Evaluation of Session Segmentation Methods Using Behavior and Text Embeddings

Yongzhi Jin[1] and Kazushi Okamoto[1][0000−0002−9571−8909]

The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu, Tokyo 182-8585, Japan
{jin-yongzhi,kazushi}@uec.ac.jp

Abstract. In the context of a recommendation system, a user session is a list of user actions or operations that occur during a specific period while using a website, application, or online service. These sessions are used in session-based recommendation systems to learn user preferences, and are applied to item recommendations. However, because user preferences change dynamically, different topic elements are combined in a single session. To accurately capture changes in user preferences, it is important to divide sessions by topic and analyze the separate subsessions. Therefore, the selection of the best session-segmentation method is an important issue. In this study, we investigate cosine similarity, k-means clustering, and classifier including Light Gradient Boosting Machine (LightGBM), Support Vector Machine (SVM), and Logistic Regression to segmentation methods. In addition, Item2Vec, Word2Vec, and OpenAI were used for item embedding. We conducted a comparative evaluation using annotation data to assess the combination of the item embedding and segmentation methods. The evaluation metrics used were the F1-score, PR-AUC, and ROC-AUC. The combination of cosine similarity and Item2Vec exhibited the best performance. The values achieved were an F1-score of 0.714 and PR-AUC of 0.794.

Keywords: Session Segmentation · Embedding · Classifier · k-means

1 Introduction

In recent years, many online platforms, such as e-commerce sites, music streaming services, and video sharing services, have increasingly integrated recommender systems (RSs) to improve user experience [9, 10]. Session-based Recommender Systems (SBRSs) have gained particular attention because of their ability to respond to users' recent preferences in real time. An information recommendation session refers to a list of actions taken by a user while interacting with an online service [11]. The browsing history during a visit to an e-commerce site is an example of such session. An SBRS learns user preferences from input session data and recommends items. User preferences change dynamically, and a session may contain a mix of content on various topics. Training a model directly using full sessions can degrade the prediction accuracy. Although session-segmentation plays an important role in handling session data, few studies have

explored session-segmentation methods, and there is no standardized evaluation metric for segmentation accuracy.

The number of studies on RSs that use embedding has increased in recent years. Embedding techniques map data into a vector space, representing words or items as vectors. For example, in the context of natural language processing (NLP), Word2Vec [8] is a prominent method for word embedding. In addition, as an example of the use of Word2Vec in a RS, Item2Vec [2], which is a method for obtaining a distributed representation of product items based on user behavior, was proposed. Embeddings are frequently employed in session-segmentation tasks. The most common approach involves calculating the cosine similarity between embeddings before and after a target item within a session and comparing this value to a predefined threshold. Zhang [13] et al. proposed a session segmentation method based on cosine similarity using Item2Vec. However, there has been limited research on session-segmentation methods in the field of item recommendation, and no comprehensive comparative evaluation of these methods has been conducted.

This study aims to identify the optimal embedding-based session segmentation method for item recommendation. Various embedding methods are introduced to determine the most effective segmentation method. Because few existing studies have employed methods other than cosine similarity, we applied several segmentation methods (supervised and unsupervised approaches) and evaluated them using a unified metric, comparing their performance with existing methods. However, as there is no publicly available dataset specifies session segmentation points, we conducted an annotation process to obtain the necessary ground-truth data. All ground-truth data utilized in the evaluation experiments and training of the classifier model were meticulously prepared through this annotation process. Annotators were recruited to label the correct session segment points within a session. Finally, we evaluated the segmentation performance of each segmentation method using three metrics: ROC-AUC, PR-AUC, and F1-score, using the ground-truth segmentation point data generated through annotation.

Our contribution is that we clarified an optimal approach for segmenting human action sequences by determining the optimal session-segmentation method utilizing embeddings. This advancement facilitates a more accurate analysis of user behavior, potentially yielding benefits such as the delivery of personalized services and enhanced accuracy in behavioral analysis.

