Weakly Supervised Attention for Hashtag Recommendation using Graph Data

Amin Javari[1] Zhankui He[2] Zijie Huang[3] Raj Jeetu[1]
Kevin Chen-Chuan Chang[1]

[1]University of Illinois at Urbana-Champaign

[2]University of California San Diego

[3]University of California Los Angeles

April 22, 2020

Hashtag recommendation

- Overwhelming amount of microblogs are generated on websites like Twitter and Weibo.
- Hashtags are profoundly effective for labeling microblogs.
- However, a huge and dynamic pool of hashtags exists.
- Two perspectives of addressing the problem:
  - Hashtag recommendations for microbloggs
  - Hashtag recommendation for users.

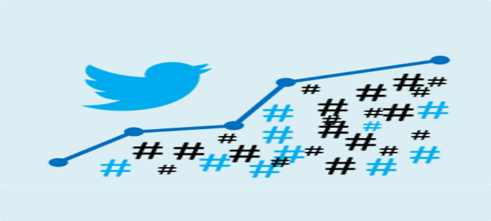


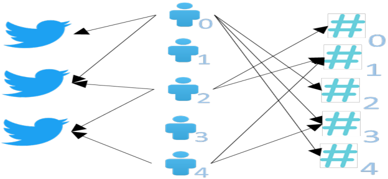
Image: Twitter.com

Hashtag Recommendation for Users

How to build a user profile?

- Content based recommenders.
- Collaborative filtering based recommenders.

Only a small portion of users tend to generate content data.



Problem: How to recommend hashtags to a given user, in particular in the absence of sufficient content data?

Hashtag recommendation based on graph data

The problem can be approached by relying on graph data.

- Link data is more abundant than content data.
- A large portion of links can be regarded as interest-based links

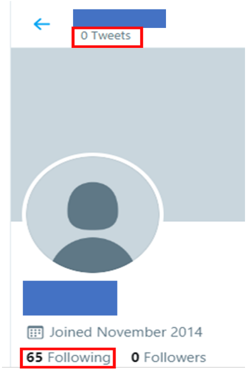
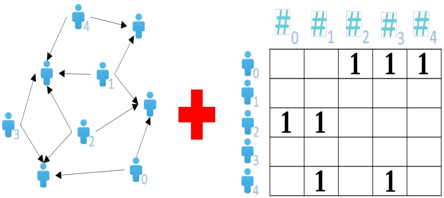


Image: Twitter.com

Problem formulation

Inputs:

- $G(V,E)$: The user's directed connections
- $\mathbf{Y} \in \mathbb{R}^{M \times N}$: User-hashtag interaction matrix between the target set of users $U \subseteq V$ and the set of hashtags H



Task: Estimate the scores of unobserved entries in matrix **Y**.

PHAN: Model structure

Generalized Matrix Factorization (GMF) is used for relevance prediction.

$$\hat{y}^{uh} = \sigma(\mathbf{W}^{\mathbf{T}}(\mathbf{h}^{\mathbf{h}} \odot \mathbf{u}^{uh}) + b)$$

- Hashtag embedding
- User embedding

Embedding hashtags

- Hashtags are not associated with any extra data but identities.
- Build a lookup table to embed hashtag.

How to embed users?

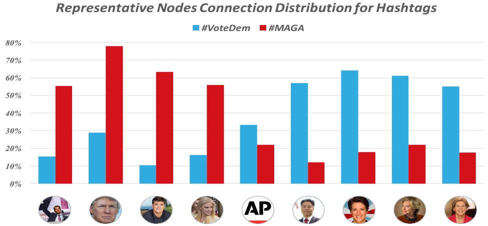
A trivial approach for user profiling using graph data

Profile a set of users based on their content/interaction data and then propagating their embeddings in the network.

- **Efficiency:** It is computationally expensive to propagate profiles in multiple iterations to a huge network like Twitter.
- **Effectiveness:** Such models build a fixed profile for each user.

Main idea: User profiling based on hub nodes

Key observation: Scale-freenes of microblogging networks.



Main Idea: Profile users based on the profiles of hub nodes.

- Hubness of a user indicates that the user represents a topic of interest.
- Hub nodes are associated with ample data.
- Hub nodes have low dynamic.

