Enhancing Collaborative Filtering Recommender with Prompt-Based Sentiment Analysis

Elliot Dang
New York University
yd1008@nyu.edu

Zheyuan Hu
New York University
zh2095@nyu.edu

Tong Li
New York University
tl2204@nyu.edu

Abstract

Collaborative Filtering(CF) recommender is a crucial application in the online market and e-commerce. However, CF recommender has been proven to suffer from persistent problems related to sparsity of the user rating that will further lead to a cold-start issue. Existing methods address the data sparsity issue by applying token-level sentiment analysis that translate text review into sentiment scores as a complement of the user rating. In this paper, we attempt to optimize the sentiment analysis with advanced NLP models including BERT and RoBERTa, and experiment on whether the CF recommender has been further enhanced. We build the recommenders on the Amazon US Reviews dataset, and tune the pretrained BERT and RoBERTa with the traditional fine-tuned paradigm as well as the new prompt-based learning paradigm. Experimental result shows that the recommender enhanced with the sentiment ratings predicted by the fine-tuned RoBERTa has the best performance, and achieved 30.7% overall gain by comparing MAP, NDCG and precision at K to the baseline recommender. Prompt-based learning paradigm, although superior to traditional fine-tune paradigm in pure sentiment analysis, fail to further improve the CF recommender.

1 Introduction

Collaborative filtering (CF) is a techniques widely used by recommender systems. As one of the two categories of CF, explicit CF exploit feedback such as numerical ratings of an existing user's community to predict which items the current user probably like most (Schafer et al., 2007). Explicit CF perform well as long as there is sufficient rating information. However, their effectiveness deteriorates if there exist insufficient ratings, which is known as data sparsity, and data sparsity would further lead to a cold start issue (Bobadilla et al., 2013). Meanwhile, the rating criteria differ for each user, and inconsistent rating criteria would weaken the reliability of the recommendation.

One possible way to reduce the data sparsity and inconsistent rating criteria is to integrate the text reviews into the recommender system, and apply sentiment analysis on the reviews (García-Cumbreras et al., 2013). The numerical scores decoded from the sentiment analysis can be used as a supplement to the ratings with a uniform standard, and then filled into an expanded user-item matrix to enhance the recommender.

Recently, prompt learning (pretrain - prompt - predict) as a new paradigm in Natural Language Processing(NLP), has shown its potential to outperform the fine-tune - pretrain paradigm, and has been proved successful in a wide range of NLP tasks, including sentiment analysis. (Liu et al., 2021).

In this paper, we investigated whether the state-of-the-art techniques in NLP would help with the sentiment analysis to further diminish the impact of data sparse issue. We developed three different recommenders, the baseline recommender fit on the original sparse dataset, the one enhanced by sentiment analysis with finetune-pretrain paradigm, and the one enhanced by sentiment analysis with prompt learning paradigm. We compared the above recommenders based on the evaluation metrics that measured both predictive accuracy as well as the user experience, including MAP at K, NDCG, and Precision at K.

2 Related Work

Early Solution One of the early solutions to overcome the cold-start recommendation issue is to exploit social tags of the users (Ghabayen and Noah). Tagging indeed has the potential of enhancing finding similar users, but it is also based on the naive assumption that users are interested with the items that were annotated.

Transition to sentiment analysis According to

(García-Cumbreras et al., 2013), a sentiment analysis approach can be applied to textual reviews in order to infer users' preferences and such preferences can be subsequently mapped into some numerical ratings that the CF algorithm relies on. Ricci (Ricci et al., 2015) also discussed the cold start challenge in recommender systems, and proposed that applying sentiment analysis to be one of the solutions.

Token level sentiment analysis In this case, Osman and Nurul (Osman et al., 2021) has developed a CF recommender system that integrated with sentiment analysis on text reviews as an enhancement. They designed an algorithm that translates text reviews into sentiment scores, and then supplement the missing ratings with the sentiment scores. However, the algorithm is simply based on the occurrence of positive and negative terms, so it only captures the sentiment at the token level, but fail to retrieve the contextual sentiment information at the sentence level.

