# ПРОГРЕСИВНІ ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

# PROGRESSIVE INFORMATION TECHNOLOGIES

UDC 004.852

## SIMPLE, FAST AND SCALABLE RECOMMENDATION SYSTEMS VIA EXTERNAL KNOWLEDGE DISTILLATION

**Androsov D. V.** – Post-graduate student, Institute for Applied System Analysis, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Kyiv, Ukraine.

**Nedashkovskaya N. I.** – Dr. Sc., Professor, Department of Mathematical Methods of System Analysis, Institute for Applied Systems Analysis, National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute", Associate Professor, Kyiv, Ukraine.

### ABSTRACT

**Context.** Recommendation systems are important tools for modern businesses to generate more income via proposing relevant goods to clients and achieve more loyal attendees. With deep learning emergence and hardware capabilities evolution it became possible to grasp customer behavioral patterns in data-driven way. However, accuracy of prediction is dependent on complexity of system, and these factors lead to increased delay in model's output. The object of the study is the task of issuing sequential recommendations, namely the next most relevant product, subject to restrictions on system response time.

**Objective.** The goal of the research is the synthesis of a deep neural network that can retrieve relevant items for a large portion of users with minimal delay.

**Method.** The proposed method of obtaining recommendation systems that leverages a mixture of Attention-based deep learning model architectures with application of knowledge graphs for prediction quality enhancement via explicit enrichment of recommendation candidate pool, demonstrates the benefits of decoder-only models and distillation learning framework. The latter approach was proven to demonstrate outstanding performance in solving recommendation retrieval task while responding fast for large user batch processing.

**Results.** A model of a recommender system and a method for its training are proposed, combining the knowledge distillation paradigm and learning on knowledge graphs. The proposed method was implemented via two-tower deep neural network to solve recommendation retrieval problem. A system for predicting the most relevant proposals for the user has been built, which includes the proposed model and its training method, as well as ranking indicators MAP@k and NDCG@k to assess the quality of the models. A program has been developed that implements the proposed architecture of the recommendation system, with the help of which the problem of issuing the most relevant proposals has been studied. When conducting experiments on a large amount of real data from user visits to an online retail store, it was found that the proposed method for designing recommender systems guarantees high relevance of the recommendations issued, is fast and unpretentious to computing resources at the stage of receiving responses from the system.

**Conclusions.** Series of conducted experiments confirmed that the proposed system effectively solves the problem in a short period of time, which is a strong argument in favor of its use in real conditions for large businesses that operate millions of visits per month and thousands of products. Prospects for further research within the given research topic include the use of other knowledge distillation methods, such as internal or self-distillation, the use of deep learning architectures other than the attention mechanism, and optimization of embedding vector storage.

**KEYWORDS:** knowledge distillation, knowledge graphs, decoder-only models, node embeddings, transformer models, attention mechanism, recurrent neural networks, long short-term memory networks, deep neural networks, personalized sequential recommendations, predicting the next most relevant product, user modeling.

### ABBREVIATIONS

CBF is a content-based filtering;
CF is a collaborative filtering;
CNN is a convolutional neural network;
DNN is a deep neural network;
KD is a knowledge distillation process;
KG is a knowledge graph;
KLD is a Kullback-Leibler divergence;
LSTM is a long short-term memory network;
MAP is a mean average precision;
NDCG is a normalized discounted cumulative gain;
MHA is a multi-head attention network;
NBO is a next-best offer problem;
NBA is a next-best action problem;
NDCG is a normalized discounted cumulative gain;
PMI is a pointwise mutual information;
RS is a recommender system.