2 Related Work

2.1 Application of Session Data

According to Wang [11] et al., in the context of an SBRS, a session is defined as "a series of interactions between a user and an item within a specific time frame". For example, a session is a list of products viewed or purchased by a user in a single access. The concept of a session is used not only in RSs but also in the Web industry. In the Web industry, a session is a series of actions performed by

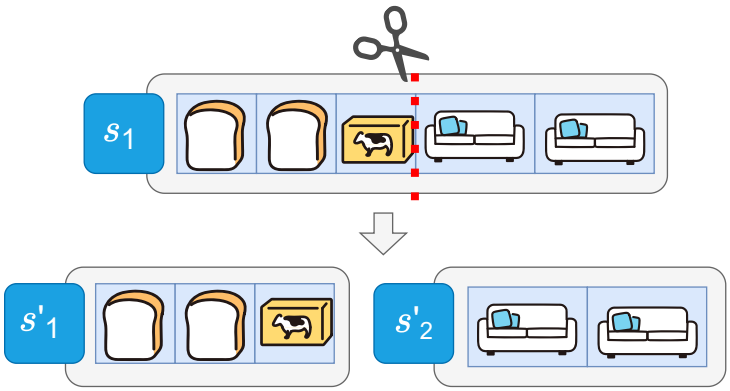


Fig. 1: Description of Session-Segmentation Task

a user after accessing a website. In this study, we aim to segment user sessions on e-commerce sites by topic. Fig. 1 illustrates the session-segmentation task, which involves partitioning an original session into distinct subsessions, such as "looking for breakfast" and "looking for a sofa," as depicted in Fig. 1

2.2 Text-based and Behaviour-based Embedding Methods

Text-based Embedding Word2Vec effectively captures word meanings in NLP tasks by converting word representations into dense vectors. Word2Vec is a neural network model based on the distributional hypothesis, which states that a word's meaning is derived from the context provided by its surrounding words in context. Typically, skip-gram and continuous bag-of-words (CBOW) are employed as architectures for training Word2Vec. In skip-gram, a word vector (one-hot) is given in the input layer, and a vector of surrounding words is output in the output layer. Conversely, CBOW is the reverse of skip-gram, where the input layer receives the vectors of the surrounding words and the output layer outputs the vectors of the target words. In both architectures, the training adjusts the weights from the input layer to the hidden layer, and these weights become a matrix of embeddings for each word. A notable strength of Word2Vec is its additive compositionality. Additive compositionality refers to the ability of Word2Vec-generated embeddings to reflect semantic relationships through vector arithmetic. For example, subtracting the "male" embedding and adding the "female" embedding to the "king" embedding results in a word that is similar to the "queen" embedding.

Besides Word2Vec, other embedding methods are also available, such as those developed by OpenAI. Specifically, OpenAI's text-embedding-ada-002 model demonstrates superior performance compared to the Davinci model for sentence embedding tasks.

Behavior-based Embedding An example of behavior-based embedding is Item2Vec. Item2Vec is a method derived from Word2Vec. While Word2Vec focus on learning the relationships between target words and their surrounding words, Item2Vec focuses on learning the relationship between the target and surrounding items within a session. The embedding of an item (user behavior) can be obtained by inputting the session into the Word2Vec model as training data. In the genre-clustering task, Item2Vec has demonstrated superior performance compared to the SVD model in terms of accuracy [2]. The application of a highly accurate embedding model is expected to enhance the accuracy of session-segmentation tasks. In addition, behavior-based embedding approaches have been actively employed in RS [1].

2.3 Segmentation Methods

Zhang [13] et al. proposed a method for preventing attribute inference attacks by employing behavior segmentation with cosine similarity. Specifically, to preserve security from external attribute inference attacks, they segmented a user session containing user browsing history or item ratings into several dummy sessions thereby misleading potential attackers.

The cosine similarity-based segmentation method, combined with embeddings, has been used not only in RSs but also in chat-session segmentation tasks. A chat session contains utterances between users. Lee [7] et al. utilized Doc2Vec to embed utterances and performed segmentation based on changes in the meaning of interactions, by evaluating the similarity between utterance embeddings.

Additionally, research has explored session-segmentation methods that do not use embeddings. Hou [5] et al. proposed a session-segmentation method based on the COBWEB method for partitioning web sessions and mining user preferences. Hienert [4] et al. segmented sessions based on session topics labeled with annotations to detect points in time when user behavior changes.

3 Experimental Environment

This study's experimental procedure comprised two distinct phases: the creation of ground-truth data through annotation and session segmentation. In this section, we present an overview of these experimental phases, followed by a detailed description of the dataset employed and the results obtained from the annotation process.