Selecting representative nodes

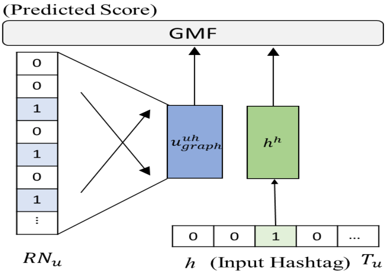
Hub nodes as user features:

- Given U and $G = (V, E)$, the set of representative nodes $RN = \{r_1, r_2, ..., r_n\}$ is defined as nodes in V with at least L followers in U.
- RN_u is the representative followees of the user u
- L is decided by the size of U.

How to embed users based on hub nodes?

Design 1: Fixed embedding

Project the one-hot encoded vector to a fully connected dense embedding vector.

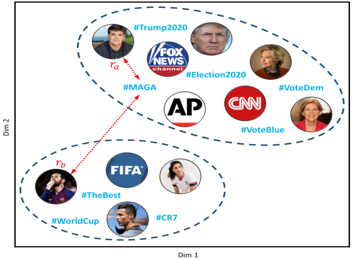


- This trivial approach is oblivious to the target hashtag information.

Design 2: Embedding based on classic attention

Hashtags and representative nodes are projected into a shared latent space.

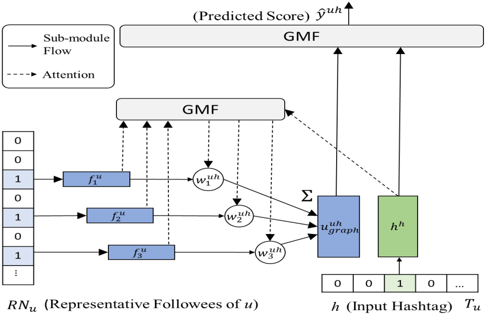
- If a user representative node r and uses hashtag f, it indicates proximity between r and f.



The embeddings of representative followees are aggregated in a weighted way.

- Weights are determined based on the relevance of the representative nodes to the target hashtag.

Design 2: Embedding based on classic attention (continued)



Relevance score

$$\mathbf{u}^{uh}_{graph} = n^{-\beta} \sum_{j \leq n} \hat{w}^{uh}_j \cdot \mathbf{f}^{\mathbf{u}}_{\mathbf{j}}$$

Representative node embedding

$$\hat{w}^{uh}_j = \sigma(\mathbf{W}^{\mathbf{T}}(\mathbf{h}^{\mathbf{h}} \odot \mathbf{f}^{\mathbf{u}}_{\mathbf{j}}) + b).$$

Why classic attention does not generate accurate attention maps?

- The accuracy of the prediction model depends on the effectiveness of the attention model.
- Classically, the attention mechanism can be trained based on the final objective of the model.
- It is not quite effective for models with high complexity.
  - Overfitting issue
  - Vanishing gradient problem

Design 3: Embedding based on weakly supervised attention

Idea: Add supervision to the attention model.

Challenge: The labels of relevance for representative nodes and hashtags do not exist.

weak supervision: Generate weak labels for the relevance of nodes and hashtags by relying on statistical methods.

- **Co-occurrence based label**

$$w_j^{uh} = P(j|h) = \frac{\overbrace{c(h,j)}^{\text{Co-occurrence count of hashtag } h \text{ and node } j}}{\underbrace{c(h)}_{\text{Occurrence count of hashtag } h}}$$

- **Informativeness based label**

$$w_j^{uh} = \frac{c(h,j)}{c(h)} * \log \frac{\overbrace{c(j)}^{\text{Occurrence count of node } j}}{\underbrace{c(h',j)}_{\text{Occurrence count of node } j \text{ in absence of hashtag } h}}$$

Loss function

$$L_{main}(\hat{y}^{uh}, y^{uh})$$

$$
\begin{aligned}
Loss = & -(\hat{y}^{uh} \log y^{uh} + (1 - \hat{y}^{uh}) \log(1 - y^{uh})) \\
& - \lambda \; \frac{1}{n} \sum_{j \leq n} \hat{w}_j^{uh} \log w_j^{uh} + (1 - \hat{w}_j^{uh}) \log(1 - w_j^{uh})
\end{aligned}
$$

$$L_{attention}(\hat{W}^{uh}, W^{uh})$$

The joint loss function consist of two parts:

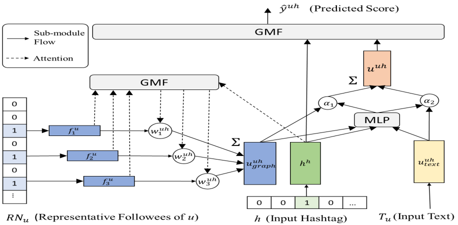
- The main loss function
- The attention based loss function

λ is a trade-off setting.

