MULTILINGUAL NEXT PRODUCT RECOMMENDATION SYSTEM

Oben Özgür, Faruk Orak, Doğa Türkseven

Introduction

Today, recommender systems are getting a lot of interest in both academia and business. While constantly evolving, recommender systems allow e-commerce platforms to provide a high level of customization for users and buyers, allowing businesses to gain a deeper understanding of their customers. However, there is a scarcity of research examining session-based recommendations in real-world multilingual and imbalanced scenarios. In our project, we build deep learning-based methods with several hyperparameters to tackle this problem.

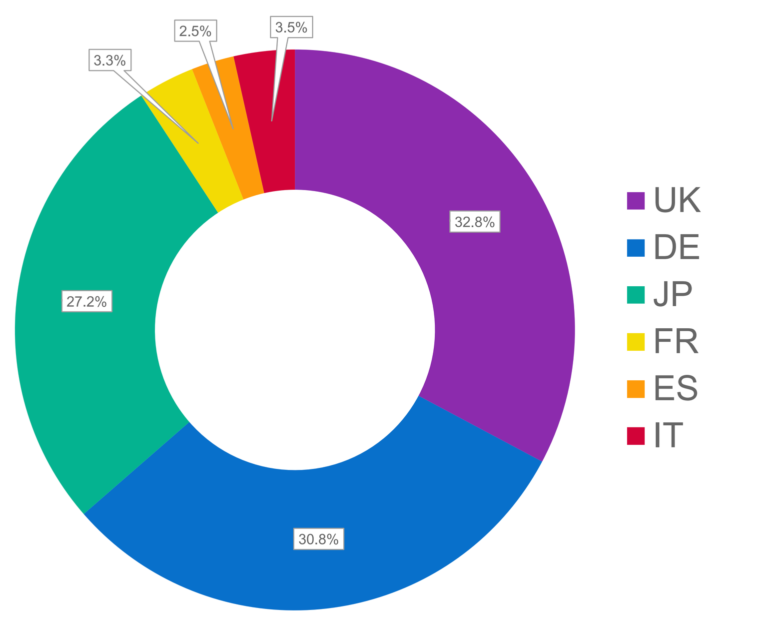
Dataset Analysis

The data we worked on during the project consisted of product information and session data of anonymous users. This data is provided by AICrowd Amazon KDD Cup'23.

