0.496	0.489	0.493	0.745	0.642	0.692
PMPR	0.432	0.426	0.429	0.577	0.654	0.753
Ranking	0.586	0.580	0.583	0.697	0.765	0.829
A-UUTTM	0.755	0.745	0.750	0.820	0.856	0.871
U-MNN	0.729	0.720	0.725	0.787	0.806	0.846
U-HMNN	0.792	0.782	0.787	0.839	0.858	0.888
AU-HMNN	**0.814**	**0.804**	**0.809**	**0.857**	**0.876**	**0.902**

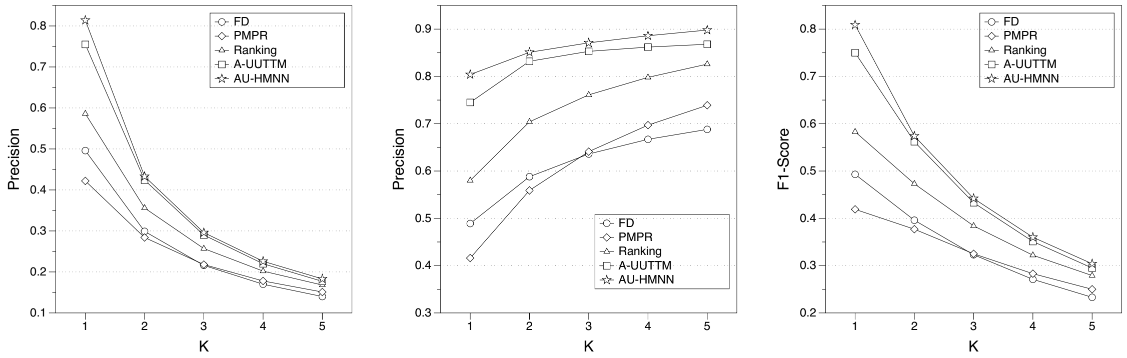

Figure 2: Precision, Recall, and F-Score with Different Number of Recommended Users

F-score of FD is 0.493 and its MRR is 0.745, which gives an indication of the difficulty of the task. However, because FD only recommends a consistent ranked list for each user, the effectiveness of the method would be very limited.

A comparison of the results of A-UUTTM and AU-HMNN shows that AU-HMNN achieves a significantly better result than A-UUTTM, which demonstrates that the neural network can also achieve a great performance on this task. The result of Ranking is worse than those of the other methods on this task, but it was also very effective.

Both U-MNN and U-HMNN are variants of our proposed method (AU-HMNN). U-MNN incorporates the user history interests using a normal one-layer attention encoder. In contrast to U-MNN, U-HMNN introduces the encoder architecture proposed in this paper with a hierarchical attention mechanism to model the user history interests. The comparison of U-MNN and U-HMNN shows that our proposed encoder architecture is more expressive and obtains a higher-quality user interest representation from a user history document. To investigate how information about the author's history interests affects the performance, we also show a comparison between AU-HMNN and U-HMNN. From the results of AU-HMNN and U-HMNN, we can observe that the author history information can significantly improve the performance.

In Figure 2, we list the precision, recall, and F-score of the different methods with various numbers of recommended users. The number of recommended users ranges from 1 to 5. Based on the results, we can see that the performance of AU-HMNN is the highest in all the curves, which demonstrates that our proposed model outperforms all the baseline methods in all metrics. When we recommend the top user for each tweet, we can obtain the best F-score, and if we want to obtain the highest recall, we can recommend more users for each tweet. In all the cases, the proposed method achieve the best performance.

3.4 Parameters and Efficiency Analysis

The proposed model contains several critical parameters. To analyze how these parameters influence the performance of our model, we designed a contrast experiment and list the results in Table 3.

The first parameter we evaluated is the number of hops, which we varied from 1 to 7 in this experiment. The results listed in the table show that the number of hops can influence the performance. With an increase in the number of hops, the results are better. The best performance was obtained when we trained the model with 6 hops. This indicates that multiple hops are important for a robust performance on this task. Surprisingly, the performance using 7 hops was not better than that using 6 hops, which indicated that the model can read, retrieve, and update the support memory successfully with 6 hops, and more hops cannot further improve the performance.