3.1 Experiment Flow

Fig. 2 illustrates the experimental workflow. In this study, we began by obtaining the respective embeddings for all items in the observed sessions. Specifically, we employed three distinct embedding methods: Item2Vec, Word2Vec, and text-embedding-ada-002. Next, we focused on two consecutive items using the obtained embeddings, and determined whether they were segment points in the

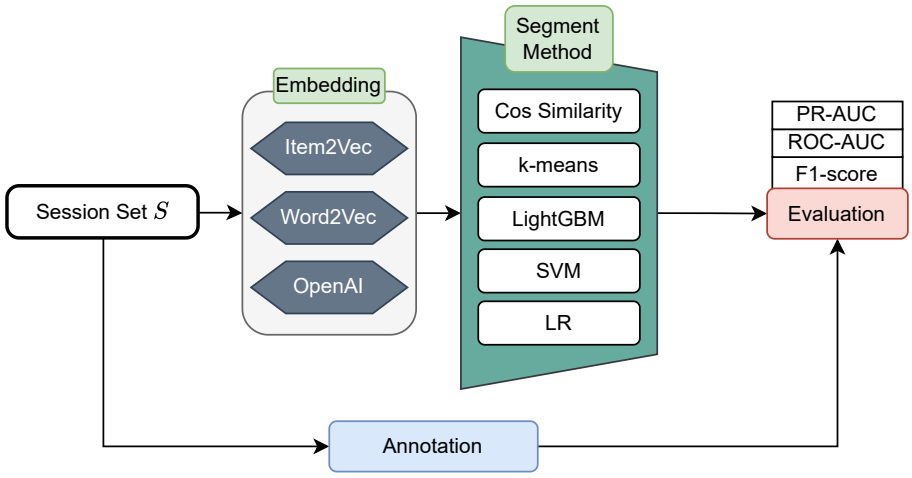


Fig. 2: Experiment Flow

session using different methods. Concurrent to this, because the session data did not contain information on the segment points of user behavior, an annotation process was conducted to generate the ground-truth data for session segmentation. Finally, the annotated ground-truth data were utilized to evaluate the performance in across three metrics: PR-AUC, ROC-AUC, and F1-score.

3.2 Dataset

This study used the Amazon-M2 [6] session dataset from the Amazon KDD Cup 2023, which encompasses millions of user sessions from six distinct locales The dataset includes items in various languages, namely English, German, Japanese, French, Italian, and Spanish. In this study, we focused on sessions and items that fell under the Japanese locale (Locale = JP) of the Amazon-M2 dataset. This data comprises 979,119 sessions and 389,888 items. The Amazon-M2 dataset contains two types of session data: "prev item," which is the user's previous browsing history, and "next item," which is the next item to be browsed. In this study, we combined "prev item" and "next item" to constitute session data. Each item in the dataset is characterized by the following 11 data fields: Locale, ID, Title, Price, Brand, Color, Size, Model, Material, Author, and Description.

3.3 Annotation

In this study, an annotation process was conducted to generate the ground-truth data. A custom annotation form was developed to facilitate annotation tasks. The server for the annotation form was established using Apache. In the annotation form, the code for data transfer to the server was embedded in an HTML file using PHP. The sessions to be annotated were a set of sessions $S_r \subset S$

link	title
リンク	【指定第2類医薬品】メンソレータムメディクイックE 30mL ※セルフメディケーション税制対象商品
	分割点 ☐
リンク	【第2類医薬品】ミーミエイド 5g ※セルフメディケーション税制対象商品
	分割点 ☑
リンク	ソニー ハイブリッドイヤーピース EP-EX11M : Mサイズ 4個入り ブラック EP-EX11M B

Fig. 3: Example of Annotation Form

randomly selected from session set S. Annotators were tasked with labeling the segment points within the subset S_r. The annotators were students enrolled in the author's laboratory. Fig. 3 shows an example of the annotation form.

A total of eight annotators labeled 2,400 sessions. Of the 2,400 sessions, there were 10,904 points between the two items, that is, the amount of data to be labeled. Specifically, 1,230 positive examples (segmented points) and 9,674 negative examples (non-segmented points) were identified. Fig. 4 depicts the distribution of the number of segment points attached to sessions of different lengths, by annotator. As shown in Fig. 4, approximately 40% of the sessions required segmentation, highlighting the necessity for segmentation of the session data.

