Zero-Shot Recommendation as Language Modeling

Damien Sileo[0000−0002−3274−291X], Wout Vossen, and Robbe Raymaekers

KU Leuven, Belgium
damien.sileo@kuleuven.be

Abstract. Recommendation is the task of ranking items (e.g. movies or products) according to individual user needs. Current systems rely on collaborative filtering and content-based techniques, which both require structured training data. We propose a framework for recommendation with off-the-shelf pretrained language models (LM) that only used unstructured text corpora as training data. If a user u liked *Matrix* and *Inception*, we construct a textual prompt, e.g. *"Movies like Matrix, Inception, <m>"* to estimate the affinity between u and m with LM likelihood. We motivate our idea with a corpus analysis, evaluate several prompt structures, and we compare LM-based recommendation with standard matrix factorization trained on different data regimes. The code for our experiments is publicly available[1].

1 Introduction

Recommender systems predict an affinity score between users and items. Current recommender systems are based on content-based filtering (CB), collaborative filtering techniques (CF), or a combination of both. CF recommender systems rely on (USER, ITEM, INTERACTION) triplets. CB relies on (ITEM, FEATURES) pairs. Both system types require a costly structured data collection step. Meanwhile, web users express themselves about various items in an unstructured way. They share lists of their favorite items and ask for recommendations on web forums, as in (1)[2] which hints at a similarity between the enumerated movies.

(1) *Films like Beyond the Black Rainbow, Lost River, Suspiria, and The Neon Demon.*

The web also contains a lot of information about the items themselves, like synopsis or reviews for movies. Language models such as GPT-2 [14] are trained on large web corpora to generate plausible text. We hypothesize that they can make use of this unstructured knowledge to make recommendations by estimating the plausibility of items being grouped together in a prompt. LM can estimate the probability of a word sequence, $P(w_1, ...w_n)$. Neural language models are trained over a large corpus of documents: to train a neural network, its parameters Θ are optimized for next word prediction likelihood maximization over k-length sequences sampled from a corpus. The loss writes as follows:

$$\mathcal{L}_{\text{LM}} = -\log \sum_i P(w_i | w_{i-k} ....w_{i-1}; \Theta) \quad (1)$$

[1] https://colab.research.google.com/drive/...?usp=sharing
[2] https://www.reddit.com/r/MovieSuggestions/...lost_river/

We rely on existing pretrained language models. To make a relevance prediction , we build a prompt for each user:

$$p_{u,i} = \textit{Movies like} <m_1>, ...<m_n>, <m_i> \quad (2)$$

where $<m_i>$ is the name of the movie m_i and $<m_1...m_n>$ are those of randomly ordered movies liked by u. We then directly use $\widehat{R}_{u,i} = P_\Theta(p_{u,i})$ as a relevance score to sort items for user u.

Our contributions are as follow (i) we propose a model for recommendation with standard LM; (ii) we derive prompt structures from a corpus analysis and compare their impact on recommendation accuracy; (iii) we compare LM-based recommendation with next sentence prediction (NSP) [12] and a standard supervised matrix factorization method [9,15].

2 Related work

Language models and recommendation Previous work leveraged language modeling techniques to perform recommendations. However, they do not rely on natural language: they use sequences of user/item interactions, and treat these sequences as sentences to leverage the architectures inspired by NLP, such as Word2Vec [7,1,4,11] or BERT [19].

Zero-shot prediction with language models Neural language models have been used for zero-shot inference on many NLP tasks [14,2]. For example, they manually construct a prompt structure to translate text, e.g. *Translate english to french : "cheese"* =>, and use the language model completions to find the best translations. Petroni et al. [13] show that masked language models can act as a knowledge base when we use part of a triplet as input, e.g. *Paris in* $<mask>$. Here, we apply LM-based prompts to recommendation.

Hybrid and zero-shot recommendation The cold start problem [17], i.e. dealing with new users or items is a long-standing problem in recommender systems, usually addressed with hybridization of CF-based and CB-based systems. Previous work [20,10,5,6] introduced models for zero-shot recommendation, but they use zero-shot prediction with a different sense than ours. They train on a set of (USER, ITEM, INTERACTION) triplets, and perform zero-shot predictions on new users or items with known attributes. These methods still require (USER, ITEM, INTERACTION) or (ITEM, FEATURES) tuples for training. To our knowledge, the only attempt to perform recommendations without such data at all is from Penha et al. [12] who showed that BERT [3] next sentence prediction (NSP) can be used to predict the most plausible movie after a prompt. NSP is not available in all language models and requires a specific pretraining. Their work is designed as a probing of BERT knowledge about common items, and lacks comparison with a standard recommendation model, which we here address.

