DeepTagRec: A Content-cum-User based Tag Recommendation Framework for Stack Overflow

Suman Kalyan Maity[1], Abhishek Panigrahi[2], Sayan Ghosh[2], Arundhati Banerjee[2], Pawan Goyal[2] and Animesh Mukherjee[2]

[1]Northwestern University; [2]Dept. of CSE, IIT Kharagpur, India*
*Complex Networks Research Group (CNeRG) *http://www.cnergres.iitkgp.ac.in/*

1 Objective

To develop a tag recommendation system that draws inference not only from the content of the question, but also utilizes the latent relationships between users and the tags they frequently use on community based question answering (CQA) websites.

2 Example Question-Tag Recommendation

Title: Implementing Custom Codec with JSR-135 MMAPI
Body: Is it possible to implement a custom codec decoder and play a media file using JSR135? Actually I am interested in playing back a custom media file in J2ME. I think JSR135 is a good starting point but I am open to other suggestions. Thanks.
Expected Set of Tags: algorithm, image-processing, c#
DeepTagRec$_{content}$: osx, image-processing, performance, image, android, iphone, java, objective-c, compression, ios
DeepTagRec: algorithm, image-processing, performance, wpf, image, html, c#, java, .net, graphics

3 Quick Review

- Text encoding - Gated Recurrent Unit (GRU)
- Word embedding - Word2Vec
- Network node embedding - Node2Vec
- The best performing baseline architecture **TagCombine** [4] mainly considers the text information and does not account for features from usage patterns.

4 Dataset

We have **0.5 million** training questions from Stack Overflow. Each question has different number of tags associated with it. For testing the model, we use 10K questions. The output of the model is a probability distribution over **38,196 tags**. We take the k tags with the highest probability for further evaluation.

- **Pre-processing**: Maximum length of the question is fixed as 300 words. Each word is represented as a 300(m) dimension vector by using the predefined word2vec embeddings.
- The tags are represented as one-hot vectors. For a training example with t tags we add these t one hot vectors as the output for that training example.

5 Proposed Framework

- Content representation extraction from question title and body using GRU model to encode the content as a sequence of words, concatenating the title and body. Given the title of the question T and main body of the question B, we first run a GRU to learn representation of T, denoted by c_T. In the next step, we learn the representation of B denoted by c_B using a GRU, having c_T as the initial hidden state.