4 Session-Segmentation Experiment

4.1 Item Embedding

First, we begin by detailing the embedding methodology using Item2Vec, which is based on Word2Vec. Behavior embeddings are generated by learning the associations between target items and surrounding items in a session. In this study, an Item2Vec model was trained using the Word2Vec implementation available in the Gensim library. While Word2Vec was implemented with a set of sentences for text-based embedding task, Item2Vec was applied to a collection of sessions for behavior-based embedding tasks. The ground-truth data obtained through the annotation process were utilized to evaluate the performance of the segmentation method. However, to prevent data leakage, the session set S minus the session set S_r was input for Item2Vec training. The tunable parameters of the Word2Vec model were configured as follows:

- vector size = 200 : number of dimensions of embedding
- sample = 10^{-3}: sampling threshold to ignore high frequency words

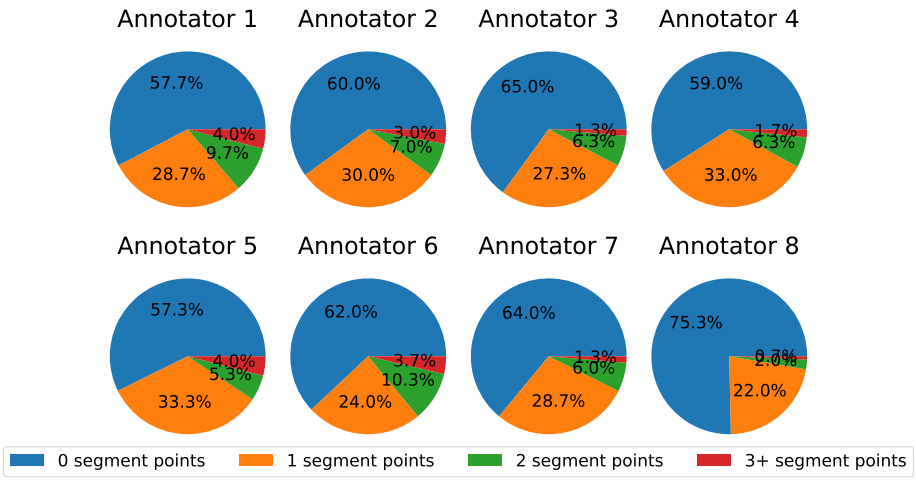


Fig. 4: Segmented and Non-segmented Points Distribution. The legend indicates the number of labeled segment points in each session.

- negative = 1: Number of negative examples to be sampled for each positive example in negative sampling
- window = 6: Size of the context window used for training
- min count = 1: Minimum number of occurrences of a word used for training
- epochs = 100: Number of times the entire training data is repeated
- sg = 1 : Whether skip-grams are used or not

Next, we describe the text-based embedding implementation of Word2Vec. Section 2.2 explains the additive compositionality of Word2Vec. The meanings of the words are added by adding the embedding of multiple words. In this study, we leveraged this property of additive compositionality. Treating the item name as a sentence, the text of the item name was segmented using the Japanese Morphological Analysis Library, which is used for document segmentation. For each segmented word, we utilized the pre-trained model, the WikiEntVec model (jawiki.all_vectors.100d), to obtain the embedding for each word. Finally, we computed the average of the vectors of word embeddings and considered them item embeddings. This approach—averaging word embeddings to obtain sentence embeddings—has been widely accepted as a basic baseline method [12]. Hereafter, this text-based embedding is referred to as Word2Vec.

Finally, we introduce the text-based embedding produced by OpenAI, specifically the text-embedding-ada-002 model, which was released in 2022 [3]. The model not only provides text, but also includes features for text search, text similarity, code search, and text embedding. In addition, it outperforms the old Davinci model for most tasks and cost 99.8 percent less. In this study, we employed text-embedding-ada-002 to embed item names and treat the resulting vectors as item embeddings.

4.2 Session Segment Methods

Unsupervised Methods First, the cosine similarity method is described. A session $s_i = \{v_1, v_2, \cdots, v_{|s_i|}\}$, for two consecutive items v_k and v_{k+1} in the session, the item embeddings $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ are obtained using the embeddings introduced in Section 3.3. Next, the cosine similarity $sim(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})$ of the embeddings of the previous and the following items is computed: The cosine similarity indicates the similarity between items. If the cosine similarity is low, it suggests a break in user behavior. In other words, the lower the cosine similarity, the more likely it is that a session will be segmented. Therefore, we consider $1 - sim(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})$ as the segment probability p of the session segmentation point based on the cosine similarity. Even if $sim(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})$ is negative, p must never exceed 1.

