Enhancing ID and Text Fusion via Alternative Training in Session-based Recommendation

Juanhui Li[1], **Haoyu Han**[1], **Zhikai Chen**[1], **Harry Shomer**[2],
Wei Jin[3], **Amin Javari**[4*], **Hui Liu**[1]
[1]Michigan State University, [2]University of Texas at Arlington,
[3]Emory University, [4]Amazon Inc.
{lijuanh1,hanhaoy1,chenzh85, liuhui7}@msu.edu, harry.shomer@uta.edu,
wei.jin@emory.edu, ajavari@amazon.com

Abstract

Session-based recommendation systems have attracted growing interest for their ability to provide personalized recommendations based on users' in-session behaviors. While ID-based methods have shown strong performance, they often struggle with long-tail items and overlook valuable textual information. To incorporate text information, various approaches have been proposed, generally employing a naive fusion framework. Interestingly, this approach often fails to outperform the best single-modality baseline. Further exploration indicates a potential imbalance issue in the naive fusion method, where the ID tends to dominate the training and the text is undertrained. This issue indicates that the naive fusion method might not be as effective in combining ID and text as once believed. To address this, we propose AlterRec, an alternative training framework that separates the optimization of ID and text to avoid the imbalance issue. AlterRec also designs an effective strategy to enhance the interaction between the two modalities, facilitating mutual interaction and more effective text integration. Extensive experiments demonstrate the effectiveness of AlterRec in session-based recommendation.

1 Introduction

In recent years, predicting the next item in user-item interaction sequences, such as clicks or purchases, has gained increasing attention (Wu et al., 2019b; Li et al., 2017; Pang et al., 2022; Hou et al., 2022a; Yang et al., 2023). These sequences, common in e-commerce, search engines, and media platforms, reflect user preferences that are dynamic and and evolve over time (Tahmasbi et al., 2021). Moreover, in many systems, only the user's behavior history during an ongoing session is accessible. Therefore, analyzing interactions in active sessions becomes essential for real-time recommendations. This need has spurred the development of session-based recommendations (Wu et al., 2019b; Hou et al., 2022a), which utilizes the sequential patterns in a session to understand and predict the latest user preferences.

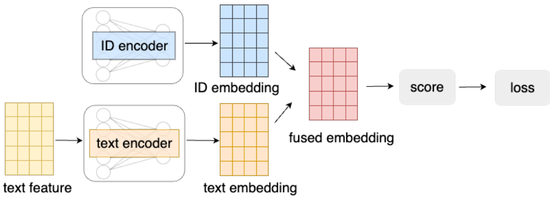


Figure 1: An illustration of a naive fusion framework.

In this domain, ID-based methods (Kang and McAuley, 2018; Sun et al., 2019; Wu et al., 2019b) have become the predominant approach, significantly influencing the recommendation paradigm (Yuan et al., 2023; Li et al., 2023). These methods involve assigning unique ID indexes to users and items, transforming them into vector representations. Their popularity stems from their simplicity and effectiveness across various applications (Li et al., 2023). Despite their proven effectiveness, these methods still have limitations. One drawback is their heavy reliance on the ID-based information and often ignore valuable information such as textual information, leading to less informative representations. It can be problematic in scenarios with limited interactions between users and items. However, most items experience sparse interactions, known as long-tail items (Park and Tuzhilin, 2008), which presents a significant challenge for these methods.

Recognizing these limitations, there has been a shift (Liu et al., 2025) towards integrating text data for recommendations. The surging volume of text data emphasizes the crucial role of text in various domains like news and e-commerce (Li et al., 2022; Wu et al., 2019a; Jin et al., 2023). These systems increasingly leverage user reviews, product descriptions, and articles to better capture user pref-

[*] This work was conducted when Amin was with The Home Depot.

Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 185–199
December 20-24, 2025 ©2025 Association for Computational Linguistics

erences. Recent trends indicate an increasing reliance on language models (Kenton and Toutanova, 2019; Brown et al., 2020; Wei et al., 2023; Harte et al., 2023) for extracting semantic information due to their exceptional ability to encode text effectively. This progress has sparked considerable interest in enhancing recommendation beyond traditional user-item interaction data.

The prevailing approach in current literature for combining ID and text typically employs a **naive fusion** framework (Hou et al., 2022b; Zhang et al., 2019; Wei et al., 2023; Chen et al., 2025), which merges embeddings from ID and text encoders for joint training (see Figure 1). However, our preliminary study in section 3.2 reveals that the naive fusion may not be as effective as previously believed. **1)** Notably, training on ID information alone can achieve comparable or even better performance than naive fusion. It indicates that fusion may not always enhance and could potentially degrade results. This finding aligns with the studies in multi-modal learning (Huang et al., 2022; Wang et al., 2020; Du et al., 2023a), which indicates that the fusion of multiple modalities doesn't always outperform the best single modality. **2)** We further explore one naive fusion implementation as an example to have a deeper understanding of this finding. It reveals a potential imbalance issue: the model over-relies on ID component while undertraining text. This imbalance implies that the unexpected finding might be a result of the naive fusion framework's inability to balance the contributions of the two types of information effectively, thereby hindering optimal overall performance.

The imbalance issue identified in the naive fusion significantly hinders the accurate integration of textual data. Despite increased efforts to integrate textual data, these methods often fail to effectively capture essential semantic information, resulting in a considerable loss of valuable information. This realization shifts our focus towards independent training, which does not exhibit this issue. However, independent training overlooks the potential for ID and text to provide complementary information that could benefit each other. To address these limitations, we propose a **Alter**native training strategy for session-based **Rec**ommendation (**AlterRec**). AlterRec separates the training of ID and text to mitigate imbalance while introducing implicit interactions between the two modalities. This design enables each to inform and learn from each other, and thus further enhance the performance. We conduct comprehensive experiments to validate the superior effectiveness of AlterRec over a variety of baselines in real-world datasets.

2 Related Work

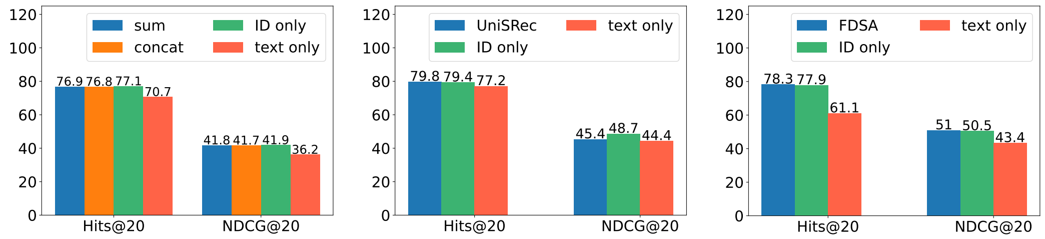
ID-based Methods. These methods (Hidasi et al., 2016) convert each user or item into a vector representation using unique ID indices. More recent advancements have seen the adoption of sophisticated architectures as encoders. For instance, SASRec (Kang and McAuley, 2018) and BERT4Rec (Sun et al., 2019) employ the Transformer architecture to delineate user preferences within sequences. SR-GNN (Wu et al., 2019b) and HG-GNN (Pang et al., 2022) construct graphs from the user-item interaction data to capture complex patterns across multiple sessions. However, these methods overlook additional valuable text information, potentially leading to less informative representations.

Text-Integrated Methods. These methods combine the text information to perform recommendations. For instance, FDSA (Zhang et al., 2019) leverages concatenation and S^3-Rec (Zhou et al., 2020) uses self-supervised tasks to combining textual information. UniSRec (Hou et al., 2022b) employs the BERT model, EAGER (Wang et al., 2024) uses the Sentence-T5 and LLM2BERT4Rec (Harte et al., 2023) uses the text feature from by the large language model (LLMs) as initialization. RLMRec (Ren et al., 2023) and LLMRec (Wei et al., 2023) both use LLMs for generating user/item profiles. Among the methods discussed, the majority follows the naive fusion framework (Zhang et al., 2019; Hou et al., 2022b; Wei et al., 2023), which may not effectively incorporate text as identified in the section 3.2.