Pretrained language model With the arrival of pretrained language models(PLMs) such as Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2018), Sentiment analysis from text data has undergone a colossal transformation. Shivaji and Manit (Alaparthi and Mishra, 2021) concluded that BERT has the undisputed superiority in sentiment classification from text, by running experiments on the IMDB movie review dataset and comparing the performance of the unsupervised SentiWordNet, Logistic Regression, LSTM, and Bert-Base pretrained on Wikipedia and Bookcorpus.

Prompt Learning Recently, prompt-based learning has been proved successful in multiple NLP tasks, including sentiment analysis (Liu et al., 2021). Prompting shortens the gap between the objectives for the pre-training stage and the objectives for the downstream tasks by designing a template that transforms the downstream inputs to a certain format, so that enables the input to fit the PLM properly.

3 Methodology

3.1 Data

The original dataset is the Amazon US Reviews dataset with the category "Video", which contains both ratings and textual reviews, and can be accessed through the HuggingFace API. The preprocessed dataset contains roughly 280K items, each represented by the following 4 columns required for running sentiment analysis and building CF recommender:

- customer_id: Random identifier represents customer who purchased the video.
- product_id: The unique Product ID the review pertains to.
- star_rating: The 1-5 numerical rating of the video.
- review_headline: The content of the textual review.

In order to obtain a sparse dataset to simulate the data sparsity issue, we randomly dropped 40% of the user rating, so the dataset contains only 60% of the true user rating. Further operations (e.g.create user-item matrix, obtain sentiment ratings) and experimentation are achieved with this preprocessed sparse dataset.

3.2 Explicit Collaborative Filtering

Collaborative Filtering (CF) model is widely used for RecSys, which usually takes a model-based approach. This approach strives to construct latent representations for User and Item and uses Alternating Least Squares (ALS) to learn the representations. ALS is based on the concept of Matrix Factorization and iteratively solves for the best lower-dimensional latent representations. Each rating, or explicit user-item interaction, represents the preference of the user to the item, where a higher rating usually suggests a higher preference. CF recommends items to users by constructing a expected score for the missing user-item interactions by using the latent representations, and the scores reflect the magnitude of confidence for the recommendation.

3.3 Fine-Tuned PLM

Masked language model (MLM) has been proved powerful for sentiment analysis and its generalizability for the downstream tasks has been guaranteed by previous studies as well. As a first attempt that applies both PLM and prompt learning to enhance the CF recommender, we consider BERT(Devlin et al., 2018) and RoBERTa (Liu et al., 2019) to be the best choices of the PLM, since they are both representative MLMs, and are feasible enough for the prompt-based learning paradigm. Specifically, we decide to use the base version for both BERT and RoBERTa, that are pretrained on the Wikipedia and Bookcorpus dataset, and contain 110M and 125M parameters respectively, due to the performance-runtime trade-off.

Since sentiment analysis is basically a sequence classification task, where the numeraical ratings from 1-5 can be considered as 5 labels, we add an multilayer perceptron(MLP) as the classification head of the PLMs that maps the last hidden layer to the label space with the softmax activation function. We then fine tune the PLMs on the 60% of the dataset where the rating exist, with the embedded textual reviews as the input and corresponding numerical ratings as the target label. Once we obtain the fine-tuned PLMs, we complement the 40% whose rating has been dropped with the sentiment scores predicted by the fine-tuned PLMs. Eventually, we can build CF recommenders that are enhanced by sentiment analysis with the fine-tuned PLMs with 60% of the true user rating and 40% of the sentiment rating.

3.4 Prompt Learning

We implement manual prompt design by basically following the pipeline proposed by OpenPrompt (Ding et al., 2021), which is a standard, and flexible framework. Figure 1 illustrates the overall workflow of OpenPrompt. In order to construct the PromptModel and PromptDataset, we first define the following object:

- PLMs - The backbone model for the prompt-based learning. Here we choose BERT base and RoBERTa base with a sequence classification head on top.

- Verbalizer: A verbalizer class maps original labels to label words in the vocabulary. In this case, we have 5 labels representing the numerical ratings,so the verbalizer will map {1,2,3,4,5}to the word vocabulary {awful, bad, fair, good, wonderful}, respectively.