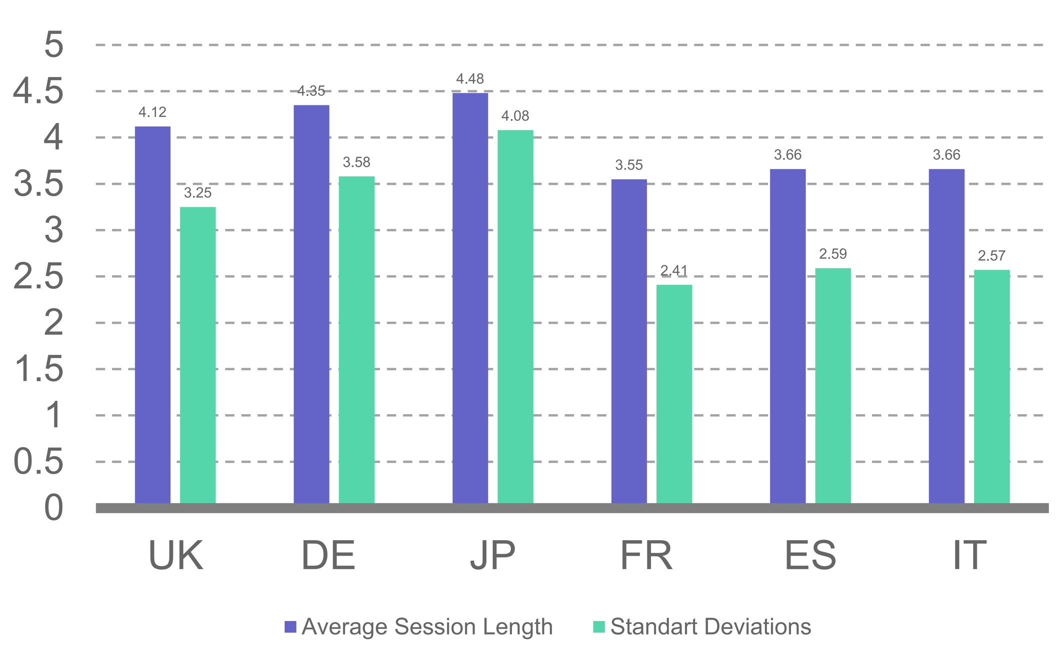
Session Data

- Session data shows which products a user viewed in sequence during that session.
- Session data was created by collecting session data of users from 6 different locales, namely UK, DE, FR, IT, ES, and JP.
- The session data file consists of 3 columns, 3606249 rows. Each row is session data. The first column, prev_items, contains the ids of the products viewed in that session. The second column, next_item, is the first item that should be offered to the user at the end of this session. The 3rd column, locale, indicates which locale the session belongs to.

Session Distribution



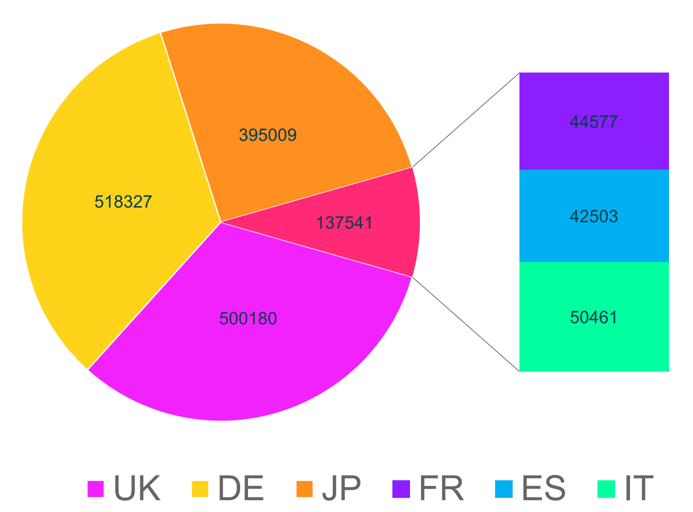
Session Statistics by Locale



Products Data

- In the products data, there is information about all the products and these products in the sessions and test data. There are 1551057 rows and 11 columns in this file. Each row represents a product. A product is represented in more than one locale. Therefore, there are no unique products equal to the number of rows.
- There are several features of the products that detailed in the products data such as id, title, locale, price, brand, color, size, model, material, author, description

Product Distribution

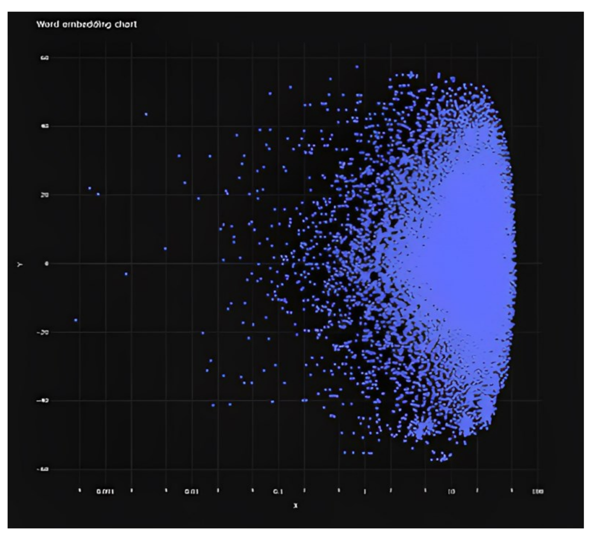


Methods

In our model, we have used Word2Vec and Wikipedia2Vec, for embeddings. Finally, we used SR-GNN to train our model and predict 100 next best product.

Word2Vec

- We used the Word2Vec [1] algorithm to generate word embedding for products.
- We thought of each session as a sentence and each product as a word of that sentence and applied the Word2Vec algorithm to the structure. The higher we keep the window size of Word2Vec algorithm, the relationship between the products in the session will be more.