3 Preliminaries

3.1 Session-based Recommendation

Consider a set of users $\mathcal{U}$ and items $\mathcal{V}$, with user-item interaction sequences (sessions) denoted by $\mathcal{S}$. Each session $\mathbf{s} = \{s_1, s_2, ..., s_n\} \in \mathcal{S}$ represents a sequence of item interactions by a user, where $s_i \in \mathcal{V}$ and n is the number of interactions. Each item i is associated with textual information, such as product descriptions or titles, represented as $t_i = \{w_1, w_2, ..., w_c\}$, where w_j is a word from a shared vocabulary and c is the truncated text length. The goal of session-based recommendation is to predict the next item in a session by generating a ranked



(a) NFRec (b) UnisRec (c) FDSA

Figure 2: Session-based recommendation results (%) on the Amazon-French dataset. We compare the models combing ID and text against models trained independently on either ID or text information alone.

score list for candidate items: $\mathbf{y_s} = [y_{\mathbf{s},1}, ..., y_{\mathbf{s},|\mathcal{V}|}]$, where each $y_{\mathbf{s},i}$ indicates the likelihood of item i being the next interaction.

The Naive Fusion Framework. In session-based recommendation, combining ID-based and text-based information has the potential to improve performance. Most existing methods adopt a naive fusion strategy with joint training (Hou et al., 2022b; Zhang et al., 2019; Wei et al., 2023), as shown in Figure 1. This involves generating embeddings $\mathbf{X}^{ID}$ and $\mathbf{X}^{text}$ using ID and text encoders, then merging them into a unified embedding $\mathbf{Z}$ via methods such as summation or concatenation. This final embedding is used to compute relevance scores between the session and candidate items, estimating the likelihood of the next interaction. We refer to this framework as **naive fusion**. Notably, existing methods such as UniSRec (Hou et al., 2022b), FDSA (Zhang et al., 2019), and LLMRec (Wei et al., 2023) follow this approach. We implemented a naive fusion method which is named NFRec and more details are in section A in the Appendix.

3.2 Preliminary Study

In this subsection, motivated by multi-modal learning (Wang et al., 2020; Huang et al., 2022; Peng et al., 2022), we conduct a preliminary study to investigate potential challenges in combining ID and text for session-based recommendation, aiming to inspire more effective integration strategies.

In the naive fusion, ID and text are treated as distinct modalities intended to complement each other. However, prior works (Wang et al., 2020; Du et al., 2023a; Huang et al., 2022) have revealed a consistent **phenomenon: fusing two modalities does not usually outperform the best single modality trained independently**. Namely, combining modalities may not enhance, and could potentially reduce, overall performance. Various multi-modal learning studies have focused on this phenomenon, offering analysis from different perspectives (Wu et al., 2022; Huang et al., 2022; Du et al., 2023b). To effectively merge ID and text for session recommendations, we conduct an investigation to first verify the presence of this phenomenon and then explore its underlying causes.

3.2.1 Naive Fusion vs. Independent Training

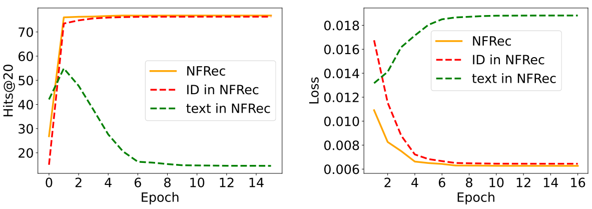
To verify whether the above phenomenon exists in session-based recommendation, we compare the performance of naive fusion models including our implementation NFRec, UniSRec (Hou et al., 2022b) and FDSA (Zhang et al., 2019) against their corresponding two single modality models (ID and text) that are trained independently. For a fair comparison, we employ same ID/text encoder, scoring and loss function across both naive fusions and independent training frameworks. More details are given in Section A of the Appendix.

The results on the Amazon-French dataset (see Section 5.1) are shown in Figure 2, where “ID only” and “text only” refer to respective ID and text models trained independently, and “sum” and “concat” represent NFRec using summation and concatenation respectively. We employ two widely used metrics Hits@20 and NDCG@20, where higher scores indicate better performance. We have the following observations (similar phenomenon is observed on the HD dataset in Figure 8 in the Appendix):

Observation 1 *The ID-only models often perform comparably to or better than naive fusion models, highlighting the ineffectiveness of naive fusion for combining ID and text.*

Observation 2 *The text-only model generally achieves the worst performance, often exhibiting a large gap compared to the ID-only approach.*

The first observation aligns with findings in multi-modal learning (Peng et al., 2022; Huang et al., 2022), suggesting that naive fusion is less effective than expected in session-based recommendation. To gain a more comprehensive understand-



(a) Amazon-French

(b) Amazon-French

Figure 3: Test performance (a) and training loss (b).

ing, we further explore this issue through an in-depth analysis of NFRec in the next subsection.

3.2.2 Exploration of NFRec

In our exploration, we aim to understand how ID and text components perform in NFRec, shedding light on why a naive fusion framework may not yield the expected improvement. To this end, we take NFRec applying concatenation as one example. It can be conceptually divided into two segments: the ID and text components (details in section B in the Appendix). Test performance and training loss on the Amazon-French dataset are shown in Figure 3, with the components labeled as “ID in NFRec" and “text in NFRec." Similar phenomenon is also observed on the HD dataset (Figure 9 in the Appendix).

Figure 3 highlights a clear imbalance issue in NFRec: the ID component’s performance and loss nearly overlap with those of NFRec, indicating a strong reliance on ID. The ID dominates the overall training while the text component contributes little. This suggests that naive fusion appears incapable of balancing the modalities to achieve optimal overall performance. It tends to overly depend on the stronger ID modality (as noted in the Observation 2). Supporting this hypothesis is from various studies (Peng et al., 2022; Huang et al., 2022; Wang et al., 2020; Wu et al., 2022) in multi-modal learning which offer empirical and theoretical insights. Further investigation into the underlying causes is left for future work.

4 Framework

Having identified the potential imbalance issue in naive fusion, we explore to combine ID and text by training them separately. However, simply training them independently may not fully exploit their potential to provide complementary strength. To address these challenges, we introduce AlterRec, an alternative training method illustrated in Figure 4. AlterRec consists of two uni-modal networks

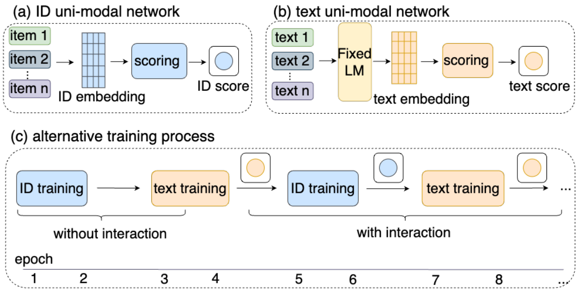


Figure 4: Overview of AlterRec. (a), (b): the ID and text uni-modal networks. (c): Two networks are trained alternately, learning from each other through predictions generated by the other network.

for ID and text. We employ the predictions from one network as training signals for the other, facilitating interaction and mutual learning through these predictions. AlterRec separates the training of ID and text, effectively avoiding the imbalance issue. Moreover, it goes beyond independent training by facilitating interaction between ID and text, enabling them to learn mutually beneficial information and incorporate the text more effectively.

4.1 ID and Text Uni-modal Networks

The ID and text unimodal networks share similar architectures, each using its own encoder to generate embeddings. Then a scoring function is adopted to calculate the relevance between a given session and candidate items. We first introduce the encoders and then show how to define the scoring function.