- Template - A modifier that converts the input text into certain format. Here the format is designed to be "Overall, it was a [x] movie", and the verbalizer is defined as above. For example, if we have a user review "I love this movie" with a 4 star rating, the Template would then replace "[x]" with the word "good" mapped from the numerical rating 4, that is, the review will be converted to "Overall, it was a good movie" by the Template.

Once the PLM, Template, and Verbalizer has been well defined, the text reviews in the original dataset will be modified by the template and then be tokenized with the tokenizer pretrained by the PLM to obtain the PromptDataset. Meanwhile, the PromptModel object that practically participates in training and inference will be constructed by combining the Verbalizer, the PLM, and the Template together. The Trainer module will be responsible for tuning the PLM and prompts simultaneously with the same training configuration as the pretrain-finetune paradigm.

Similarly, we train on 60% of the dataset where the rating exists and predict the prompt-based sentiment rating for the remaining 40% where rating has been dropped.

4 Experiments

4.1 Experiments Setup

We plan to compare the RecSys performance trained with five different datasets: **SPARSE** the baseline, **SENT-BERT** that uses BERT, **SENT-ROBERTA** that uses RoBERTa, **SENT-PROMPT-BERT** that incorporates BERT with Prompt Learning, and finally **SENT-PROMPT-ROBERTA** that combines RoBERTa with Prompt Learning. The schemes used to create these datasets are explained in Section 3. We use the same validation set the measure the performance.

4.2 Evaluation Results

Model	Accuracy	F1
BERT	0.7418	0.7268
RoBERTa	0.7662	0.7551
BERT-Prompt	0.7600	0.7641
RoBERTa-Prompt	**0.7832**	**0.7658**

Table 1: Model performance on 5-label sentiment classification

The four PLMs have been tuned to achieve the accuracy and F1 score for sentiment classification as shown in Table 1. RoBERTa outperforms BERT in both accuracy and F1 as expected. Meanwhile, prompt-learning paradigm has demonstrated its superiority over the traditional finetune-pretrain paradigm for both BERT and RoBERTa. We are then interested in whether the CF recommender further benifits from prompt-learning.

RecSys	MAP	NDCG@30	P@30	Avg. Imp%
SPARSE	0.4598	0.4906	0.0304	—
SENT-BERT	0.4854	0.5409	**0.0518**	28.74%
SENT-ROBERTA	**0.5999**	**0.6403**	0.0399	**30.74%**
SENT-PROMPT-BERT	0.5832	0.6258	0.0392	27.78%
SENT-PROMPT-ROBERTA	0.5770	0.6196	0.0389	26.58%

Table 2: Evaluation results for recommenders

We adopt Spark.MLlib to train an Alternating Least Squares (ALS) for our Collaborative Filtering RecSys with explicit feedback (rating). We evaluate our recommendation performance on three different metrics: Mean Average Precision (MAP), Normalized Discounted Cumulative Gain at K (NDCG@K), and Average Precision at K (P@K). These are commonly used metrics for recommendation systems, which compute metrics between the ranked list of recommendation, and the set of group truth items (items that the user actually consume), where ranking is based on the score produced by the RecSys.

Suppose we have M users, and $U = \{u_1, u_2, ..., u_M\}$ be the set of all users. For each user u_i, it has a set of N_i ground truth items $D_i = \{d_1, d_2, ..., d_{N_i}\}$. And also a ranked list containing Q_i recommended items $R_i = [r_1, r_2, ..., r_{Q_i}]$. Define a relevance function $rel_D(r) = 1 \ if \ r \in D, 0$ otherwise, and let $n = \min(\max(Q_i, N_i), K)$, $IDCG(D, K) = \sum_{j=1}^{min(|D|,K)} \frac{1}{\log(j+1)}$.The metrics are defined as follows:

- $P@K = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{K} \sum_{j=1}^{\min(Q_i,K)} rel_{D_i}(R_i(j))$
- $MAP = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{N_i} \sum_{j=1}^{Q_i} \frac{rel_{D_i}(R_i(j))}{j}$
- $NDCG@K = \frac{1}{M} \sum_{i=1}^{M} \frac{\sum_{j=1}^{n} \frac{rel_{D_i}(R_i(j))}{\log(j+1)}}{IDCG(D_i,K)}$

P@K measures the fraction of first K recommended items that the user did actually consume averaged on users, MAP measures the fraction of recommended items on the set of true relevant items, and NDCG@K measures similar to MAP but takes the rank of the recommendation into account.