Session embeddings of ES locale

Wikipedia2Vec

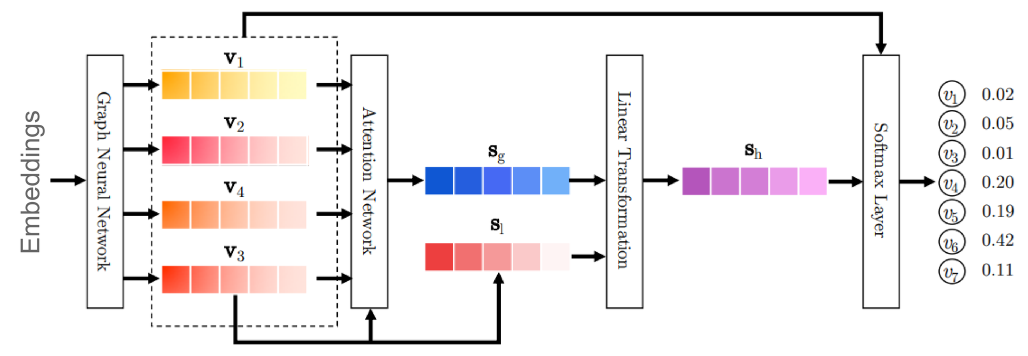
- When it comes to generating product description embeddings, Wikipedia2Vec [2] met our needs.
- Wikipedia2Vec is a sophisticated tool that generates vector representations of words and entities using information from Wikipedia. We used titles and descriptions from product data to generate embeddings.

Embedding Composition

- Two embeddings consisting of title and description data were combined with weighted average and new embedding is obtained. Then, again using weighted averages, new embedding and session embedding were used to generate the final embedding.

SR - GNN

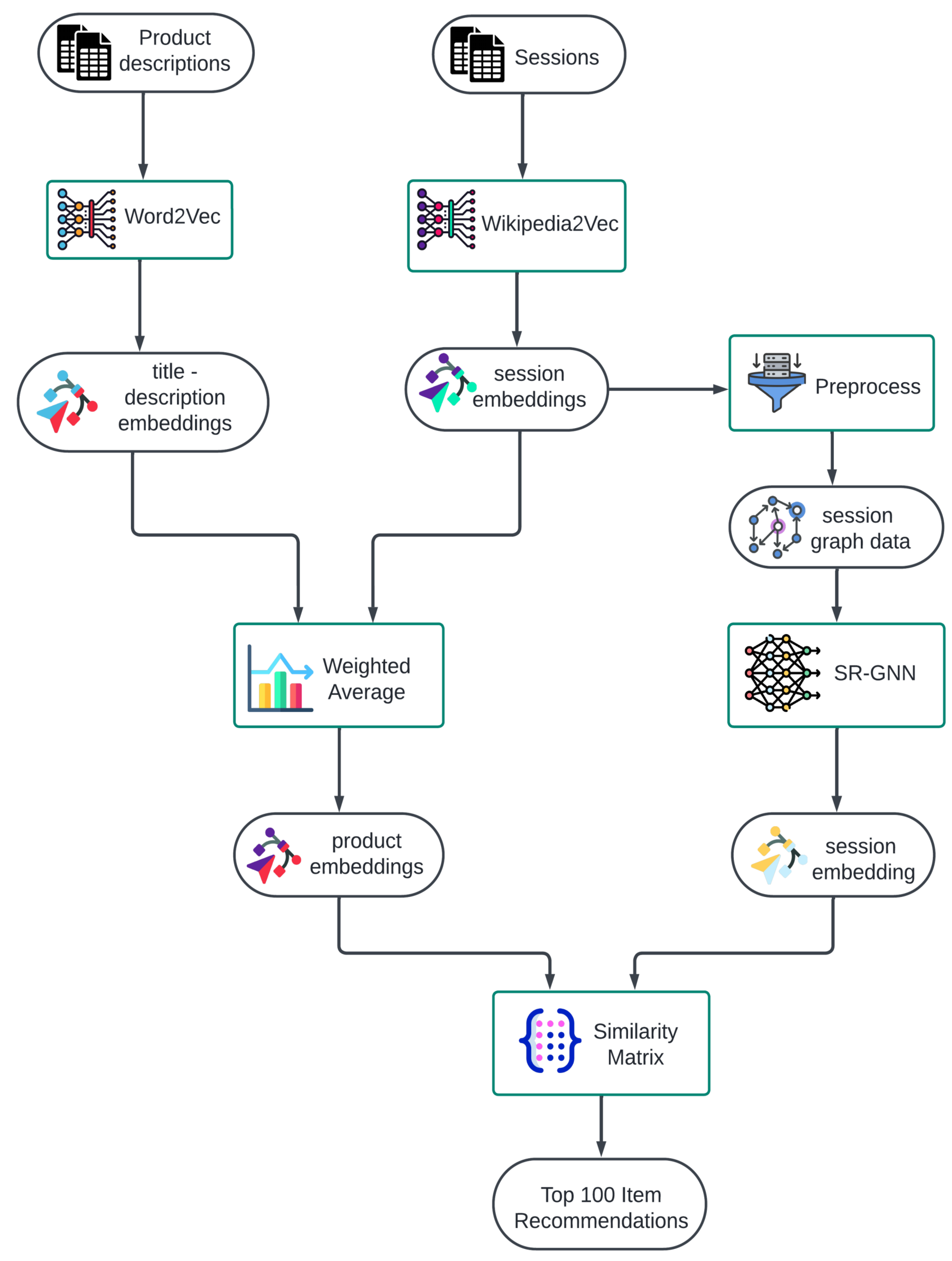
Upon generating the embeddings for each item, we used the SR-GNN (Session-based Recommendation with Graph Neural Networks) [3] architecture. Since the task requires an item prediction task that is solely based on current item interactions defined in a session, SR-GNN became prominent for us due to its capability to capture repeated and complex patterns among items by utilizing Graph Neural Networks. We have directly optimized our model against the cosine similarity between the next item's embedding and the session's embedding the determined the next 100 items in another step. Instead of learning the probabilities, our model directly learns the session's embedding dependent on the cosine similarity.