DOI 10.15588/1607-3274-2025-3-12

## NOMENCLATURE

$s$ is a session data;
$I$ is an item data;
$t$ is a (discrete) time point;
$i_j^{(t)}$ is an item at time point $t$;
$s_t$ is a sample of time series;
$R(\cdot|\cdot)$ is a conditional probability mass function;
$G$ is a weighted graph, i.e. KG;
$V$ is a set of users and items – the vertices of $G$;
$\aleph(v)$ is a set of neighbors of node $v$ in a graph $G$;
$P$ is a projection operator over graph $G$;
$\Pr(\cdot)$ is a probability mass function in PMI definition;
$\Pr(\cdot,\cdot)$ is a joint probability mass function in PMI;
$\sigma$ is a sigmoid activation function;
$T$ is a softmax temperature;
$L$ is a RS loss function;
$h_t$ is a hidden state of neural network at time $t$, $h_t \in \mathrm{h}$;
$b_h$ is a bias vector for hidden state of a recurrent neural network;
$b_u$ is a bias vector for output state of a recurrent neural network;
$W_{\cdot h}$ is a weight matrix for hidden state of a recurrent neural network;
$W_{\cdot u}$ is a weight matrix for output state of a recurrent neural network;
$\alpha$ is a PMI threshold, $\alpha \in \mathbb{R}$;
$v_i$ is a logit for the $i$-th score produced by the student model;
$z_i$ is a logit by teacher model;
$q_i$ is a soft target output for $i$-th score produced by the student model;
$\hat{q}_i$ is a soft target output for $i$-th score produced by the teacher model;
$||$ is a vector concatenation operator;
$Q_A$ is an attention query weight matrix;
$K_A$ is an attention key weight matrix;
$V_A$ is an attention value weight matrix;
$d_K$ is a number of columns in matrix $K$.

## INTRODUCTION

Users' purchase decisions are significantly influenced not only by their general preferences but also by their most recent interactions with a given platform or marketplace. Understanding user behavior patterns is crucial for any customer-oriented business, as this obtained knowledge allow to propose the most relevant items to a given customer base, increasing revenue in both short- and long-term perspective. Such item proposal systems are called recommendation systems.

A recommendation system (RS) consists of a set of statistical models that analyze a user's interaction history, along with knowledge about the user and the items available, to generate relevant content recommendations [1]. Relevance, in this context, refers to the likelihood of a user engaging with the items presented. Consequently, there exists a broad spectrum of recommendation approaches, including non-personalized, semi-personalized, and personalized methods [1]. This work focuses specifically on the development of personalized recommendation systems, and thus, the terms "recommendation system" and "personalized recommendation system" are used interchangeably.

The content filtering (CBF) approach for recommendation systems construction is based on idea that the user is interested in items that are similar to items that were already interesting to this user earlier. Unlike collaborative filtering (CF) models, the similarity of items is determined not by a set of user actions, but based on the internal characteristics of the items themselves. To address the problem of items' feature descriptions extraction, deep learning methods are often used in the process of content filtering systems construction.

In recent years, RS have achieved substantial success across various real-world applications, including e-commerce platforms, streaming services, and online retail. A particularly notable application of recommendation systems is the next best offer (NBO) task, which involves predicting the items a user is likely to view or purchase after interacting with a platform.

NBO, also referred to as next best action (NBA) [2], or more broadly as next-basket recommendation (NBR) [3], is a prevalent use case for any enterprise engaged in business-to-consumer (B2C) operations. Marketing teams in these enterprises have been implementing NBO/NBA projects for many years, though many of these initiatives have failed to meet expectations [2]. Several factors contribute to this underperformance, including reliance on traditional methods, failure to update NBO models with new features (resulting in underutilization of both the breadth and depth of available data), inadequate campaign validation methods, technological shortcomings, and more.

The advent of machine learning and, consequently, deep neural networks (DNN) has introduced new opportunities for NBO/NBA by enabling the utilization of advanced technologies and large data sets to improve and optimize basket recommendations more effectively than ever before.

For instance, by leveraging deep learning techniques, the delivery of personalized offers and recommendations has been significantly enhanced, leading to notable improvements in customer engagement. These advancements can increase customer satisfaction and loyalty, ultimately driving higher sales and revenue for businesses [3, 4].