4.1.1 ID and Text Encoder

The ID encoder is designed to create a unique embedding for each item based on its ID index. This is achieved using an ID embedding matrix $\mathbf{X} \in \mathbb{R}^{|\mathcal{V}|\times d}$, where d is the size of the embedding. Each row, $\mathbf{X}_i$, corresponds to the ID embedding of item i. Notably, this matrix is a learnable parameter and updated during the optimization.

The text encoder is designed to extract semantic information from item text. Leveraging the advanced language modeling capabilities, we utilize the Sentence-BERT (Reimers and Gurevych, 2019) in this work. Given an item i with text $t_i = \{w_1, w_2, ..., w_c\}$, Sentence-BERT generates a sentence-level embedding, which is then projected into a d-dimensional space via an MLP (Delashmit et al., 2005):

$$\mathbf{H}_i = \text{MLP}(\text{SBERT}(w_1, w_2, ..., w_c)), \qquad (1)$$

where $\mathbf{H} \in \mathbb{R}^{|\mathcal{V}|\times d}$ and each row $\mathbf{H}_i$ corresponds to the text embedding of item i. Considering practical constraints, we fix the language model which isn't updated during the optimization process due to the high training cost.

4.1.2 Scoring Function

In session-based recommendation, the goal is to predict the next item in session $\mathbf{s} = \{s_1, s_2, ..., s_n\}$. For each candidate item j, we compute a prediction score $y_{\mathbf{s},j}$, used to rank candidates. The top-ranked item is predicted as the next item. This begins by generating a session embedding $\mathbf{q_s} \in \mathbb{R}^d$ that captures user behavior. Then relevance between the session embedding and each item embedding is used as the score for ranking.

Session&Item Relevance. The session embedding is computed from the embeddings of items in the session—either ID or text embeddings. Using ID embeddings as an example (the process is similar for text), we define a function g to generate the session embedding: $\mathbf{q_s} = g(\mathbf{X}_{s_1}, \mathbf{X}_{s_2}, ..., \mathbf{X}_{s_n})$. A simple yet effective choice is the **mean function**, which averages item embeddings. Alternatively, a **Transformer** (Vaswani et al., 2017) can be used to capture item-item transitions patterns, where the last item's output serves as the session embedding. The session embedding can be derived using either of these functions based on their empirical performance. Relevance scores are then computed via dot product: $y^{ID}_{\mathbf{s},i} = \mathbf{X}_i^T \mathbf{q_s}$ for ID, and $y^{text}_{\mathbf{s},i} = \mathbf{H}_i^T \hat{\mathbf{q}}\mathbf{s}$ for text, with $\hat{\mathbf{q}}_\mathbf{s} = g(\mathbf{H}_{s_1}, ..., \mathbf{H}_{s_n})$.

4.2 Alternative Training

The ID and text data offer different types of information. Our goal is to facilitate their interaction, enabling mutual learning and thereby enhancing overall performance. To this end, we propose an alternative training strategy to use predictions from one uni-modal network to train the other network. These predictions encode information of one modality, allowing one network to learn information from the other. We leverage the predictions from one modality to the other in two aspects. 1) First, we select top-ranked items as augmented positive training samples. These items with top scores are likely very relevant to the current session from the perspective of one modality that could provide more training signals for the other modality especially for items with fewer interactions. 2) Second, we choose other high-scored items as negative samples. These items are ranked higher but not the most relevant ones for one modality and we aim to force the other modality to distinguish them from positive samples. Such negative samples are much harder to be distinguished compared to those from traditional random sampling (Rendle et al., 2012). Thus, we refer to them as hard negative samples in this work.

Hard Negative Samples. As an example, we illustrate how predictions from the ID uni-modal network guide the training of the text uni-modal network. For a session s, we first generate prediction scores from the ID network and rank them in descending order: $r^{ID}_{\mathbf{s}} = \text{argsort}(y^{ID}_{\mathbf{s},1}, ..., y^{ID}_{\mathbf{s},|\mathcal{V}|})$, where $r^{ID}_{\mathbf{s}}$ gives the sequence of ID indices corresponding to the sorted scores. Items ranked from k_1 to k_2, denoted as $r^{ID}_{\mathbf{s}}[k_1 : k_2]$, are selected as hard negatives for training the text network. This encourages the text network to learn from patterns captured by the ID network. These hard negatives are used in a cross-entropy loss function, following prior work (Hou et al., 2022b; Wu et al., 2019b; Pang et al., 2022), where s_t is the target item:

$$L^{text} = -\sum_{\mathbf{s}\in\mathcal{S}} \log(f(y^{text}_{\mathbf{s},s_t})) \tag{2}$$

where f is the Softmax function applied over the target item s_t and the hard negatives $r^{ID}_{\mathbf{s}}[k_1 : k_2]$. Similarly, we can derive hard negatives $r^{text}_{\mathbf{s}}[k_1 : k_2]$ from the text uni-modal network by sorting its prediction scores: $r^{text}_{\mathbf{s}} = \text{argsort}(y^{text}_{\mathbf{s},1}, ..., y^{text}_{\mathbf{s},|\mathcal{V}|})$. These are then used to define the loss for training the ID uni-modal network:

$$L^{ID} = -\sum_{\mathbf{s}\in\mathcal{S}} \log(f(y^{ID}_{\mathbf{s},s_t})) \tag{3}$$

where the Softmax is applied over the target item s_t and the negative samples in $r^{text}_{\mathbf{s}}[k_1 : k_2]$.

Positive Sample Augmentation. Besides hard negatives, we can select top-ranked items as augmented positive samples to further improve both uni-modal networks. For the text network, $r^{ID}_{\mathbf{s}}[1 : p]$ serves as additional positive targets; similarly, $r^{text}_{\mathbf{s}}[1 : p]$ is used as supplementary positive samples for training the ID network. We typically set $p < k_1$. The corresponding loss functions in Eq. (2) and (3) are modified accordingly for this variation:

$$\begin{aligned} L^{text}_a = -\sum_{\mathbf{s}\in\mathcal{S}} \Bigg(& \log(f(y^{text}_{\mathbf{s},s_t})) + \\ & \beta * \sum_{s_k \in r^{ID}_{\mathbf{s}}[1:p]} \log(f(y^{text}_{\mathbf{s},s_k})) \Bigg) \end{aligned} \tag{4}$$

$$L_a^{ID} = -\sum_{\mathbf{s}\in\mathcal{S}} \Bigg(\log(f(y_{\mathbf{s},s_t}^{ID})) + \beta * \sum_{s_k \in r_{\mathbf{s}}^{text}[1:p]} \log(f(y_{\mathbf{s},s_k}^{ID})) \Bigg) \quad (5)$$

Here, β is a parameter to adjust the importance of the augmented samples. Note that within each network, these augmented samples are paired with the same corresponding hard negative samples as the target item s_t.

Training Algorithm. This algorithm focuses on facilitating the interaction between two networks, and we use the Figure 4(c) as a more straightforward illustration. The training process consists of two stages. **1) Initially**, due to the lower quality of the learned embeddings, we don't employ interaction between two networks. Thus, we apply random negative samples during the first m_{random} epochs. This involves replacing the hard negatives in Eq. (2) and Eq. (3) with randomly selected negatives with equal number. **2) Subsequently**, we shift to training with hard negatives. We start by training the ID uni-modal network using hard negatives derived from the text uni-modal network. After m_{gap} epochs, the training focus shifts to the text uni-modal network, which is trained using hard negatives from the ID uni-modal network. Following another m_{gap} epochs, we resume training the ID uni-modal network and repeat this alternating process. This approach ensures that each network continually learns from the other, thereby potentially improving overall performance. Pseudo-code of the training process is given in section C in the Appendix.