SR – GNN architecture[1]

Model

- Preprocess: To create embeddings, we remove excessive punctuation from session data. For the product data, we filter out unnecessary characters using language-specific regex filters.
- Session embedding generation: We used built on Word2Vec.
- Title-description embedding generation: We utilized Wikipedia2Vec which uses data from Wikipedia.
- Model training: We used SR-GNN architecture for our model. Each session is turned into graph data, then passed through a GRU. Lastly, session embeddings are generated.
- Prediction: Similarity between each item embedding and session embedding is calculated and the top 100 similar item is recommended for each given session.



Experiment and Results

Mean Reciprocal Rank, often known as MRR, is a performance metric used in information retrieval and recommendation systems to evaluate how effectively a model produces accurate outcomes.

$$MRR = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i}$$

Used Embedding	Window Size	α_1	α_2	Learning Rate	MRR
e_f	average	0.65	0.4	0.001	0.06671
e_f	average	0.65	0.4	0.01	0.08989
e_f	average	0.65	0.4	0.1	0.08615

Used Embedding	Learning Rate	α_1	MRR
e_{td}	0.01	0	0.03462
e_{td}	0.01	0.65	0.10099
e_{td}	0.01	0.8	0.10138

Used Embedding	Window Size	α_1	α_2	Learning Rate	MRR
e_f	average	0.65	0.4	0.01	0.08989
e_f	max	0.65	0.4	0.01	0.10198
e_f	max	0.8	0.4	0.01	0.10831

Used Embedding	Window Size	α_2	Learning Rate	MRR
e_s	max	1	0.01	0.02525

e_{td} : title – description embedding
e_s : session embedding
e_f : final embedding
α_1 : title – description ratio
α_2 : e_{td} - e_s ratio

Conclusion

We proposed a model for session-based recommender systems in this research. We conducted experiments by changing different hyperparameters, including window size, learning rate, title-description embedding ratio, and session-product embedding ratio. In the prediction step, we calculate the similarity between each embedding and recommend the top 100 for each session accordingly. According to the results of our experiments, we got the best results by giving the title more weight than the description. Additionally, we discovered that while simultaneously maximizing the window size, it was advantageous to give the title-description embedding ratio larger than the session embedding ratio.

References

1. T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," arXiv.org, Sep. 07, 2013. https://arxiv.org/abs/1301.3781

2. I. Yamada et al., "Wikipedia2Vec: An Efficient Toolkit for Learning and Visualizing the Embeddings of Words and Entities from Wikipedia," EMNLP 2020, Sep. 2020, doi: https://doi.org/10.18653/v1/2020.emnlp-demos.4.

3. S. Wu, Y. Tang, Y. Zhu, L. Wang, X. Xie, and T. Tan, "Session-Based Recommendation with Graph Neural Networks," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 346–353, Jul. 2019, doi: https://doi.org/10.1609/aaai.v33i01.3301346.

İTÜ

ISTANBUL TECHNICAL UNIVERSITY

Department of Computer Engineering

Faculty of Computer and Informatics Engineering
İTÜ Ayazaga Campus, 34469, Maslak/Istanbul
bbf@itu.edu.tr
http://bm.itu.edu.tr