DOI 10.15588/1607-3274-2025-3-12

**The object of study** is the next-best offer (NBO) recommendation problem. NBO is a difficult task, since most session-based models' prediction pool is too narrow to accomplish the goal of grasping long-term inter-item dependencies and user behavior patterns. On the other hand, leveraging ubiquitously used collaborative filtering (CF) models do not capture short-term dependencies between items, what may be unsuitable for marketing campaigns design. Therefore, it is proposed to construct a new model, based on multi-head attention (MHA) mechanism, knowledge graphs (KG) and knowledge distillation (KD) techniques.

**The subject of study** is methods for sequential recommendation retrieval.

**The purpose of the work** is to create fast and scalable RS to solve NBO/NBA task for a large number of users.

## 1 PROBLEM STATEMENT

For a given multiset $s = \{i_j^{(t)} \mid j \in \mathbb{N}, j \le |I|, t \in \mathbb{N}\}$ of items in some set of available items (goods) $I$ and $t$ is a (discrete) time, called session, it is desired to model a likelihood function $R$ such that:

$$\hat{i}^{(t)} = \arg\max_i P(i \mid s). \quad (1)$$

Suppose the items and users are described by many categorical and numerical (continuous) features. Each categorical feature is presented by an embedding vector, thus generalizing the concept of latent variables in matrix factorization.

The main difficulty of such task is that it should be approached both by CBF and CF methods, since solving (1) solely with respect to the given user's session may diminish the explorative capabilities of RS, while applying only collaborative filtering, considering the demand of such models to be trained on large historical interactions datasets, may result in a system which cannot adapt to drift in the user behavior and is feasible to use for recognition of general preferences of users. The second challenge arises on the inference step – it is preferrable to update the recommendations on-line, adapting them to the newest user actions, thus limiting the complexity of the obtained RS. Current research addresses these challenges by developing a hybrid method of building RS.

## 2 REVIEW OF THE LITERATURE

Considering the variability in user session lengths, it becomes essential to capture both short- and long-term dependencies that exist between items within a session and the potential future items that a user might interact with. This challenge has led to the emergence of models based on high-order Markov chains, which offer a sophisticated approach to understanding and predicting user behavior. Among these models, context tree models (CT) [5, 6] and Markov chain similarity models [7] have proven particularly effective.

Context tree models function by first constructing a partition tree that represents each user session. This partition tree is then traversed to define a high-order Markov chain, allowing the model to encapsulate the user session [6]. The hierarchical structure of the partition tree provides a powerful framework for modeling the sequential nature of user interactions, enabling the recommendation system to account for complex patterns and long-term dependencies that simpler models might overlook.

In addition to context tree models, another promising approach involves integrating high-order Markov chains with similarity-based methods, such as sparse linear methods (SLIM) and factored item similarity models (FISM). This hybrid approach leverages the strengths of both Markov chains and similarity measures to capture a comprehensive range of relationships within the data. By combining these methodologies, the model is capable of simultaneously addressing short-term and long-term dependencies between users and items, as well as item-to-item relationships, thereby offering a more nuanced and accurate prediction of user preferences [7]. The integration of similarity-based techniques with high-order Markov chains enhances the model's ability to generalize across different users and sessions, ultimately leading to more personalized and effective recommendations.

Aside from Markov chain-based models, deep learning techniques have increasingly gained traction in addressing the challenges imposed by sequential recommendation tasks. Among the various deep neural networks (DNN) architectures, recurrent neural networks (RNNs) have emerged as a leading choice due to their capacity to model sequences of data, capturing both the short-term and long-term dependencies that characterize user interactions over time.