After both networks converge, we compute the final relevance score by combining their scores with weighted contributions. This score is used during the **inference stage** and is defined as:

$$y_{\mathbf{s},i} = \alpha * y_{\mathbf{s},i}^{ID} + (1-\alpha) * y_{\mathbf{s},i}^{text} \quad (6)$$

Here, $y_{\mathbf{s},i}$ is the final score for the candidate item i given session s, and α is a pre-defined parameter.

5 Experiment

In this section, we conduct comprehensive experiments to validate the performance of AlterRec. Model performance is evaluated using Hits@K and NDCG@K, with K set to 10 and 20. Higher scores indicate better performance. More implementation details are given in the section F in the Appendix.

5.1 Experimental Settings

Datasets. We adopt two real-world session recommendation datasets including textual data. **HD**: It is from an e-commerce company that is derived from user purchase logs on its website. **Amazon-M2** (Jin et al., 2023): It's a multilingual dataset. For the purpose of this study, which does not focus on multilingual data, we extracted unilingual sessions to create individual datasets for three languages: Spanish, French, and Italian. They are denoted as **Amazon-Spanish**, **Amazon-French**, and **Amazon-Italian**, respectively. More details are given in section E in the Appendix.

Baselines. In our study, we refer to the model without augmentation as **AlterRec** and the augmented version as **AlterRec_aug**. We include several baseline methods: **BSARec** (Shin et al., 2024), **CORE** (Hou et al., 2022a), **SASRec** (Kang and McAuley, 2018), **BERT4Rec** (Sun et al., 2019), **SR-GNN** (Wu et al., 2019b), and **HG-GNN** (Pang et al., 2022) as ID-based methods. Text-integrated methods include **LLM2BERT4Rec** (Harte et al., 2023),**UniSRec** (Hou et al., 2022b), **FDSA** (Zhang et al., 2019), and **S³-Rec** (Zhou et al., 2020). Notably, **UniSRec (FHCKM)** refers to the model pretrained on the FHCKM dataset (Hou et al., 2022b), while **UniSRec** in this work denotes the model pretrained on our datasets (HD and Amazon-M2). LLM2BERT4Rec uses BERT4Rec as a backbone model, and we also test **LLM2SASRec**, which uses SASRec. To ensure fairness, each baseline method uses the same input features as AlterRec, except for UniSRec (FHCKM), which is pretrained with fixed dimension sizes.

5.2 Performance Comparison

The comparison results are presented in Table 1. Since the Amazon-M2 dataset lacks user information, it is not feasible to obtain results for HG-GNN (Pang et al., 2022), which are denoted as "N/A". Our observations are as follows: 1) Alter_aug consistently outperforms other baseline models across a range of datasets, with AlterRec often achieving the second-best performance, hightlighting the effectiveness of our alternative training strategy. Moreover, it demonstrates that integrating augmentation data can further enhance performance. Although UniSRec and FDSA exhibit strong performance in some cases, they do not consistently excel across all metrics. In contrast, AlterRec maintains a balanced and superior

Table 1: Performance Comparison (%) results which are mean and standard deviation over three seeds. The best results are highlighted in bold, and the second-best results are underlined.

	HD				Amazon-Spanish			
	Hits@10	Hits@20	NDCG@10	NDCG@20	Hits@10	Hits@20	NDCG@10	NDCG@20
SASRec	33.58 ± 0.27	40.93 ± 0.14	18.23 ± 0.06	20.09 ± 0.06	70.95 ± 0.32	80.46 ± 0.32	44.88 ± 0.33	47.29 ± 0.34
BERT4Rec	26.06 ± 0.26	31.85 ± 0.45	15.61 ± 0.3	17.08 ± 0.35	64.6 ± 0.13	74.0 ± 0.33	44.6 ± 0.16	46.98 ± 0.18
BSARec	35.02 ± 0.21	42.08 ± 0.08	19.65 ± 0.03	21.44 ± 0.07	71.69 ± 0.09	80.89 ± 0.05	48.16 ± 0.77	50.5 ± 0.76
SRGNN	30.09 ± 0.07	36.0 ± 0.19	15.31 ± 0.13	15.73 ± 0.13	67.02 ± 0.29	76.37 ± 0.12	46.75 ± 0.33	49.12 ± 0.26
HG-GNN	33.17 ± 0.13	40.72 ± 0.20	18.27 ± 0.49	20.19 ± 0.51	N/A	N/A	N/A	N/A
CORE	37.04 ± 0.11	44.73 ± 0.06	19.86 ± 0.14	21.81 ± 0.14	71.83 ± 0.15	81.14 ± 0.17	41.05 ± 0.06	43.41 ± 0.08
UnisRec (FHCM)	36.03 ± 0.12	43.67 ± 0.06	20.14 ± 0.79	22.08 ± 0.77	72.15 ± 0.01	81.3 ± 0.02	44.87 ± 0.1	47.2 ± 0.1
UnisRec	34.56 ± 0.23	42.19 ± 0.16	19.01 ± 0.08	20.92 ± 0.08	72.33 ± 0.06	81.42 ± 0.16	45.51 ± 0.05	47.82 ± 0.06
FDSA	32.1 ± 0.34	39.11 ± 0.2	20.44 ± 0.1	22.21 ± 0.06	70.55 ± 0.24	79.84 ± 0.08	49.83 ± 0.15	52.18 ± 0.13
S^3-Rec	26.69 ± 0.1	33.04 ± 0.34	16.01 ± 0.14	17.62 ± 0.11	69.61 ± 0.4	78.85 ± 0.62	47.25 ± 0.45	49.6 ± 0.4
LLM2SASRec	34.12 ± 0.29	42.13 ± 0.18	18.69 ± 0.26	20.72 ± 0.22	71.55 ± 0.06	80.68 ± 0.12	48.45 ± 0.15	50.77 ± 0.17
LLM2BERT4Rec	29.51 ± 0.35	37.3 ± 0.33	16.25 ± 0.3	18.22 ± 0.3	66.47 ± 0.2	76.95 ± 0.27	40.29 ± 0.32	42.95 ± 0.35
AlterRec	38.25 ± 0.14	46.31 ± 0.11	20.72 ± 0.06	**22.76 ± 0.06**	72.41 ± 0.17	**81.49 ± 0.09**	**50.59 ± 0.14**	**52.9 ± 0.12**
AlterRec_aug	**38.46 ± 0.1**	**46.37 ± 0.08**	**20.74 ± 0.05**	22.75 ± 0.03	**72.47 ± 0.19**	81.45 ± 0.04	50.58 ± 0.02	52.86 ± 0.05
	Amazon-French				Amazon-Italian			
	Hits@10	Hits@20	NDCG@10	NDCG@20	Hits@10	Hits@20	NDCG@10	NDCG@20
SASRec	69.2 ± 0.15	78.4 ± 0.1	44.89 ± 0.43	47.23 ± 0.44	68.25 ± 0.08	78.37 ± 0.06	43.24 ± 0.18	45.81 ± 0.18
BERT4Rec	63.01 ± 0.11	72.47 ± 0.15	43.84 ± 0.04	46.24 ± 0.07	62.24 ± 0.26	72.38 ± 0.13	42.42 ± 0.15	44.99 ± 0.11
BSARec	69.90 ± 0.13	79.04 ± 0.12	47.36 ± 0.36	49.69 ± 0.35	69.45 ± 0.04	79.27 ± 0.06	45.79 ± 0.43	48.28 ± 0.45
SRGNN	65.61 ± 0.09	74.93 ± 0.09	46.27 ± 0.1	48.64 ± 0.08	65.62 ± 0.26	75.2 ± 0.15	44.85 ± 0.17	47.28 ± 0.15
CORE	69.93 ± 0.02	79.32 ± 0.1	39.4 ± 0.05	41.79 ± 0.07	69.42 ± 0.12	79.4 ± 0.1	39.27 ± 0.05	41.8 ± 0.05
UniSRec (FHCM)	70.35 ± 0.04	79.73 ± 0.13	43.99 ± 0.12	46.37 ± 0.1	69.95 ± 0.06	79.84 ± 0.07	42.97 ± 0.18	45.48 ± 0.2
UniSRec	70.54 ± 0.09	79.74 ± 0.03	44.5 ± 0.06	46.84 ± 0.06	69.99 ± 0.07	79.63 ± 0.03	43.42 ± 0.08	45.87 ± 0.06
FDSA	68.94 ± 0.29	78.16 ± 0.13	48.62 ± 0.11	50.96 ± 0.08	67.88 ± 0.07	77.97 ± 0.11	47.04 ± 0.11	49.6 ± 0.12
S^3-Rec	62.82 ± 1.78	72.85 ± 1.01	40.84 ± 2.57	43.39 ± 2.37	60.6 ± 2.92	71.67 ± 2.19	37.88 ± 3.42	40.69 ± 3.22
LLM2SASRec	70.01 ± 0.1	79.15 ± 0.08	48.13 ± 0.08	50.45 ± 0.11	69.2 ± 0.14	79.11 ± 0.06	46.22 ± 0.39	48.73 ± 0.4
LLM2BERT4Rec	65.48 ± 0.02	75.91 ± 0.08	39.8 ± 0.16	42.45 ± 0.15	64.88 ± 0.44	75.9 ± 0.14	31.23 ± 0.26	32.0 ± 0.24
AlterRec	70.61 ± 0.03	79.75 ± 0.07	49.53 ± 0.02	**51.86 ± 0.01**	69.98 ± 0.01	79.75 ± 0.05	**47.87 ± 0.14**	**50.35 ± 0.14**
AlterRec_aug	**70.82 ± 0.09**	**79.84 ± 0.1**	**49.56 ± 0.06**	**51.86 ± 0.07**	**70.13 ± 0.03**	**79.86 ± 0.11**	**47.87 ± 0.13**	50.34 ± 0.15