Fusing content and graph data

Task: Find the relevance score between a hashtag h and a pair of text-followee list $< T_u, RN_u >$.

Encode text and followee data separately and then fuse them.



- Microblog-hashtag recommendation (MHR)
  - LSTM[1] is used to encode the single input microblog
- User-hashtag recommendation (UHR)
  - The memory model[2] is used to encode multiple microblogs.

[1] Sepp Hochreiter et al. "Long Short-Term Memory". In: *Neural Computation* (1997).

[2] Haoran Huang et al. "Hashtag Recommendation Using End-To-End Memory Networks with Hierarchical Attention". In: *COLING*. 2016.

Experimental setting

Datasets

- Twitter[3]
- Weibo[4]

Statistics of the datasets.

Dataset	**#Tweet**	**#Followees**	**#RN**	**#Hashtag**	**#User**	Avg(**#RN**$_{user}$)
Twitter	217965	253818	9309	2873	23169	33.42
Weibo	10521	585475	8426	2017	7023	121.98

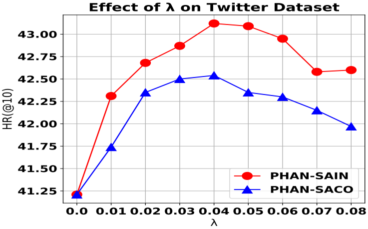
Metrics

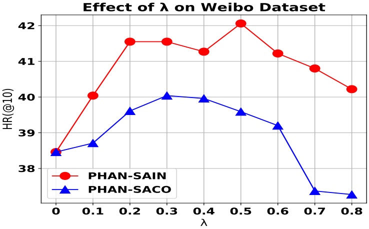
- Hit Ratio (HR@K)
- Normalized Discounted Cumulative Gain (NDCG@K)

[3] **https://www.kaggle.com/hwassner/TwitterFriends/home**.

[4] **https://www.aminer.cn/weibo-net-tweet**.

Experiments: How effective is the proposed attention model?





The model's performance as a function of λ.

- Supervision improves the accuracy of the attention model
- Higher values of λ misled the model's objective function

Experiments: Performance of the proposed model for user-hashtag recommendation

	Twitter		Weibo	
Model	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**
LDAR[5]	18.74%	7.97%	15.96%	6.87%
LSTMR[6]	23.14%	11.59%	39.47%	27.22%
MF[7]	26.30%	14.98%	23.26%	12.29%
MNR[8]	28.29%	17.00%	30.92%	20.75%
PHAN-FIXD	41.21%	24.33%	38.43%	24.78%
PHAN-CA	41.35%	24.25%	38.46%	24.83%
PHAN-SACO	42.79%	26.20%	40.45%	26.48%
PHAN-SAIN	**43.73%**	**26.33%**	**42.06%**	**27.94%**

- Representative nodes are informative for user profiling.
- PHAN with supervised attention outperforms the other variants of PHAN.

[5] Ralf Krestel et al. "Latent Dirichlet Allocation for Tag Recommendation". In: *RecSys*. 2009.

[6] Sepp Hochreiter et al. "Long Short-Term Memory". In: *Neural Computation* (1997).

[7] Hamidreza Alvari. "Twitter hashtag recommendation using matrix factorization". In: *arXiv* (2017).

[8] Haoran Huang et al. "Hashtag Recommendation Using End-To-End Memory Networks with Hierarchical Attention". In: *COLING*. 2016.

Experiments: Performance of the proposed model for microblog-hashtag recommendation

	Twitter		Weibo	
Model	**HR@10**	**NDCG@10**	**HR@10**	**NDCG@10**
LDAR	43.79%	22.03%	34.06%	27.43%
LSTMR	56.21%	38.31%	52.43%	35.47%
EmTaggeR[9]	62.47%	49.69%	57.63%	44.09%
CNNAR[10]	63.64%	49.48%	55.95%	43.71%
PHAN-FIXD	74.65%	55.76%	61.58%	51.40%
PHAN-CA	73.46%	53.79%	61.70%	51.42%
PHAN-SACO	74.64%	55.27%	63.41%	51.54%
PHAN-SAIN	**77.57%**	**58.54%**	**65.31%**	**52.94%**

[9] K. Dey et al. “EmTaggeR: A Word Embedding Based Novel Method for Hashtag Recommendation on Twitter”. In: *ICDMW*. 2017.

[10] Yuyun Gong and Qi Zhang. “Hashtag Recommendation Using Attention-based Convolutional Neural Network”. In: *IJCAI*. 2016.

Thank you!