Next, the clustering method is described. The k-means method was used to cluster the embeddings $\boldsymbol{x}$ of all items v in item set V. The k-means method is a nonhierarchical clustering method that divides a dataset into k groups under a specified number of clusters k. Clustering using the k-means method enables the grouping of items that are similar in content. Clustering is performed on the embedding $\boldsymbol{x}$ of the set of all items V, and the clusters of items in session $s_i = \{v_1, v_2, \cdots, v_{|s_i|}\}$ are recorded. Subsequently, the clusters of consecutive items v_k and v_{k+1} are examined; If these items belong to different clusters, the point is marked as a session segment point.

Supervised Methods Next, we explain the application of the classifiers. In this study, three classifiers-LightGBM, SVM (with an RBF kernel), and Logistic Regression (LR)-were applied to the session-segmentation task. A classifier is a model for classifying input data into different classes or categories. It can be trained using a training dataset to predict and classify unknown data. The following steps were performed to apply the classifier

1. Concatenate the embeddings $\boldsymbol{x}_k$ and $\boldsymbol{x}_{k+1}$ of the previous and next items to form a vector $\boldsymbol{x}_{input}$.
2. Let y represent the segment point information obtained from the $\boldsymbol{x}_{input}$ annotation.
3. $\boldsymbol{x}_{input}$ and y are used to train the optimal model by hyperparameter tuning with cross-validation.
4. Finally, the trained model is applied to the test data and the segment probability p is output.

4.3 Hyper Parameter Tuning

In Section 3.4, we discuss the use of different segmentation methods. In this section, we focus on the adjustment of the hyperparameters for the four methods: k-means, LightGBM, SVM, and LR.

First, preprocessing was performed on the annotation data S_a. For the embedding $\boldsymbol{x}_i$ of each item v_i in the session $s \in S_a$, we extracted two consecutive

Table 1: Methods and Hyperparameters

Method	Hyperparameter
k-means	number of clusters $k \in \{10, 20, \cdots, 300\}$
LightGBM	feature_fraction $\in \{0.6, 0.7, \cdots, 1.0\}$, num_leaves $\in \{25, 30, \cdots, 45\}$
SVM	$C \in \{0.1, 1, 10, 100\}$, $\gamma \in \{0.001, 0.01, 0.1, 1, 10\}$
LR	$C \in \{0.001, 0.01, 0.1, 1, 10, 100\}$

item pairs $\boldsymbol{x}' = (\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$ for each item in the session $\{\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{|S|}\}$ where $i = \{1, 2, \cdots, |S| - 1\}$. This process was performed for all sessions $s \in S_a$. We then define the set $\boldsymbol{x}'$ as the dataset $\boldsymbol{X}$, define y as the segment point information corresponding to each item pair $\boldsymbol{x}'$ obtained by annotation, and define Y as the set of all the ground-truth data y. Finally, we divided dataset $\boldsymbol{X}$ into training and validation data $\boldsymbol{X}_{train\_valid}$ and test data $\boldsymbol{X}_{test}$ at a ratio of 8:2.

Next, we explain the hyperparameters of each model, the search range, and the hyperparameter tuning method. The search parameter ranges for each method are listed in Table 1.

Finally, the hyperparameter-tuning methods for each segmentation method are described. In this study, hyperparameter tuning was performed using 5-fold cross-validation. Here, the training data $\boldsymbol{X}_{train}$ and validation data $\boldsymbol{X}_{valid}$ used in the cross-validation are assumed to be segmented from $\boldsymbol{X}_{train\_valid}$ at a ratio of 8:2. The specific tuning method is outlined below.

First, for k, the search parameter of the aforementioned k-means method, clustering is performed on the embeddings of all item sets V using all k values in the range. Unlike the classification models, k-means clustering does not involve model learning but instead requires cross-validation for consistency in evaluation. Specifically, for each 5-fold validation data $\boldsymbol{X}_{valid}$, the session was segmented according to clustering results with k clusters. The optimal hyperparameters were determined by evaluating F1-score across all folds. Finally, the value of k clusters with the highest average F1-score was taken as the optimal parameter. Due to the binary nature of segmentation results, ROC and PR curves typically present only discrete points on the graph, even when adjusting thresholds; thus, ROC-AUC and PR-AUC were recorded as zero.

