Co-Attentive Multi-Task Learning for Explainable Recommendation: Supplementary Materials

1 Implementation Details

1.1 Hyper-parameter Settings

The most important parameters are listed at the **Implementation details** in the paper. We here describe detailed parameters in our implementations. For CF models (PMF, NMF and SVD++), the number of latent factors is set to 50. For NARRE, RLRec and NRT, we keep the same settings proposed by the authors. For MPCN and our model CAML, we reuse the most of hyper-parameters in MPCN. The word-embedding size and dimensions of hidden projection layer are set to 50. The maximum length of all reviews is also set to 600 (20 reviews of 30 words each). The number of factors in FM is set to 10. The hidden units of the RNN decoder is set to 400 which are the same as NRT, the maximum length of explanations is 30. We set the beam size to 4 and use the average log-likelihood (dividing by the sequence length) to encourage generating longer explanations. The maximum for the training epochs is 30 and models are stopped early if their performance drops in 5 epochs.

1.2 Implementation of the Baseline Lexrank

Similar with [Li *et al.*, 2017], we first collect all the reviews $\mathcal{D}_u$ and $\mathcal{D}_v$ for each user u and item v. We filter the less related reviews in $\mathcal{D}_u$ and $\mathcal{D}_v$ by considering: 1) the difference between their ratings and the ground truth rating $r_{u,v}$; 2) the unique words which are not appeared in any other set of reviews. The remaining reviews in $\mathcal{D}_u$ and $\mathcal{D}_v$ are split into sentences and fed into the Lexrank to extract a sentence as the final explanation.

2 Instructions on Running CAML

The source code of CAML is publicly available at https://github.com/3878anonymous/caml.
It can be run through following steps:

- **Step 1**. Download the Amazon dataset or Yelp dataset, pre-process the dataset and randomly choose 80%, 10% and 10% of the data as the training, validation and testing data respectively.
- **Step 2**. Utilize Microsoft Concept Graph[1] [Wu *et al.*, 2012; Wang *et al.*, 2015] and derive the corresponding concepts for each reviews in the dataset. Filter rarely used and domain-dependent frequent concepts and merge them into the vocabulary according to the description in the experiment section. All the data formats are in https://github.com/3878anonymous/caml/data/.
- **Step 3**. Use "train_CAML.py" to train our model. All the model parameters are defined in the "parser_CAML.py".
- **Step 4**. Use "test_CAML.py" to test the model on both rating prediction and explanation generation. This code calculates the RMSE of rating, the BLEU score of generated explanation and also output explanations in a folder for ROUGE evaluation.
- **Step 5**. Calculate ROUGE scores with the perl package "ROUGE-1.5.5" in the "evaluate" folder. The example calculating script is the "example_evaluation.sh".

[1] https://concept.research.microsoft.com/

3 Case Analysis of Diversity

To show the capability of our generation model, we share several cases with several representative explanations generated by our model in Table 1. Our model can not only generate explanations with different expressions of the same concept, but also obtain multiple diverse explanations which contain various appropriate concepts.

Case 1	If you are a **fan** of the 80's, you'll love this. Not the best of the old **horror** movies, but it's still a good one. Nice to see the old classic **horror** movies.
	I remember watching this film when I was a 8 years **kid**, I was so terrified, i didn't want to go to the bath alone!
Case 2	This movie is a total waste of **time**. Save your **money**. You know the movie is a **joke**.
	Embarassingly painful is what this crap.
Case 3	As a **fan** of the phantom of the opera, I was very excited to see this movie. If you are a **fan** of the phantom, you will love this movie. The **story** was a bit rushed to the end. What a great **cast**.
	As a **person** who really enjoyed phantom of the opera, I was expecting quite a bit from this **piece**.

Table 1: Examples of various explanations generated by CAML. The last line of each case is the ground truth explanation.

References

[Li *et al.*, 2017] Piji Li, Zihao Wang, Zhaochun Ren, Lidong Bing, and Wai Lam. Neural rating regression with abstractive tips generation for recommendation. In *Proceedings of the International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 345–354, 2017.

[Wang *et al.*, 2015] Zhongyuan Wang, Haixun Wang, Ji-Rong Wen, and Yanghua Xiao. An inference approach to basic level of categorization. In *CIKM*, pages 653–662. ACM, 2015.

[Wu *et al.*, 2012] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A probabilistic taxonomy for text understanding. In *SIGMOD*, pages 481–492. ACM, 2012.