The second parameter is the embedding dimension. To investigate how it influenced the performance, we fixed the number of hops to 6 and tried different embedding dimensions. The comparison results listed in Table 3 show that the models with a high embedding dimension performed better than those with a low dimension. The results improved when the dimension was increased from 50 to 300, with similar results for dimensions equal to 300 and 400, which indicated that the expression ability is an important factor in this task. This showed that if we want to give more effective sugges-

Table 3: Performances of the Proposed Model with Different Parameters

Model	Embedding Dim.	# of Hops	Precision	Recall	F-Score	MRR
AU-HMNN	300	1	0.795	0.785	0.790	0.840
	300	2	0.799	0.789	0.794	0.844
	300	3	0.803	0.793	0.798	0.847
	300	4	0.807	0.797	0.803	0.850
	300	5	0.810	0.800	0.805	0.853
	50	6	0.746	0.737	0.741	0.804
	100	6	0.783	0.774	0.779	0.833
	200	6	0.808	0.798	0.803	0.850
	300	6	**0.814**	**0.804**	**0.809**	**0.857**
	400	6	0.813	0.803	0.808	0.855
	300	7	0.806	0.796	0.801	0.848

tions, a high dimension will be a good choice.

To meet practical efficiency requirements, the list of recommendations must be shown quickly before the users have to wait too long. Actually, the major time cost comes from the training cost of our proposed model. In practice, the training procedure is offline and won't impact the user's experience. Thus, the online computational cost mainly comes from the recommendation procedure. To investigate this performance issue, we recorded the time cost of our proposed model on a server with an Nvidia TITAN X graphic card. The average time cost of recommendation for our model with 6 hops and 300 dimensions is approximately 0.031 seconds, calculated from total cost of 849.511 seconds for 26,653 test instances, which demonstrates that our proposed model is efficient.

4 Related Work

One major area related to this work is recommendation tasks on social media. Previous works have studied a variety of recommendation problems on social media from different aspects, such as personalized tweet recommendation [Uysal and Croft, 2011; Yan *et al.*, 2012], hashtag recommendation [Sedhai and Sun, 2014; Gong and Zhang, 2016], and mention related recommendation [Wang *et al.*, 2013; Gong *et al.*, 2015]. The mentioned recommendation task have been studied from different aspects. The work of whom-to-mention [Wang *et al.*, 2013] want to mention the users who can fastly spread the tweets, which proposed a recommendation scheme with several manually constructed features related to a user interest match. Zhou et al. [2015] propose a personalized ranking model with consideration on multi-dimensional relations among users and mention tweets. Li et al. [2015] proposed a factor graph method to solve this recommendation problem. Instead of aiming at expanding diffusion of tweets, some works focus on targeting right person for mention behavior. A learning-to-rank based framework was proposed in [Tang *et al.*, 2015] with four categories of features to solve the recommendation task. Gong et al. [2015] treated this task as a translation problem and proposed a topical translation model incorporating the content of microblogs and users histories to perform the task.

Another major related area of this work is memory networks. Based on ideas of the attention mechanism and external memory, Sukhbaatar et al. [2015] proposed End-to-end Memory Networks to select explicit memories for query answering. The idea about memories have shown its effectiveness in many studies and leaded to significant improvements. Recently, variants of memory networks have also been studied and applied on various tasks, such as dialog systems [Dodge *et al.*, 2015; Weston, 2016], reading comprehension [Hill *et al.*, 2015; Weissenborn, 2016] and question answering [Weston *et al.*, 2015a; Kumar *et al.*, 2016; Miller *et al.*, 2016].

Motivated by the the memory networks, we proposed a novel neural architecture based on memory networks and applied it on mention recommendation task. We incorporated the interests of users and the interests of author with external memory and introduced a hierarchical attention mechanism for better performance.

5 Conclusion

Along with the dramatically increasing of social medias and requirement of improving the usability of user experience on mention action, in this work, we investigated the problem of recommending usernames for mention action in twitter-like social media. Previous works show that the interests of author and candidates users can provide valuable information for this task. To incorporate these information, external memory with a hierarchical attention mechanism was applied to capture these information. To demonstrate the effectiveness of the proposed method, we evaluated the proposed method on a large scale dataset collected from Twitter. The experimental results show that the proposed method can achieve better performance than state-of-the-art approaches in most cases. Since the focus of this work is the similarity between the interest of users and the given tweet, we did not consider the social relationship which is also valuable for this problem. Thus, how to incorporate this kind of information may be an interesting question in the future work.