Here we choose K = 30. The results are shown in Table 2. We find noticeable increases over the **SPARSE** baseline for all other four datasets enhanced with fine-tuning and/or prompt learning. We compute the Average Percent Improvements (Avg. Imp%) for all three metrics over the baseline. Overall, **SENT-ROBERTA** achieves the highest performance in MAP and NDCG@30 (0.5999 and 0.6403 respectively), and has the highest average % increase (30.74%). **SENT-BERT** has the highest P@30 score (0.0518), while having the lowest MAP and NDCG@30 (0.4854 and 0.5409 respectively) among the enhanced RecSys. After incorporating Prompt Learning, **SENT-PROMPT-BERT** improves significantly over **SENT-BERT** in MAP and NDCG@30 scores (from 0.4854 to 0.5832 and 0.5409 to 0.6258 respectively) while its P@30 score decreases (from 0.0518 to 0.0392). However, **SENT-PROMPT-ROBERTA**'s performance drops when compared to **SENT-ROBERTA** on all three metrics. Its MAP score decreases from 0.5999 to 0.577, NDCG@30 from 0.6403 to 0.6196, and P@30 from 0.0399 to 0.0389. One potential explanation for this is that the Collaborative Filtering may not at its pure optimal configuration and thus is underfitting the specific **SENT-PROMPT-ROBERTA** dataset. However, from the results above, we do find that enhanced RecSys improves the baseline RecSys noticeably, indicating the enhancements are working pretty well on the RecSys.

5 Conclusion

In this project, we demonstrate that utilizing advanced pretrained models such as BERT and RoBERTa to predict sentiment rating based on text reviews greatly address the data sparsity issue in CF recommender. Prompt-based learning paradigm, although superior to traditional fine-tune paradigm in sentiment analysis, shows no advantage to further improve the CF recommender. We believe this conclusion needs to be tested on additional datasets in the future to avoid the serendipity of a single dataset. Besides, potential future work include optimizing the prompt design(e.g. the format of the template), and attempting on different prompt methods such as soft prompt.

6 Collaboration Statement

Elliot Dang - Prompt Learning + Report Writing
Zheyuan Hu - Data Processing + Model Fine-Tuning + Experiment Design + Report Writing
Tong Li - Recommender System + Experimenting + Report Writing

7 GitHub Link

https://github.com/hhhhzy/nlu_project

References

Shivaji Alaparthi and Manit Mishra. 2021. Bert: a sentiment analysis odyssey. *Journal of Marketing Analytics*, 9(2):118–126.

J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. 2013. Recommender systems survey. *Knowledge-Based Systems*, 46:109–132.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ning Ding, Shengding Hu, Weilin Zhao, Yulin Chen, Zhiyuan Liu, Hai-Tao Zheng, and Maosong Sun. 2021. Openprompt: An open-source framework for prompt-learning. *CoRR*, abs/2111.01998.

Miguel Á. García-Cumbreras, Arturo Montejo-Ráez, and Manuel C. Díaz-Galiano. 2013. Pessimists and optimists: Improving collaborative filtering through sentiment analysis. *Expert Systems with Applications*, 40(17):6758–6765.

A. S. Ghabayen and Shahrul Azman Noah. Exploiting social tags to overcome cold start recommendation problem.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *CoRR*, abs/2107.13586.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Nurul Aida Osman, Shahrul Azman Mohd Noah, Mohammad Darwich, and Masnizah Mohd. 2021. Integrating contextual sentiment analysis in collaborative recommender systems. *PLOS ONE*, 16(3):1–21.

Francesco Ricci, Lior Rokach, and Bracha Shapira. 2015. *Recommender Systems: Introduction and Challenges*, pages 1–34. Springer US, Boston, MA.