Using the aforementioned parameter grid, the optimal hyperparameters for the classifier were identified through a grid search aimed at maximizing the PR-AUC. As the dataset in this task was unbalanced, the training data $\boldsymbol{X}_{train}$ were under-sampled. The training data consist of two consecutive items and their segment labels. Because the segment labels are not expected to change, even if the order of the items changed, the data are expanded by switching the order of the two items in the item pair $\boldsymbol{x}'$. For example, $\boldsymbol{x}' = (\boldsymbol{x}_i, \boldsymbol{x}_{i+1})$ is changed to $\boldsymbol{x}'_{new} = (\boldsymbol{x}_{i+1}, \boldsymbol{x}_i)$ and is treated as new data. Each classifier outputs the segmentation result in the form of probability. however, for the purpose of session segmentation and F1-score evaluation, the segmentation decision threshold is set to 0.5.

Table 2: PR-AUC of session-segmentation results for each method. The values in bold indicate the highest scores among the three embedding methods, and the asterisk indicate the highest scores among the five segmentation methods

Segmentation Method	Embedding Method		
	Item2Vec	Word2Vec	text-embedding-ada-002
Cosine Similarity	**0.794***	0.343	0.683*
k-means	–	–	–
LightGBM	**0.668**	0.497	0.590
SVM	0.468	0.504*	**0.594**
LR	**0.174**	0.161	0.158

Table 3: ROC-AUC of session-segmentation results for each method. The bold font and asterisk indicate the same information as that in Table 2

Segmentation Method	Embedding Method		
	Item2Vec	Word2Vec	text-embedding-ada-002
Cosine Similarity	0.935*	0.874	**0.950***
k-means	–	–	–
LightGBM	**0.938**	0.889*	0.909
SVM	0.876	0.883	**0.895**
LR	**0.665**	0.627	0.640

5 Results and Discussion

In this section, we present the results of the performance evaluation of the segmentation methods using test data $\boldsymbol{X}_{test}$. Tables 2 and 3 show the results of the evaluation of session-segmentation performance for the combinations of the segmentation methods and embedding methods. Finally, Table 4 provides the F1-score calculated with a session-segmentation threshold set to 0.5. In these tables, each column corresponds to a specific embedding method, while each row represents a particular session-segmentation method.

Values highlighted in bold in Tables 2, 3, 4 indicate the highest scores achieved among the three embedding methods. Additionally, values marked with an asterisk denote the highest scores among the five segmentat ion methods for a given embedding method.

From Tables 2, 3, and 4, it is evident that the session-segmentation method employing Item2Vec achieves the highest performance in terms of PR-AUC and F1-score. Conversely, the session-segmentation method utilizing text-embedding-ada-002 demonstrates superior performance with respect to ROC-AUC. These findings suggest that the session-segmentation approach based on cosine similarity is most suitable for session-segmentation tasks. Furthermore, these results highlight the advantages of behavior-based embedding in session-segmentation tasks. Item2Vec embeds user behaviors, whereas Word2Vec and text-embedding-ada-002 embed content information, such as item names. Because Item2Vec outperformed both Word2Vec and text-embedding-ada-002 in overall segmentation

Table 4: F1-score of session-segmentation results for each method. The bold font and asterisk indicate the same information as that in Table 2