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

[Chen and Manning, 2014] Danqi Chen and Christopher D Manning. A fast and accurate dependency parser using neural networks. In *EMNLP*, pages 740–750, 2014.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537, 2011.

[Dodge *et al.*, 2015] Jesse Dodge, Andreea Gane, Xiang Zhang, Antoine Bordes, Sumit Chopra, Alexander Miller, Arthur Szlam, and Jason Weston. Evaluating prerequisite qualities for learning end-to-end dialog systems. *CoRR*, abs/1511.06931, 2015.

[Gong and Zhang, 2016] Yuyun Gong and Qi Zhang. Hashtag recommendation using attention-based convolutional neural network. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 2782–2788, 2016.

[Gong *et al.*, 2015] Yeyun Gong, Qi Zhang, Xuyang Sun, and Xuanjing Huang. Who will you "@"? In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, CIKM '15, pages 533–542, New York, NY, USA, 2015. ACM.

[Hill *et al.*, 2015] Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. The goldilocks principle: Reading children's books with explicit memory representations. *CoRR*, abs/1511.02301, 2015.

[Kumar *et al.*, 2016] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: Dynamic memory networks for natural language processing. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1378–1387, 2016.

[LeCun *et al.*, 2015] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[Levine *et al.*, 2016] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

[Li *et al.*, 2015] Quanle Li, Dandan Song, Lejian Liao, and Li Liu. Personalized mention probabilistic ranking–recommendation on mention behavior of heterogeneous social network. In *International Conference on Web-Age Information Management*, pages 41–52. Springer, 2015.

[Miller *et al.*, 2016] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. Key-value memory networks for directly reading documents. *CoRR*, abs/1606.03126, 2016.

[Sedhai and Sun, 2014] Surendra Sedhai and Aixin Sun. Hashtag recommendation for hyperlinked tweets. In *The 37th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '14, Gold Coast , QLD, Australia - July 06 - 11, 2014*, pages 831–834, 2014.

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

[Tang *et al.*, 2015] Liyang Tang, Zhiwei Ni, Hui Xiong, and Hengshu Zhu. Locating targets through mention in twitter. *World Wide Web*, 18(4):1019–1049, 2015.

[Uysal and Croft, 2011] Ibrahim Uysal and W. Bruce Croft. User oriented tweet ranking: a filtering approach to microblogs. In *Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011*, pages 2261–2264, 2011.

[Wang *et al.*, 2013] Beidou Wang, Can Wang, Jiajun Bu, Chun Chen, Wei Vivian Zhang, Deng Cai, and Xiaofei He. Whom to mention: Expand the diffusion of tweets by @ recommendation on micro-blogging systems. In *Proceedings of the 22Nd International Conference on World Wide Web*, WWW '13, pages 1331–1340, New York, NY, USA, 2013. ACM.

[Weissenborn, 2016] Dirk Weissenborn. Separating answers from queries for neural reading comprehension. *CoRR*, abs/1607.03316, 2016.

[Weston *et al.*, 2015a] Jason Weston, Antoine Bordes, Sumit Chopra, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698, 2015.

[Weston *et al.*, 2015b] Jason Weston, Sumit Chopra, and Antoine Bordes. Memory networks. In *Proceedings of the International Conference on Learning Representations*, 2015.

[Weston, 2016] Jason Weston. Dialog-based language learning. *CoRR*, abs/1604.06045, 2016.

[Yan *et al.*, 2012] Rui Yan, Mirella Lapata, and Xiaoming Li. Tweet recommendation with graph co-ranking. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 516–525, 2012.

[Zeng *et al.*, 2014] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. In *COLING*, pages 2335–2344, 2014.

[Zhou *et al.*, 2015] Ge Zhou, Lu Yu, Chu-Xu Zhang, Chuang Liu, Zi-Ke Zhang, and Jianlin Zhang. A novel approach for generating personalized mention list on microblogging system. In *Data Mining Workshop (ICDMW), 2015 IEEE International Conference on*, pages 1368–1374. IEEE, 2015.