3 Experiments

3.1 Setup

Dataset We use the standard the MovieLens 1M dataset [8] with 1M ratings from 0.5 to 5, 6040 users, and 3090 movies in our experiments. We address the relevance prediction task[3], so we consider a rating r as positive if $r \geq 4.0$, as negative if ≤ 2.5 and we discard the other ratings. We select users with at least 21 positive ratings and 4 negative ratings and thus obtain 2716 users. We randomly select 20% of them as test users[4]. 1 positive and 4 negative ratings are reserved for evaluation for each user, and the goal is to give the highest relevance score to the positively rated item. We use 5 positive ratings per user unless mentioned otherwise. We remove the years from the movie titles and reorder the articles (a, *the*) in the movie titles provided in the dataset (e.g. *Matrix, The (1999)* $\rightarrow$ *The Matrix*).

Evaluation metric We use the mean average precision at rank 1 (MAP@1) [18] which is the rate of correct first ranked prediction averaged over test users, because of its interpretability.

Pretrained language models In our experiments we use the GPT-2 [14] language models, which are publicly available in several sizes. GPT-2 is trained with LM pretraining (equation 1) on the WebText corpus [14], which contains 8 million pages covering various domains. Unless mentioned otherwise, we use the GPT-base model, with 117M parameters.

3.2 Mining prompts for recommendation

3-6 gram	**#Count**
$<m>$ *and* $<m>$	387
$<m>$, $<m>$, $<m>$	232
Movies like $<m>$	196
$<m>$, $<m>$, $<m>$, $<m>$	85
Movies similar to $<m>$	25

Table 1: Occurrence counts of 3-6 grams that contain movie names in the Reddit corpus. $<m>$ denotes a movie name.

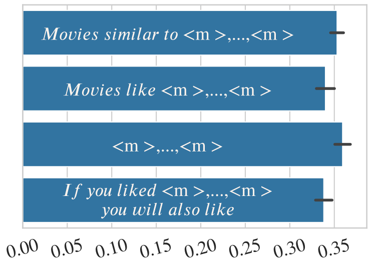

Fig. 1: Comparison of LM recommendations MAP@1 with different prompt structures.

We analyze the Reddit comments from May 2015[5] to find out how web users mention lists of movies in web text. This analysis will provide prompt candidates for LM-based

[3] Item relevance could be mapped to ratings but we do not address rating prediction here.
[4] Training users are only used for the matrix factorization baseline.
[5] `https://www.kaggle.com/reddit/reddit-comments-may-2015`

recommendations. We select comments where a movie name of the MovieLens dataset is present and replace movies with a $<m>$ tag. This filtered dataset of comments has a size of $> 900k$ words. We then select the most frequent pattern with at least three words, as shown in table 1. Movie names are frequently used in enumerations. The patterns *Movies like* $<m>$ and *Movies similar to* confirm that users focus on the similarity of movies.

Figure 1 shows that prompt design is important but not critical for high accuracy. Our corpus-derived prompts significantly outperform *if you like* $<m_1...m_n>$*, you will like* $<m_i>$ used in [12]. We will use $<m_1...m_n>$, $<m_i>$ in the remaining of the paper due to its superior results and its simplicity.

3.3 Effect of the number of ratings per test user

We investigate the effect of the number of mentioned movies in prompts. We expect the accuracy of the models in making recommendations to increase when they get more info about movies a user likes. We compare the recommendation accuracy on the same users 0,1,2,3,5,10,15 or 20 movies per prompt.

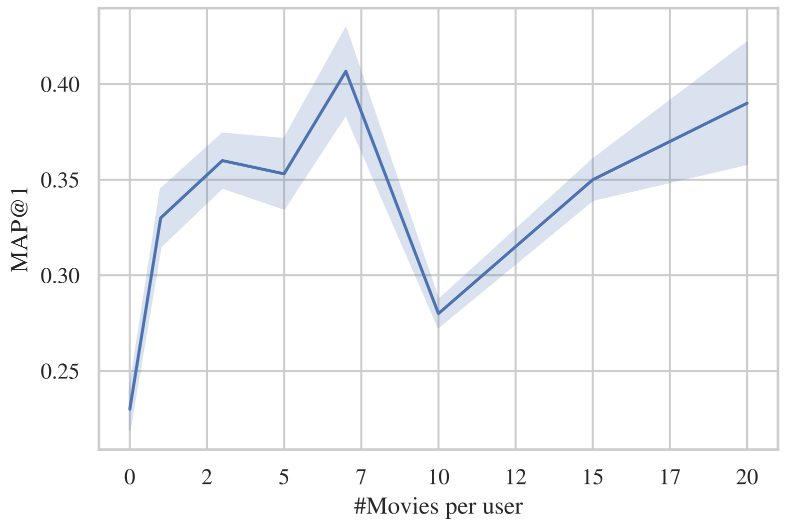

Fig. 2: MAP@1 of LM models with a varying number of movies per user sampled in the input prompt.