$$c_T = G(T, 0); c_B = G(B, c_T) \quad (1)$$

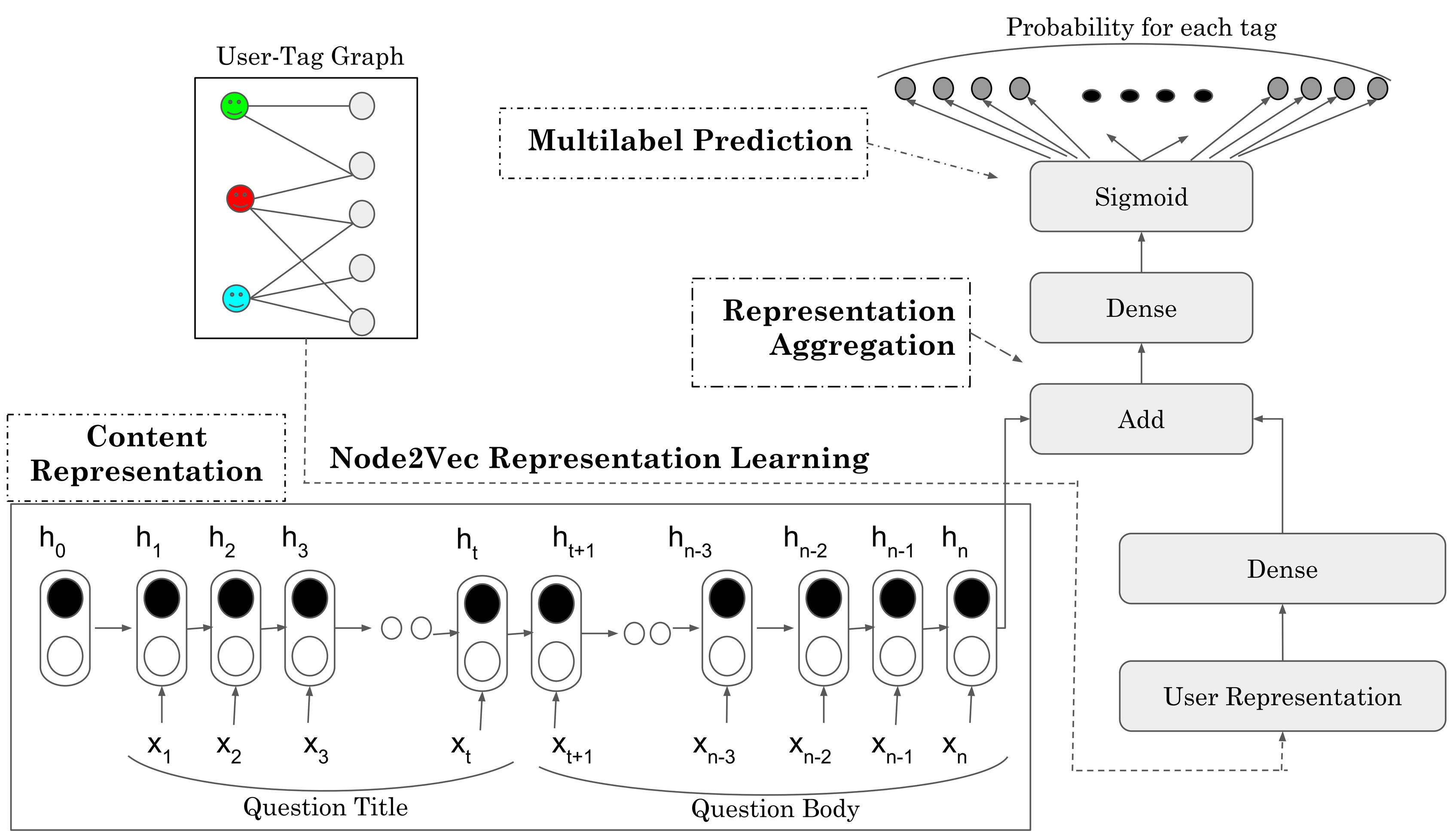

Figure 1: DeepTagRec framework

- Learning user representation from heterogeneous user-tag network using node2vec motivated by the idea that *a user's representation should have high similarity with his/her adjacent tag nodes* in the graph. An **edge exists** between a user node and a tag node in the network if the user has posted some question with the tag present in the question's tagset.
- Tag prediction as a *multilabel prediction* task using *representation aggregation* from both the word2vec representation (Q_w) of the question data and the node2vec representation (U_n) of users.

6 Results

Model	P@3	P@5	P@10	R@3	R@5	R@10
Krestel et al. [2009]	0.0707	0.0603	0.0476	0.0766	0.1097	0.1738
Lu et al.[2009]	0.1767	0.1351	0.0922	0.1952	0.2477	0.3362
Wu et al. [2016]	0.21	0.16	0.106	0.2325	0.2962	0.3788
#TAGSPACE [2014]	0.105	0.087	0.063	0.111	0.162	0.511
fastText [2016]	0.102	0.0783	0.149	0.0388	0.149	0.227
TagCombine	0.3194	0.2422	0.1535	0.3587	0.4460	0.5565
DeepTagRec$_{content}$	0.4442	0.3183	0.184	0.5076	0.591	0.6702
DeepTagRec	**0.5135**	**0.3684**	**0.2125**	**0.5792**	**0.6736**	**0.7613**

Table 1: Precision (P) and Recall (R) in for *DeepTagRec* and the other baselines.

- **Top-k accuracy** : The fraction of questions correctly annotated by at least one of the *top-k* tags recommended by the algorithm.
- **Exact-k accuracy**: The fraction of questions correctly annotated by the k^{th} recommended tag.

Model	$k=3$	$k=5$	$k=10$	$k=1$	$k=2$	$k=3$	$k=4$	$k=5$
TagCombine	0.688	0.769	0.851	0.481	0.289	0.188	0.145	0.108
DeepTagRec	**0.916**	**0.945**	**0.966**	**0.784**	**0.468**	**0.289**	**0.184**	**0.118**

Table 2: Top-k (first 3 columns) and exact-k accuracy.

7 Conclusion

DeepTagRec leverages both the textual content (i.e., title and body) of the questions and the user-tag network for recommending tags. It thus significantly outperforms the most competitive baseline that does not account for the user tag usage pattern.

- We improve – *precision*@3 by 60.8%, *precision*@5 by 52.1%, *precision*@10 by 38.4%, *recall*@3 by 61.5%, *recall*@5 by 51.03%, *recall*@10 by 36.8% – over *TagCombine*.
- *DeepTagRec* also performs better in terms of other metrices where it achieves 63% and 33.14% overall improvement in *exact-k accuracy* and *top-k accuracy* respectively over *TagCombine*.

8 References

[1] Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016)

[2] Krestel, R., Fankhauser, P., Nejdl, W.: Latent dirichlet allocation for tag recommendation. In: RecSys. pp. 61–68 (2009)

[3] Lu, Y.T., Yu, S.I., Chang, T.C., Hsu, J.Y.j.: A content-based method to enhance tag recommendation. In: IJCAI. vol. 9, pp. 2064–2069 (2009)

[4] Wang, X.Y., Xia, X., Lo, D.: Tagcombine: Recommending tags to contents in software information sites. J. Comput Sci. **30**(5), 1017–1035 (2015)

[5] Weston, J., Chopra, S., Adams, K.: #tagspace: Semantic embeddings from hashtags. In: EMNLP. pp. 1822–1827 (2014)

[6] Wu, Y., Wu, W., Li, Z., Zhou, M.: Improving recommendation of tail tags for questions in community question answering. In: AAAI (2016)