Consider the sample of time series $s_t \subset s$ and the remaining time series $s_{T-t}$. RNN in this case is a mapping function $f : s_t \to s_{T-s}$, and that function is a chain of non-linear transformations over affine transformations that are provided by state-space modeling of $s_{T-t}$ [8]. Vanilla RNN models these chains in a following way:

$$h_t = \sigma(W_{xh} x_t + W_{hh} h_t + b_h),$$
$$s_{T-t} = u_t = \sigma(W_{hu} h_t + b_u).$$

RNN is aimed to maximize logarithmic likelihood $\log P(s_{T-t} \mid s_t, U, W, V, c)$ [8]. However, despite its ability to model non-stationary time series, RNN has a couple of significant drawbacks – disability to parallelize hidden states computations, and consequently gradient vanishing, which is directly caused by its architecture [8].

Long short-term memory networks (LSTMs), firstly introduced in 1997 [9] have become the most widely adopted variant of RNNs for sequential recommendation tasks. LSTMs are designed to overcome the limitations of traditional RNNs, particularly the issues of vanishing and exploding gradients, by introducing memory cells that can


DOI 10.15588/1607-3274-2025-3-12

maintain and update information over long sequences. This makes LSTMs particularly effective at modeling the temporal dependencies within user sessions, allowing them to predict future interactions with a high degree of accuracy. Moreover, modifications of LSTMs, such as bi-directional LSTMs, further enhance the model's capability by enabling it to consider both past and future context when making predictions [10, 11]. Bi-directional LSTMs, contrary to "classic" ones, estimate parameters by traversing input sequences in both in forward and backward directions, i.e, at each time step, the outputs of the forward and backward LSTMs are concatenated (or combined) to form the final output. This allows the network to have access to both past and future context when making predictions. However, due to their architecture, these models are prone to violate causality requirements on sequential data, and the requirement to have all sequence available to perform backward pass make them unsuitable for online recommendation engines.

As an alternative to recurrent neural networks, relatively new family of deep learning approaches, called attention networks, have recently become ubiquitous choice for analyzing sequential data. Attention networks are based on attention mechanism, introduced in 2017 [12]. It allows the model to focus on specific parts of the input sequence when producing an output, enabling it to handle long-range dependencies more effectively than LSTMs or RNNs.

In traditional sequence-to-sequence models, such as those used in machine translation, the encoder processes the input sequence into a fixed-length context vector, which is then used by the decoder to generate the output sequence. However, this fixed-length context vector can be a bottleneck, especially for long sequences, as it forces the model to compress all information into a single vector.

The attention mechanism addresses this issue by allowing the decoder to access different parts of the encoder's output sequence directly, enabling it to focus on the most relevant parts of the input when generating each element of the output sequence [12].

Several variants of the attention mechanism exist, depending on the application and architecture. Self-attention used in transformer models, where the attention mechanism is applied to the same sequence, allowing each element to attend to all other elements in the sequence. The formula for scaled dot-product attention is:

$$\text{Attention}(Q_A, K_A, V_A) = \varsigma\left(\frac{Q_A{}^{\mathrm{T}} K_A}{\sqrt{d_K}}\right) V_A,$$

where $d_K$ is a number of columns in key matrix, $\varsigma(\cdot)$ is a softmax function.

The other popular modification of attention mechanism is multi-head attention (MHA). It extends self-attention by applying multiple attention mechanisms (heads) in parallel, each with different learned parameters, and then concatenating their outputs. This allows the model to focus on different parts of the input sequence.

Attention-based networks have become increasingly prevalent in recommendation retrieval tasks due to their ability to effectively model complex relationships in data. These networks, such as hierarchical attention networks, are designed to process and analyze inputs that capture both user-item and item-item interactions. By considering these interactions simultaneously, hierarchical attention networks can more accurately predict subsequent user actions, leading to more personalized and relevant recommendations [13].

Moreover, stochastic self-attention networks represent an another advancement in this domain. These networks leverage the self-attention mechanism to dynamically assess the importance of different elements within the input sequence, thereby generating candidate recommendations with enhanced precision. The stochastic nature of these models introduces an element of randomness, which can help in exploring a broader range of potential recommendations, thereby improving the diversity and relevance of the suggested items [14].