Figure 2 shows that increasing the number of ratings per user has diminishing returns and lead to increasing instability, so specifying $n \approx 5$ seems to lead to the best results with the least user input. After 5 items, adding more items might make the prompt less natural, even though the LM seems to adapt when the number of items keeps increasing. It is also interesting to note that when we use an empty prompt, accuracy is above chance level because the LM captures some information about movie popularity.

3.4 Comparison with matrix factorization and NSP

We now use a matrix factorization as a baseline, with the Bayesian Personalised Ranking algorithm (BPR) [15]. Users and items are mapped to d randomly initialized latent factors, and their dot product is used as a relevance score trained with ranking loss. We use [16] implementation with default hyperparameters[6] $d = 10$ and a learning rate of 0.001.

We also compare GPT-2 LM to BERT next sentence prediction [12] which models affinity scores with $\widehat{R}_{u,i} = \text{BERT}_{\text{NSP}}(p_u, <m_i>)$, where p_u is a prompt containing movies liked by u. BERT was pretrained with contiguous sentence prediction task [3] and Penha et al. [12] proposed to use it as a way to probe BERT for recommendation capabilities.

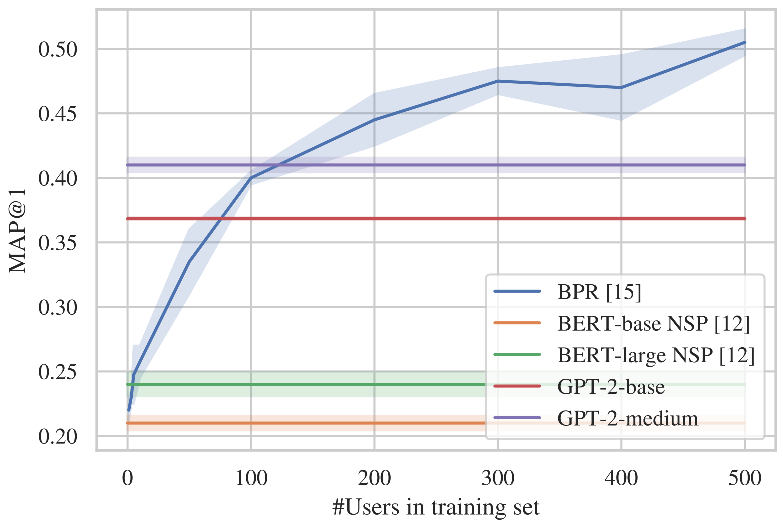

Fig. 3: MAP@1 for BPR models with increasing numbers of users compared the zero-shot language models (with 0 training user). BERT-base and BERT-large respectively have 110M and 340M parameters. GPT-2-base and GPT-2-medium have 117M and 345M parameters.

[6] `https://cornac.readthedocs.io/en/latest/models.html#bayesian-personalized-ranking-bpr`, we experimented with other hyperparameter configurations but did not observe significant changes.

Figure 3 shows that the proposed LM-based recommendation significantly outperforms BERT$_{\text{NSP}}$. We explain the difference by the fact that LM are generative and actually model text likelihood, while next sentence prediction is discriminative and can be based on simple discursive coherence features. It is also interesting to note that LM-based models outperform matrix factorization when there are few users, i.e < 50 and < 100 for BASE and MEDIUM GPT-2, which demonstrates that LM-based recommendation is viable for cold start regimes. Using models larger than the BASE versions lead to better results, however when we evaluated with larger versions (we did not perform the full experiments due to memory limitations), we did not see additional improvement, which could be explained by overfitting.

3.5 Qualitative analysis of generations

Up until there, we have used LM to score the likelihood of sequences. LM can also be used directly for text generation, unlike BERT. We here show LM-generated prompt completions randomly sampled in our dataset, using greedy decoding.

Prompt (P1): *Forrest Gump, Blade Runner, Modern Times, Amelie, Lord of the Rings The Return of the King, Shaun of the Dead, Alexander, Pan's Labyrinth, Cashback, Avatar:*

Completion (C1): *3, The Hunger Games: Mockingjay Part 2, King Arthur, A Feast for Crows, The Hunger Games: Catching Fire, Jackass, Jackass 2, King Arthur*

Prompt (P2): *Independence Day, Winnie the Pooh and the Blustery Day, Raiders of the Lost Ark, Star Wars Episode VI - Return of the Jedi, Quiet Man, Game, Labyrinth, Return to Oz, Song of the South, Matrix:*

Completion (C2): *and many more. The list can be read by clicking on the relevant section at the left of the image. To access the list of releases*

Some prompts, i.e. (P1) generate valid movie names, but others, like (P2), do not. LM-based recommender do need a post-processing to match movie names in the possible sampled generations.