Bin Fu Peter Dolog Zhihai Wang Martin Leginus Rong Pan, Guandong Xu. 2012. Improving Recommendations by the Clustering of Tag Neighbours. *JoC*, 3:13–20.

J. Ben Schafer, Dan Frankowski, Jon Herlocker, and Shilad Sen. 2007. *Collaborative Filtering Recommender Systems*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg.

Xiaoyuan Su and Taghi M. Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009:421425.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2019. BERT post-training for review reading comprehension and aspect-based sentiment analysis. *CoRR*, abs/1904.02232.

A Appendix

A.1 Prompt Learning Workflow

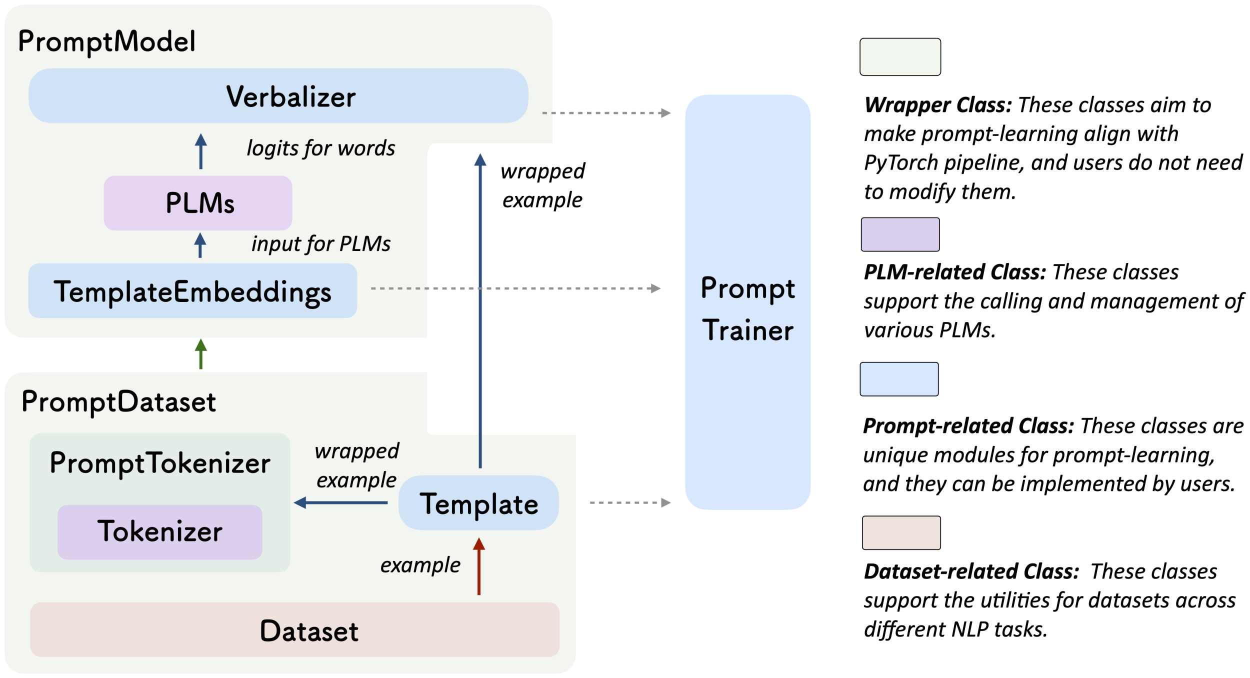
PromptModel
Verbalizer
logits for words
PLMs
input for PLMs
TemplateEmbeddings
wrapped example
PromptDataset
PromptTokenizer
Tokenizer
wrapped example
Template
example
Dataset
Prompt Trainer
Wrapper Class: These classes aim to make prompt-learning align with PyTorch pipeline, and users do not need to modify them.
PLM-related Class: These classes support the calling and management of various PLMs.
Prompt-related Class: These classes are unique modules for prompt-learning, and they can be implemented by users.
Dataset-related Class: These classes support the utilities for datasets across different NLP tasks.

Figure 1: The overall workflow of prompt learning with OpenPrompt