performance in both Hits@N and NDCG@N. For instance, AlterRec shows about a 10% relative improvement over UniSRec based on NDCG@10 and NDCG@20 on Amazon-M2 datasets. Additionally, it achieves approximately 19% and 2% relative improvements over FDSA based on Hits@10 and Hits@20 on the HD and Amazon-M2 datasets. 2) Models incorporating text data, like AlterRec, UniSRec, and FDSA, generally outperform ID-based models, indicating that text information offers complementary benefits and enhances overall performance.

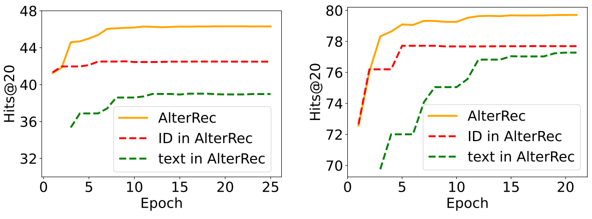
Table 2: Ablation study on key components. Reported results are mean value over three seeds.

	HD		Amazon-French	
Methods	Hits@10	Hits@20	Hits@10	Hits@20
AlterRec	**38.25**	**46.31**	**70.61**	**79.75**
AlterRec_random	37.41	45.41	70.46	79.64
AlterRec_w/o_text	35.64	42.95	68.26	77.23
AlterRec_w/o_ID	30.05	38.73	66.96	76.85

5.3 Ablation Study

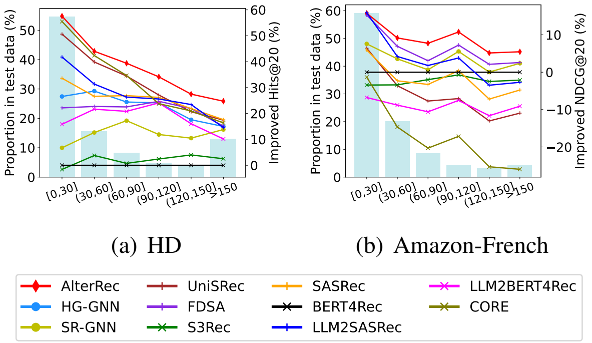
In this subsection, we evaluate the effectiveness of key components in our model: hard negative samples and the ID and text uni-modal networks. Table 2 presents the ablation study results for the following model variants: “AlterRec_random" for training with random negative samples, “AlterRec_w/o_text" for the model without the text uni-modal network, and “AlterRec_w/o_ID" for the model excluding the ID uni-modal network. AlterRec_w/o_text and AlterRec_w/o_ID are trained on a single modality.

The results in Table 2 show that using random negative samples hurts performance, as it behave likes independent training and lacks interaction between the two modalities. This underscores the effectiveness of AlterRec over independent training, which benefits from hard negative samples to enhance learning between the two uni-modal networks. Furthermore, AlterRec significantly outperforms model variants that rely only on ID information, i.e., AlterRec_w/o_text. For instance, AlterRec achieves relative improvements of 7.82% and 3.26% in terms of Hits@20 on the HD and Amazon-French datasets, respectively. These findings highlight AlterRec's superior ability to integrate text information over naive fusion methods.



(a) HD (b) Amazon-French

Figure 5: Test results during the alternative training.



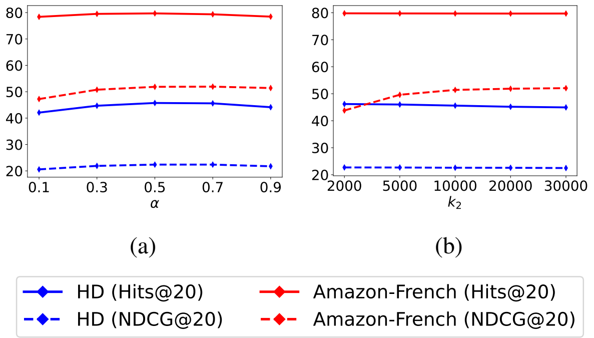
(a) HD (b) Amazon-French

Figure 6: Performance comparison w.r.t. long-tail items. The bar graph depicts the proportion of sessions in the test data for each group. The line chart illustrates the improvement ratios for Hits@20 and NDCG@20 relative to BERT4Rec.

Additionally, we present the performance of AlterRec in Figure 5, including the individual performance of the ID and text components within AlterRec across epochs. These components are denoted as "ID in AlterRec" and "text in AlterRec", respectively. The overall performance of AlterRec is based on the score $y_{\mathbf{s},i}$ in Eq. (6). The performance of "ID in AlterRec" and "text in AlterRec" are derived from the scores $y_{\mathbf{s},i}^{ID}$ and $y_{\mathbf{s},i}^{text}$ within $y_{\mathbf{s},i}$. Figure 5 demonstrates that both ID and text components are effectively trained in our model, and crucially, AlterRec does not exhibit the imbalance issue commonly associated with naive fusion.

5.4 Performance on Long-tail Items

Textual data offers valuable semantic information that can be used to enhance long-tail items in session-based recommendation. To validate this, we divide the test data into groups based on the popularity of the ground-truth item in the training data. We then compare the performance of various methods in each group against the ID-based method BERT4Rec. The comparative result is presented in Figure 6, where we also show the proportion of each group. This figure reveals that a majority of items have sparse interactions (long-tail items). In most cases, AlterRec outperforms other baselines particularly on long-tail items. For instance, AlterRec achieves the best performance in the [0,30] group on the HD and Amazon-French. It indicates that AlterRec effectively captures textual information, enhancing its performance on long-tail items.



(a) (b)

Figure 7: Performance by varying α and k_2.