4 Conclusion

We showed that standard language models can be used to perform item recommendations without any adaptation and that they are competitive with supervised matrix factorization when the number of users is very low (less than 100 users). LM can therefore be used to kickstart recommender systems if items are frequently discussed in the training corpora. Further research could explore ways to adjust LM for recommendation purposes or to combine LM with matrix factorization into hybrid systems. Another way to use of our findings would be to generate movie recommendation datasets by mining web data which could feed standard supervised recommendation techniques.

5 Acknowledgements

This work is part of the CALCULUS project, which is funded by the ERC Advanced Grant H2020-ERC-2017 ADG 788506[7].

References

1. Barkan, O., Koenigstein, N.: Item2vec: Neural item embedding for collaborative filtering. In: 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (2016). https://doi.org/10.1109/MLSP.2016.7738886
2. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, https://aclanthology.org/N19-1423
4. Devooght, R., Bersini, H.: Long and short-term recommendations with recurrent neural networks. p. 13–21. UMAP '17, Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3079628.3079670, https://doi.org/10.1145/3079628.3079670
5. Ding, H., Ma, Y., Deoras, A., Wang, Y., Wang, H.: Zero-shot recommender systems (2021)
6. Feng, P.J., Pan, P., Zhou, T., Chen, H., Luo, C.: Zero shot on the cold-start problem: Model-agnostic interest learning for recommender systems. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. p. 474–483. CIKM '21, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3459637.3482312, https://doi.org/10.1145/3459637.3482312
7. Guàrdia-Sebaoun, E., Guigue, V., Gallinari, P.: Latent trajectory modeling: A light and efficient way to introduce time in recommender systems. In: Proceedings of the 9th ACM Conference on Recommender Systems. pp. 281–284 (2015)
8. Harper, F.M., Konstan, J.A.: The movielens datasets: History and context. ACM Trans. Interact. Intell. Syst. **5**(4) (Dec 2015). https://doi.org/10.1145/2827872, https://doi.org/10.1145/2827872
9. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
10. Li, J., Jing, M., Lu, K., Zhu, L., Yang, Y., Huang, Z.: From zero-shot learning to cold-start recommendation. Proceedings of the AAAI Conference on Artificial Intelligence **33**(01), 4189–4196 (Jul 2019). https://doi.org/10.1609/aaai.v33i01.33014189, https://ojs.aaai.org/index.php/AAAI/article/view/4324
11. Li, Z., Zhao, H., Liu, Q., Huang, Z., Mei, T., Chen, E.: Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1734–1743 (2018)

[7] https://calculus-project.eu/

12. Penha, G., Hauff, C.: What does bert know about books, movies and music? probing bert for conversational recommendation. In: Fourteenth ACM Conference on Recommender Systems. p. 388–397. RecSys '20, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3383313.3412249, https://doi.org/10.1145/3383313.3412249
13. Petroni, F., Rockaschel, T., Riedel, S., Lewis, P., Bakhtin, A., Wu, Y., Miller, A.: Language models as knowledge bases? In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2463–2473. Association for Computational Linguistics, Hong Kong, China (Nov 2019). https://doi.org/10.18653/v1/D19-1250, https://aclanthology.org/D19-1250
14. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I.: Language models are unsupervised multitask learners (2019), https://openai.com/blog/better-language-models/
15. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. p. 452–461. UAI '09, AUAI Press, Arlington, Virginia, USA (2009)
16. Salah, A., Truong, Q.T., Lauw, H.W.: Cornac: A comparative framework for multimodal recommender systems. Journal of Machine Learning Research **21**(95), 1–5 (2020)
17. Schein, A.I., Popescul, A., Ungar, L.H., Pennock, D.M.: Methods and metrics for cold-start recommendations. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 253–260. SIGIR '02, Association for Computing Machinery, New York, NY, USA (2002). https://doi.org/10.1145/564376.564421, https://doi.org/10.1145/564376.564421
18. Schröder, G., Thiele, M., Lehner, W.: Setting goals and choosing metrics for recommender system evaluations. In: UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA. vol. 23, p. 53 (2011)
19. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. p. 1441–1450. CIKM '19, Association for Computing Machinery, New York, NY, USA (2019). https://doi.org/10.1145/3357384.3357895, https://doi.org/10.1145/3357384.3357895
20. Volkovs, M., Yu, G., Poutanen, T.: Dropoutnet: Addressing cold start in recommender systems. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/dbd22ba3bd0df8f385bdac3e9f8be207-Paper.pdf