In summary, attention networks play a critical role in the evolution of recommendation systems. Their ability to incorporate complex interactions and adapt to various input dynamics makes them indispensable tools for enhancing the accuracy and diversity of recommendations in modern retrieval tasks.

## 3 MATERIALS AND METHODS

Since the main drawback of sequential recommendation is in narrow candidate pool, it is crucial to enrich recommendation proposals beyond trending items and short-term user-item and item-item relationships. To overcome this challenge, it is proposed to change the structure of received data and augment the given RS with some external context.

The relationships between users and items, as well as between users themselves and items themselves, can be naturally represented by a graph $G = G(V, E, w, f)$, where $V = \langle U, I \rangle$ denotes the set of users and items – the vertices of the graph – and $E = \{(u,i) \mid u \in V, i \in V\}$ represents the set of edges connecting users with items, items with items and users with users. The edge weights $w \in \mathbb{R}$, are assigned through a mapping $f$. Since this graph represents stable relationships between items and users, it is proposed to name this structure a knowledge graph (KG).

Given this graph-based representation, geometric deep learning frameworks, such as graph neural networks (GNNs), are well-suited for addressing recommendation retrieval tasks.

GNNs are a class of deep learning models specifically designed to operate on data that is represented as graphs [15, 16]. These networks excel in tasks that require inference over graph-structured data by iteratively updating the representations of vertices through the aggregation of information from their neighboring vertices.


DOI 10.15588/1607-3274-2025-3-12

During the training process, each vertex $v \in V$ within the graph $G = G(V, E)$ refines its feature representation by incorporating features from its adjacent vertices. This iterative process of feature aggregation and representation updating can be expressed as:

$$h_v^{(k+1)} = \sigma\left( \sum_{u \in \aleph(v)} \left( W^{(k)} h_u^{(k)} + b^{(k)} \right) \right), \quad (2)$$

where $h_v^{(k)}$ denotes the feature vector of vertex $v$ at the $k$-th layer, $\aleph(v)$ represents the set of neighboring vertices of $v$. The parameters $W^{(k)}$ and $b^{(k)}$ are the weights and biases specific to $k$-th layer, respectively. The function $\sigma$ is a non-linear activation function, for example, sigmoid.

The iterative aggregation (2) allows GNNs to effectively capture and propagate local structural information throughout the network, leading to a more comprehensive understanding of the graph's global structure. As a result, GNNs are well-suited for a variety of tasks, including node classification, link prediction, and graph classification, where the relationships and dependencies between entities are naturally modeled as graphs. The ability of GNNs to leverage the inherent graph structure makes them particularly powerful for applications in domains such as social network analysis, molecular chemistry, recommendation systems, and more.

To process categorical data, embeddings are used that form a dense representation of each category.

For instance, [17] demonstrates an approach that integrates the attention mechanism with graph convolutional networks [15] to effectively learn embeddings from the user-item graph. This combined model is then leveraged to generate recommendations for the next item a user is likely to interact with.

In the current work it is proposed to modify MHA by referencing not only input sequence (i.e. self-attention heads), but to aggregate first-order neighbors of each input of a sequence with respect to the given knowledge graph.

More specifically, consider the heterogenous graph:

$$G = G(V, E, w, f, P), \quad (3)$$

where $P$ is an edge properties set. Let's define such projection operator over (3) as follows:

$$P: G, p \to G_p, \quad G_p = G(V, E, w, f, P = p). \quad (4)$$

It is obvious that graph $G_p$ is weighted graph where only those edges preserved that share same property, e.g. user-movie graph that contain only US-based users.

The proposed attention modification, named as Mixed Attention, applies self-attention over the given session multiset $s$ and operator (4) over the KG (3), performing the following computation:

$$\text{MixedAttention}(Q_A, K_A, V_A) = \underset{p \in P \cup s}{\|} \text{Attention}(Q_p, K_p, V_p). \quad (5)$$

This application of graph-based learning methods is called to enhance the potential of Attention Networks in capturing the intricate relational structures inherent in recommendation systems, improving the accuracy of the recommendations produced.