5.5 Parameter Analysis

In this subsection, we analyze the sensitivity of two key hyper-parameters: the parameter α which adjusts the contribution of ID and text scores in Eq. (6), and the end index k_2 used for selecting hard negative samples as discussed in Section 4.2. More parameter analysis (k_1, p, and β) are reported in section G in the Appendix which show stable impact on the performance. The results for Hits@20 and NDCG@20 on HD and Amazon-French are presented in Figure 7. Regarding α, an increase in performance is observed as α rises from 0.1 to 0.5, followed by a decrease when α is increased from 0.5 to 0.9. This suggests that an α value of 0.5 typically yields the best performance, indicating equal contributions from ID and text. For k_2, there is an increasing trend in NDCG@20 on Amazon-French and a decreasing trend in Hits@20 on HD as k_2 increases. This indicates that the Amazon-French may benefit from relatively more hard negatives, whereas HD does not require as many.

6 Conclusion

In this work, we propose AlterRec, an effective approach for combining ID and text information in session-based recommendation. We first identify an imbalance issue in the commonly used naive fusion framework, which limits the integration of textual information. To overcome this challenge, AlterRec trains ID and text component independently and uses an alternative training strategy that enables implicit interactions between them. By leveraging hard negatives and augmented positives from one network to train the other, AlterRec mitigates the imbalance issue and facilitates mutual learning to enhance overall performance. Extensive experiments on multiple datasets confirm its effectiveness against state-of-the-art baselines. In future work, we aim to explore more advanced text encoders, such as LLaMA, within this framework.

7 Limitations

Our investigation builds upon the naive fusion method to combine ID and text information, without exploring other fusion strategies. Moreover, due to resource constraints, we did not conduct an extensive investigation into different text encoders, which could be important to integrate the text information.

8 Acknowledgements

This work is supported by the National Science Foundation (NSF) under grant numbers CNS2321416, IIS2212032, IIS2212144, IIS 2504089, DUE2234015, CNS2246050, DRL2405483 and IOS2035472, the Michigan Department of Agriculture and Rural Development, US Dept of Commerce, Amazon Faculty Award, Meta, NVIDIA, Microsoft, SNAP and The Home Depot.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Lei Chen, Chen Gao, Xiaoyi Du, Hengliang Luo, Depeng Jin, Yong Li, and Meng Wang. 2025. Enhancing id-based recommendation with large language models. *ACM Transactions on Information Systems*, 43(5):1–30.

Walter H Delashmit, Michael T Manry, and 1 others. 2005. Recent developments in multilayer perceptron neural networks. In *Proceedings of the seventh annual memphis area engineering and science conference, MAESC*, pages 1–15.

Chenzhuang Du, Jiaye Teng, Tingle Li, Yichen Liu, Tianyuan Yuan, Yue Wang, Yang Yuan, and Hang Zhao. 2023a. On uni-modal feature learning in supervised multi-modal learning. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 8632–8656. PMLR.

Chenzhuang Du, Jiaye Teng, Tingle Li, Yichen Liu, Tianyuan Yuan, Yue Wang, Yang Yuan, and Hang Zhao. 2023b. On uni-modal feature learning in supervised multi-modal learning. *arXiv preprint arXiv:2305.01233*.

Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*, pages 1096–1102.

Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

Yupeng Hou, Binbin Hu, Zhiqiang Zhang, and Wayne Xin Zhao. 2022a. Core: simple and effective session-based recommendation within consistent representation space. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pages 1796–1801.

Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022b. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 585–593.

Yu Huang, Junyang Lin, Chang Zhou, Hongxia Yang, and Longbo Huang. 2022. Modality competition: What makes joint training of multi-modal network fail in deep learning?(provably). In *International Conference on Machine Learning*, pages 9226–9259. PMLR.

Wei Jin, Haitao Mao, Zheng Li, Haoming Jiang, Chen Luo, Hongzhi Wen, Haoyu Han, Hanqing Lu, Zhengyang Wang, Ruirui Li, and 1 others. 2023. Amazon-m2: A multilingual multi-locale shopping session dataset for recommendation and text generation. *arXiv preprint arXiv:2307.09688*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Jian Li, Jieming Zhu, Qiwei Bi, Guohao Cai, Lifeng Shang, Zhenhua Dong, Xin Jiang, and Qun Liu. 2022. Miner: multi-interest matching network for news recommendation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 343–352.

Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 1419–1428.

Ruyu Li, Wenhao Deng, Yu Cheng, Zheng Yuan, Jiaqi Zhang, and Fajie Yuan. 2023. Exploring the upper limits of text-based collaborative filtering using large language models: Discoveries and insights. *arXiv preprint arXiv:2305.11700*.

Qidong Liu, Jiaxi Hu, Yutian Xiao, Xiangyu Zhao, Jingtong Gao, Wanyu Wang, Qing Li, and Jiliang Tang. 2025. Multimodal recommender systems: A survey. *ACM Comput. Surv.*

Yitong Pang, Lingfei Wu, Qi Shen, Yiming Zhang, Zhihua Wei, Fangli Xu, Ethan Chang, Bo Long, and Jian Pei. 2022. Heterogeneous global graph neural networks for personalized session-based recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, pages 775–783.

Yoon-Joo Park and Alexander Tuzhilin. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, pages 11–18.

Xiaokang Peng, Yake Wei, Andong Deng, Dong Wang, and Di Hu. 2022. Balanced multimodal learning via on-the-fly gradient modulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8238–8247.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation learning with large language models for recommendation. *arXiv preprint arXiv:2310.15950*.

Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*.

Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. 2024. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 8984–8992.

Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, pages 1441–1450.

Hamidreza Tahmasbi, Mehrdad Jalali, and Hassan Shakeri. 2021. Modeling user preference dynamics with coupled tensor factorization for social media recommendation. *Journal of Ambient Intelligence and Humanized Computing*, 12:9693–9712.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Weiyao Wang, Du Tran, and Matt Feiszli. 2020. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12695–12705.

Ye Wang, Jiahao Xun, Minjie Hong, Jieming Zhu, Tao Jin, Wang Lin, Haoyuan Li, Linjun Li, Yan Xia, Zhou Zhao, and 1 others. 2024. Eager: Two-stream generative recommender with behavior-semantic collaboration. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3245–3254.

Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Llmrec: Large language models with graph augmentation for recommendation. *arXiv preprint arXiv:2311.00423*.

Chuhan Wu, Fangzhao Wu, Suyu Ge, Tao Qi, Yongfeng Huang, and Xing Xie. 2019a. Neural news recommendation with multi-head self-attention. 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)*, pages 6389–6394.

Nan Wu, Stanislaw Jastrzebski, Kyunghyun Cho, and Krzysztof J Geras. 2022. Characterizing and overcoming the greedy nature of learning in multi-modal deep neural networks. In *International Conference on Machine Learning*, pages 24043–24055. PMLR.

Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019b. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 346–353.

Zhengyi Yang, Jiancan Wu, Zhicai Wang, Xiang Wang, Yancheng Yuan, and Xiangnan He. 2023. Generate what you prefer: Reshaping sequential recommendation via guided diffusion. *Advances in Neural Information Processing Systems*, 36:24247–24261.

Zheng Yuan, Fajie Yuan, Yu Song, Youhua Li, Junchen Fu, Fei Yang, Yunzhu Pan, and Yongxin Ni. 2023. Where to go next for recommender systems? id- vs. modality-based recommender models revisited. *arXiv preprint arXiv:2303.13835*.

Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, and 1 others. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, pages 4320–4326.

Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*, pages 1893–1902.

A Implementation of Naive Fusion

In this section, we give more details of the naive fusion methods in section 3.2.1. We explore three approaches: our own implementation NFRec, UnisRec (Hou et al., 2022b), and FDSA (Zhang et al., 2019), with detains provided in the following.