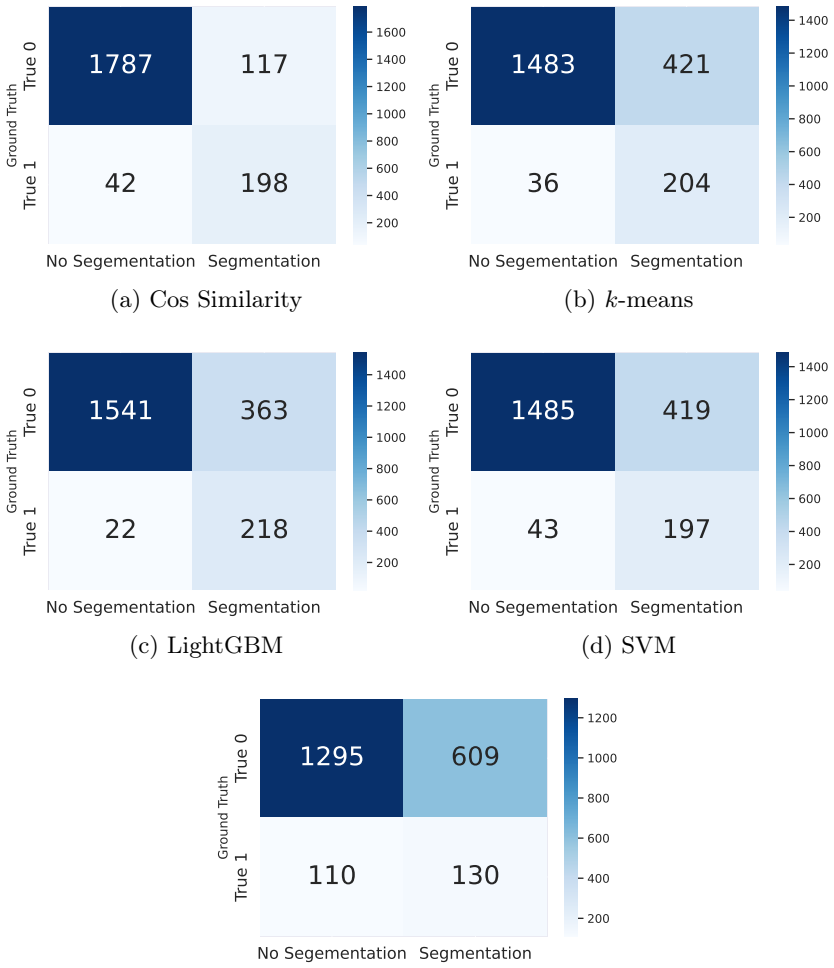
Segmentation Method	Embedding Method		
	Item2Vec	Word2Vec	text-embedding-ada-002
Cosine Similarity	**0.714***	0.093	0.000
k-means	**0.472**	0.352	0.463
LightGBM	**0.531**	0.476*	0.486*
SVM	0.460	0.465	**0.476**
LR	**0.266**	0.249	0.249

Table 5: False-Positive Rate of Item2Vec-based Segmentation Method

Segmentation Method	False-Positive Rate
Cosine Similarity	0.371
k-means	0.674
LightGBM	0.625
SVM	0.680
LR	0.824

performance, we conclude that embedding user behaviors is suitable for the session-segmentation task.

Subsequently, an analysis was performed utilizing a confusion matrix for the Item2Vec-based segmentation method. Fig. 5 presents the confusion matrix for each of the Item2Vec-based segmentation methods and Table 5 provides the false-positive rate of each result. As depicted in Fig .5, the false-positive rates of the machine learning methods (k-means, LightGBM, SVM, LR) exceed 0.5. This indicates that more than half of the segments labeled by the model were incorrect. Segmentation methods employing supervised learning techniques, such as LightGBM, SVM, and LR, performed poorly compared to the cosine similarity method, which is the simplest approach. We attribute these results to the nature of the data input to the classifier model. In this study, the classifier was trained using a vector that combined the embeddings of consecutive items in a session. However, using this method, the classifier can only learn specific action patterns, which may reduce its versatility. In contrast, the Item2Vec-based cosine similarity method demonstrated superior performance. The cosine similarity method had the lowest false positive rate among the five methods and exhibited excellent performance. Thus, the similarity between two consecutive items is effectively utilized to determine whether a segmentation point exists.



(a) Cos Similarity (b) k-means

(c) LightGBM (d) SVM

(e) LR

Fig. 5: Confusion matrix for each Item2Vec-based segmentation method

6 Conclusion

In this study, we utilized three embedding methods – Item2Vec, Word2Vec, and text-embedding-ada-002 – and investigated session-segmentation methods utilizing cosine similarity, k-means, and classifiers (LightGBM, SVM, and LR). We then evaluated each method using the annotated data and compared them based on terms of F1-score, ROC-AUC, and PR-AUC metrics. The evaluation results demonstrated that cosine similarity segmentation using Item2Vec was the best in terms of the F1-score and PR-AUC, with scores of 0.714 and 0.794, respectively. On the other hand, cosine similarity segmentation based on embedding

Table 6: Results of Hyperparameter Tuning

Segmentation Method	Embedding Method		
	Item2Vec	Word2Vec	text-embedding-ada-002
k-means	$k = 270$	$k = 30$	$k = 20$
LightGBM	feature_fraction=1.0, num_leaves=25	feature_fraction=1.0, num_leaves=35	feature_fraction=0.9, num_leaves=35
SVM	C=10, gamma=0.01	C=10, gamma=1	C = 10, gamma=1
LR	C=0.001	C=0.1	C=1

in text-embedding-ada-002 performed the best in terms of ROC-AUC, achieving an ROC-AUC of 0.950. These findings suggest that the cosine-similarity-based segmentation method is suitable for session-segmentation tasks.