However, this computation implies traversing two structures – knowledge graph $G = G(V, E, w, f, P)$ (3), (4) and session $s$ simultaneously to find each item nearest neighbors, hence the same computations are required on each inference step, which may increase complexity and latency of proposed RS. This problem could be mitigated by knowledge distillation (KD) techniques.

Knowledge distillation is a model compression technique that is designed to transfer the knowledge encapsulated within a large, highly complex model, known as the teacher, to a smaller and more computationally efficient model, referred to as the student [18, 19]. The principal idea underlying knowledge distillation is to enable the student model to mimic the behavior of the teacher model. This is achieved by training the student model to replicate the output distributions produced by the teacher model, in addition to the conventional training on labeled data. To facilitate learning, the concepts of learning on logits is introduce.

There exist multiple ways to perform KD, but the chosen one in the current proposal is performed via teacher's output distribution temperature scaling.

By transferring knowledge from teacher to student in a classification problem, we minimize the loss function of the class distribution predicted by the teacher model. Let us consider the case of accurate model, when the prediction of the probability of one of the classes (the correct one) is close to unity, and all others are close to zero. Such data is usually of little help for the student model, since it practically does not differ from the original labels. Therefore, a softmax temperature (normally set to 1) is introduced [18], which helps the student model to repeat not the classification labeled data, but the probability distribution, and allows the student model to better adopt the teacher's behavior. Let $v_i$ denote the logits or pre-softmax outputs for the $i$-th score produced by the student model. The corresponding student soft target output $q_i$ for $i$-th score is computed as follows:

$$q_i = \frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)}.$$


DOI 10.15588/1607-3274-2025-3-12

Higher values of a temperature parameter $T$ produce softer probability distributions, which contain more information about the relative confidence levels across classes.

Suppose the teacher model has logits $z_i$, which produce soft target probabilities $\hat{q}_i$, and distillation is performed at temperature $T$. Then, the gradient of the cross-entropy function $L_{CE}$ with respect to each logit $v_i$ of the student model is given by [18]:

$$\frac{\partial L_{CE}}{\partial v_i} = \frac{1}{T}(q_i - \hat{q}_i) = \frac{1}{T}\left(\frac{\exp\left(\frac{z_i}{T}\right)}{\sum_j \exp\left(\frac{z_j}{T}\right)} - \frac{\exp\left(\frac{v_i}{T}\right)}{\sum_j \exp\left(\frac{v_j}{T}\right)}\right).$$

If $T$ is high in comparison with the logits $v_i$, the gradient of loss $L_{CE}$ can be approximated as follows:

$$\frac{\partial L_{CE}}{\partial v_i} \approx \frac{1}{T}\left(\frac{1+\frac{z_i}{T}}{N+\frac{\sum_j z_j}{T}} - \frac{1+\frac{v_i}{T}}{N+\frac{\sum_j v_j}{T}}\right). \quad (6)$$

Let the logits $v_i$ and $z_i$ have a zero mean separately for each transfer case. Then, equation (6) is simplified to the following:

$$\frac{\partial L_{CE}}{\partial v_i} \approx \frac{1}{NT^2}(z_i - v_i),$$

and distillation is equivalent to minimizing $\frac{1}{2}(z_i - v_i)^2$ under the above conditions.

If $T$ is relatively low, distillation practically ignores large negative logits (which are much more negative than the average). On the one hand, this is an advantage, since such logits could be very noisy. It has been shown in [18] that intermediate temperatures $T$ work best when the student model is much too small in comparison with the teacher model.

Thus, in the paper, in process of training the student model, it is proposed to minimize Kullback-Leibler divergence (KLD) measure between student and teacher models, defined as:

$$KL(q \mid \hat{q}) = \int_{\infty}^{+\infty} q(x) \log \frac{q(x)}{\hat{q}(x)} dx. \quad (7)$$

By learning from the teacher's soft targets, the student model can generalize better on unseen data, often leading to improved performance compared to directly training the smaller model from scratch.