- **NFRec**: It consists of several key components. We give more details of these components. **ID and text encoder**: We employ the same ID and text encoder as AlterRec which is introduced in section 4.1.1 and section 4.1.1, respectively. Through these two encoders, we obtain the item-level ID embedding $\mathbf{X}$ and text embeddings $\mathbf{H}$. **Fusion operation**: We fuse the ID and text item embeddings to form a final embedding $\mathbf{Z}$ via summation or concatenation to as mentioned in section 3.2.1. **Scoring function**: For a given session s, we apply the mean function based on the fused item embedding to get the session embedding $\mathbf{q_s} = g(\mathbf{s}, \mathbf{Z})$, and then we use the vector multiplication between session embedding and the candidate item's fused embedding to get the score $y_{\mathbf{s},i} = \mathbf{Z_i}^T \mathbf{q_s}$. **Loss function**: We use the cross entropy as the loss function, which follows similar form with Eq. (3).

- **UniSRec** (Hou et al., 2022b): The model employs the same ID encoder as NFRec, which utilizes a learnable embedding for the ID representation. For text encoder, it leverages a language model enhanced by the proposed adaptor to extract textual information. After pretraining the adaptor using two contrastive loss functions, it merges the ID and text embeddings through summation. UniSRec adopts the same cross-entropy loss function as used in NFRec. We use the official code of UnisRec [1] as the implementation.

- **FDSA** (Zhang et al., 2019): The ID encoder generates ID embeddings using learnable embeddings. The text encoder employs a MLP and an attention mechanism to produce text embeddings. The Transformer is applied to items within a session to create ID and text session embeddings, which are then concatenated to form the final session embedding. Similar with NFRec and UniSRec, FDSA utilizes cross-entropy as the loss function. For the implementation of FDSA, we utilize code from the UniSRec's repository, which includes the implementation details for FDSA.

B More Details when exploring NFRec

In this section, we give more details for the exploration conducted in section 3.2.2. We elucidate the process of dividing the NFRec into its ID and text components, and describe how we evaluate the performance and obtain the loss of "ID in NFRec" and "text in NFRec." Details on the implementation of the NFRec are provided in the Appendix A.

For any given item i, we derive the ID embedding $\mathbf{X}_i$ and text embedding $\mathbf{H}_i$ from the corresponding ID and text encoders. These two embeddings are then concatenated to form a final embedding $\mathbf{Z}_i = [\mathbf{X}_i, \mathbf{H}_i]$. For a session $\mathbf{s} = \{s_1, s_2, ..., s_n\}$, the session embedding is obtained by applying the mean function to the final embeddings of the items within session $\mathbf{s}$: $\mathbf{q_s} = g_{mean}(\mathbf{s}, \mathbf{Z})$. This session embedding is represented as a concatenation of two parts derived from the ID and text embeddings, respectively:

$$\begin{aligned}\mathbf{q_s} &= [\mathbf{q_s}^{ID}, \mathbf{q_s}^{text}] \\ &= [\frac{1}{|\mathbf{s}|}\sum_{s_i \in \mathbf{s}} \mathbf{X}_{s_i}, \frac{1}{|\mathbf{s}|}\sum_{s_i \in \mathbf{s}} \mathbf{H}_{s_i}] \end{aligned} \tag{7}$$

The relevance score between a session and an item is then decomposed into two parts:

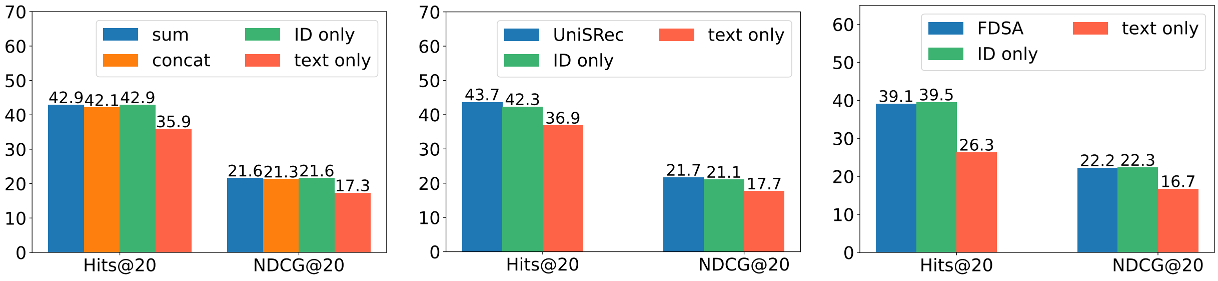
$$\begin{aligned} y_{s,i} &= \mathbf{Z}_i^T \mathbf{q}_s \\ &= [\mathbf{X}_i, \mathbf{H}_i]^T [\mathbf{q_s}^{ID}, \mathbf{q_s}^{text}] \\ &= \mathbf{X}_i^T \mathbf{q_s}^{ID} + \mathbf{H}_i^T \mathbf{q_s}^{text} \\ &= y_{s,i}^{ID} + y_{s,i}^{text} \end{aligned} \tag{8}$$

Thus, the relevance score in NFRec can be decomposed as the summation of the ID and text scores. Accordingly, we evaluate the performance and obtain the loss of "NFRec", "ID in NFRec" and "text in NFRec" based on $y_{s,i}$, $y_{s,i}^{ID}$ and $y_{s,i}^{text}$ in Eq. (8), respectively. For the loss function, the cross-entropy is employed.

C Alternative training algorithm

We present the pseudo code of the alternative training algorithm in Algorithm 1. The parameters within the ID and text uni-modal networks are denoted as θ^{ID} and θ^{text}, respectively. Initially, as indicated in line 1, both networks are randomly

[1] https://github.com/RUCAIBox/UniSRec/tree/master



(a) NFRec

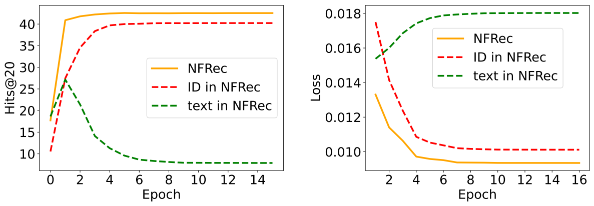
(b) UnisRec

(c) FDSA

Figure 8: Session recommendation results (%) on the HD dataset. We compare the models combing ID and text against models trained independently on either ID or text information alone.

Algorithm 1 Alternative Training

```
Require: User-item interaction set $\mathcal{S}$, epoch number using random negatives $m_{random}$, maximum epoch number $m_{max}$, gap epoch number $m_{gap}$
Ensure: Converged models $\theta^{ID}$, $\theta^{text}$
 1: Random initialize two uni-modal networks $\theta^{ID}$, $\theta^{text}$
 2: for i = 1, 2, ..., $m_{random}$ do
 3:     Train $\theta^{ID}$ using random negatives
 4: end for
 5: for i = 1, 2, ..., $m_{random}$ do
 6:     Train $\theta^{text}$ using random negatives
 7: end for
 8: for i = 0, 1, ..., $m_{max} - 2 * m_{random}$ do
 9:     if $i \mod (2 * m_{gap}) < m_{gap}$ then
10:         Compute loss in Eq. (3)
11:         Update $\theta^{ID} : \theta^{ID} \leftarrow \theta^{ID} - \alpha \nabla L^{ID}$
12:     else
13:         Compute loss in Eq. (2)
14:         Update $\theta^{text} : \theta^{text} \leftarrow \theta^{text} - \alpha \nabla L^{text}$
15:     end if
16: end for
```



(a) HD

(b) HD

Figure 9: Test performance in terms of Hits@20 (%) and training loss comparison on the HD dataset.

initialized. In the early stages of training, both networks are trained with random negative samples, as indicated in line 2-7. It's because the embedding learned in the early stage are of lower quality and might not be able to provide useful information. As training progresses, we shift towards employing hard negative samples. At first, the ID unimodal network is trained using predictions from the text unimodal network, as described in lines 9 to 11. After m_{gap} epochs, training shifts to the text unimodal network, utilizing predictions from the ID unimodal network, as indicated in lines 12 to 15. Subsequently, training alternates back to the ID network. This cycle continues until convergence is achieved for both networks. Notably, for AlterRec_aug, we replace the loss function in line 10 and 13 as Eq. (5) and Eq. (4) respectively.