However, this study has several limitations. All methods employed in this study considered only the items immediately preceding and following the segmentation point. This approach fails to leverage the full potential of session data as a sequence of behaviors. In addition to analyzing the relationship between the target item and the items immediately before it, it is necessary to analyze the relationship between the target item and more context items. Incorporating a wider context of items is anticipated to capture a more comprehensive range of behavioral characteristics, thereby enhancing the segmentation performance of the model.

A Hyperparameter Tuning Result

This section presents the results of the search for the optimal hyperparameters based on the hyperparameter tuning method introduced in Section 4. Table 6 lists the determined hyperparameters.

References

1. Barkan, O., Caciularu, A., Rejwan, I., Katz, O., Weill, J., Malkiel, I., Koenigstein, N.: Cold item recommendations via hierarchical item2vec. In: Proceedings of 2020 IEEE International Conference on Data Mining (ICDM). pp. 912–917 (2020)
2. Barkan, O., Koenigstein, N.: Item2vec: Neural item embedding for collaborative filtering. In: Proceeding of 2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (2016)
3. Greene, R., Sanders, T., Weng, L., Neelakantan, A.: Openai:new and improved embedding model. https://openai.com/blog/new-and-improved-embedding-model (2022), accessed:14-Jan-2024
4. Hienert, D., Kern, D.: Recognizing topic change in search sessions of digital libraries based on thesaurus and classification system. In: Proceeding of the 18th Joint Conference on Digital Libraries. pp. 297–300 (2020)
5. Hou, Z., Cui, M., Li, P., Wei, L., Ying, W., Zuo, W.: Session segmentation method based on cobweb. In: Proceeding of 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems. vol. 01, pp. 148–153 (2012)

6. Jin, W., Mao, H., Li, Z., Jiang, H., Luo, C., Wen, H., Han, H., Lu, H., Wang, Z., Li, R., Li, Z., Cheng, M.X., Goutam, R., Zhang, H., Subbian, K., Wang, S., Sun, Y., Tang, J., Yin, B., Tang, X.: Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation. arXiv preprint arXiv:2307.09688 (2023)
7. Lee, H., Yoon, Y.: Interest recognition from online instant messaging sessions using text segmentation and document embedding techniques. In: Proceeding of 2018 IEEE International Conference on Cognitive Computing (ICCC). pp. 126–129 (2018)
8. Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Proceeding of the 26th International Conference on Neural Information Processing Systems. vol. 02, pp. 3111–3119 (2013)
9. Palumbo, E., Damianou, A., Wang, A., Liu, A., Fazelnia, G., Fabbri, F., Ferreira, R., Silvestri, F., Bouchard, H., Hauff, C., Lalmas, M., Carterette, B., Chandar, P., Nyhan, D.: Graph learning for exploratory query suggestions in an instant search system. In: Proceeding of the 32nd ACM International Conference on Information and Knowledge Management (CIKM). pp. 4780–4786 (2023)
10. Pandey, D., Sarkar, A., Comar, P.M.: Glad: Graph-based long-term attentive dynamic memory for sequential recommendation. In: Proceeding of 46th European Conference on Information Retrieval (ECIR) (2024)
11. Wang, S., Cao, L., Wang, Y., Sheng, Q.Z., Orgun, M.A., Lian, D.: A survey on session-based recommender systems. ACM Computing Surveys **54**(154), 1–38 (2021)
12. Yamada, I., Suzuki, M., Yamada, K., Li, L.: Introduction to Large language Models. Gijutsu-Hyohron Co., Ltd (2023)
13. Zhang, Y., Gao, N., Chen, J.: A practical defense against attribute inference attacks in session-based recommendations. In: Proceeding of 2020 IEEE International Conference on Web Services (ICWS). pp. 355–363 (2020)