D Additional Results in Preliminary Study

Additional results on the HD dataset for investigations in sections 3.2.1 are displayed in Figure 8 and Figure 9, respectively. These figures indicate a trend similar to that observed with the

Amazon-French dataset. Specifically, Figure 8 reveals that models trained independently on ID data can achieve performance comparable to, or even surpassing, that of naive fusion methods. Furthermore, models relying solely on text information tend to perform the worst. In Figure 9, it is observed that the ID component dominates the performance and loss. These findings are consistent with observations made with the Amazon-French dataset, suggesting that the phenomenon identified in observations 1 and 2 in section 3.2.1, as well as the imbalance issue in NFRec, may be prevalent across various datasets.

Table 3: Data statistic of the session datasets. The Amazon-M2 datasets don't involve users. #Train, #Val, and #Test denote the number of sessions in the train, validation, and test.

Dataset	#User	#Item	#Train	#Val	#Test
HD	145,750	39,114	182,575	2,947	5,989
Amazon-Spanish	-	38,888	75,098	7,900	6,237
Amazon-French	-	40,258	96,245	10,507	8,981
Amazon-Italian	-	45,559	102,923	11,102	10,158

E Datasets

We provide the data statistics in Table 3. The **HD** dataset is a sampled dataset of purchase logs from the an e-commerce company's website. We include sessions where all items have textual data, i.e., titles, descriptions, and taxonomy. Items in each session is interacted by the same user, who may have engaged in several sessions at distinct timestamps. For the purposes of validation and testing, we select the most recent sessions from different users. Specifically, 10% of these sessions are designated for validation and 20% for testing, with the remainder allocated to training sessions. Typically, the sessions in validation appear after those in the training set, and the sessions for testing appear after those in the validation set. For the three **Amazon-M2** datasets, since there is no original validation set, we use about 10% of the training set to create a validation set.

The Amazon-M2 dataset is publicly available [2]. However, due to strict company regulations, we are unable to release the HD dataset. It's protected by confidentiality agreements and data protection policies designed to preserve sensitive business information and customer privacy. Although the insights gained from this data are crucial to our research, we must comply with these restrictions to meet legal and ethical standards for data usage and sharing. Therefore, we can only present summaries and aggregated findings instead of providing the raw datasets.

F Experimental Settings

Empirically, for the HD dataset, we use the mean function for ID session embedding and Transformer for text session embedding. For Amazon-M2, Transformer is used for both ID and text session embeddings. We set the parameters as follows: $m_{random} = 2, m_{gap} = 2, m_{max} = 30$, $\alpha = 0.5$, $\beta = 0.5$, $p = 5$. Additionally, for HD, we set $k_1 = 6$, $k_2 = 2000$, and for the Amazon-M2 datasets, $k_1 = 20$, $k_2 = 20000$ are used.

In our experimental setup, we search the learning rate in $\{0.01, 0.001\}$ and dropout in $\{0.1, 0.3, 0.5\}$, and we set hidden dimension as 300, and number of Transformer layer to be 2, for all models. The test results we report are based on the model that achieves the best performance during the validation phase. For text feature extraction in the HD dataset, we utilize Sentence-BERT with the all-MiniLM-L6-v2 model[3]. In contrast, for the three Amazon-M2 datasets, we employ Sentence-BERT with the distiluse-base-multilingual-cased-v1 model[4], due to its proficiency in handling multiple languages including Spanish, French, and Italian. For each item in the HD dataset, we use title, description, and taxonomy as the textual data. For the Amazon-M2 datasets, we use the title and description as textual data. All baseline methods employ the cross entropy as loss function and are implemented based on the RecBole [5]. All the experiments are conducted on the NVIDIA L4 GPUs with 24Gb.

We adopt two metrics Hits@K and NDCG@K which are widely used in recommendation systems to evaluate both the quality and relevance of the top-k recommendations. They are defined as follows:

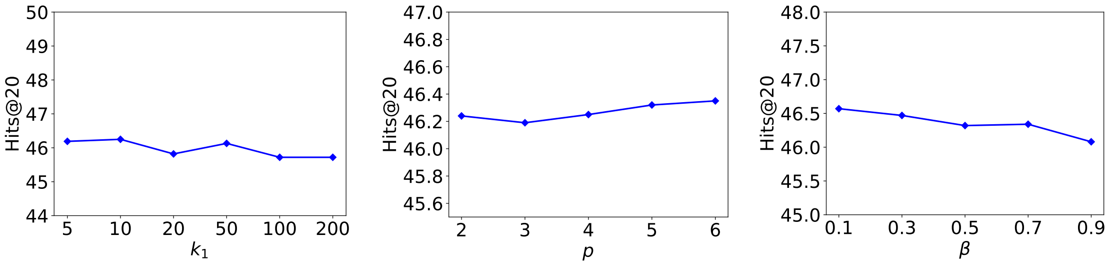
- **Hits@K**. It measures whether the true positive is within the top K predictions or not: Hits@K $= \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\text{rank}_i \leq \text{K})$. rank$_i$ is the rank of the i-th sample. The indicator function $\mathbf{1}$ is 1 if rank$_i \leq$ K, and 0 otherwise.

[2] https://www.aicrowd.com/challenges/amazon-kdd-cup-23-multilingual-recommendation-challenge

[3] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

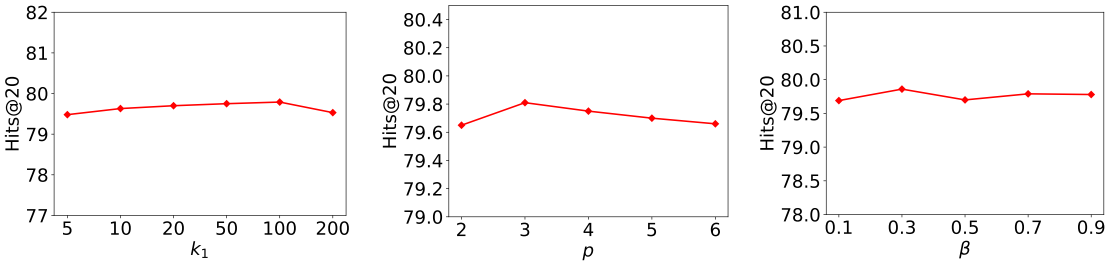
[4] https://huggingface.co/sentence-transformers/distiluse-base-multilingual-cased-v1

[5] https://recbole.io/index.html



(a) (b) (c)

Figure 10: Performance of AlterRec by varying k_1, p and β on the HD dataset.



(a) (b) (c)

Figure 11: Performance of AlterRec by varying k_1, p and β on the Amazon-French dataset.

- **NDCG@K**. It considers both the presence and the position of relevant items in the candidate list. NDCG@K $= \frac{1}{N}\sum_{i=1}^{N} \frac{\mathbf{1}(\text{rank}_i \leq \text{K})}{\log_2(\text{rank}_i * \mathbf{1}(\text{rank}_i \leq \text{K}) + 1)}$. The indicator function $\mathbf{1}$ is 1 if $\text{rank}_i \leq$ K, and 0 otherwise.

G Additional Parameter Analysis

In this section, we present the impact of three parameters: k_1, the parameter determining the starting index for selecting hard negative samples; p, the end index for augmented positive samples; and β, the weight assigned to the loss of augmented positive samples in Eq. (4) and Eq. (5). The results in terms of Hits@20 are presented in Figure 10 and Figure 11. Generally, the performance of AlterRec tends to decline as k_1 and β increase, suggesting that larger values may introduce additional noise. In addition, the performance remains relatively stable across different values of p, indicating that the model can achieve promising results with lightweight parameter tuning.