The proposed method consists of the following stages:

1. To construct the teacher model, which consists of the following elements:

– knowledge graph (3), (4);

– mixed attention, that consumes both user session and traverses knowledge graph to compute the operation (5);

– multi-layer perceptron with several hidden layers, which takes results of the mixed attention operations an input and has the softmax output layer to produce some probability distribution.

2. To construct the student model, which consists of the following elements:

– multi-head attention that takes as an input the user session sequence;

– convolutional neural network with several filters, consuming the attention scores, obtained at the previous stage along outputting softmax-mapped vector of the same size as the teacher model output.

3. To perform student model learning: the KL divergence (7) between model outputs is minimized.

## 4 EXPERIMENTS

It is proposed to solve the problem of NBO/NBA recommendation leveraging information retrieved from user interaction history and item properties.

More precisely, consider the dataset retrieved from an anonymous multi-brand and multi-category e-commerce store, which schema is provided in Table 1.

The dataset contains historical data from October 2019 to November 2019, overall storing approximately 6.5 million user sessions, or 69 million records.

Let us predict the final item for each user session leveraging the proposed method.

As a baseline model for solving the problem (1), an LSTM-based architecture is chosen (Fig. 1).

As a teacher model, the proposed extension of attention mechanism (5) is introduced instead of LSTM module, thus allowing the model to capture long-term relationships from a given KG (Fig. 2).

It could be observed, that the proposed architecture (Fig. 2) combines both content-based RS (via considering past interaction history) and collaborative filtering (via considering user-item model branching).

Table 1 – Dataset fields description

| Field Name | Data Type | Description |
|---|---|---|
| Session Id | Base64 | Unique identifier of user visit |
| Product Id | Integer | Stock keeping unit (SKU) of an item |
| Product Description | String | Description of item |
| Brand | String | Brand name |
| Category | String | Category of item, e.g. furniture |
| Price | Integer | Price in cents |
| Action | String | User action over item, e.g. add to cart, view |
| Timestamp | Timestamp | Date and time of interaction |


DOI 10.15588/1607-3274-2025-3-12

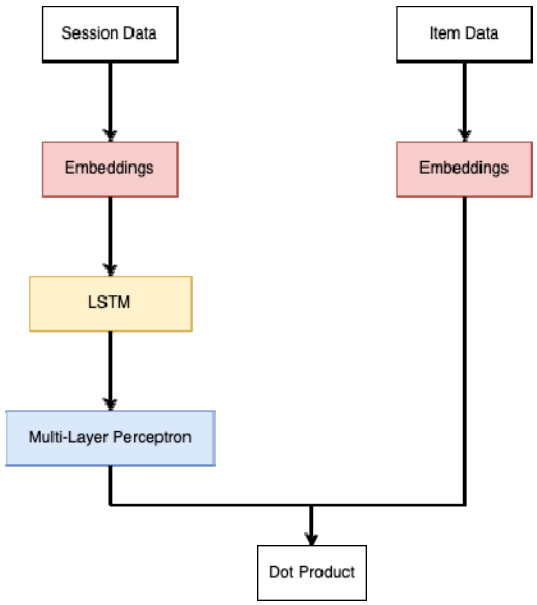


Figure 1 – LSTM-based Baseline model architecture

The general purpose of model is to generate embeddings for NBO/NBA candidate items. Thus, for solving task (1) it is proposed to minimize cross-entropy loss $L$ between obtained embedding $e_{\text{obtained}}$ and ground truth $e_{\text{true}}$:

$$L(e_{\text{obtained}}, e_{\text{true}}) = -e_{\text{obtained}}{}^{\text{T}} \log(e_{\text{true}}).$$

It is worth mentioning that the construction of KG is done in data-driven way by thresholding pointwise mutual information (PMI) between item pairs.

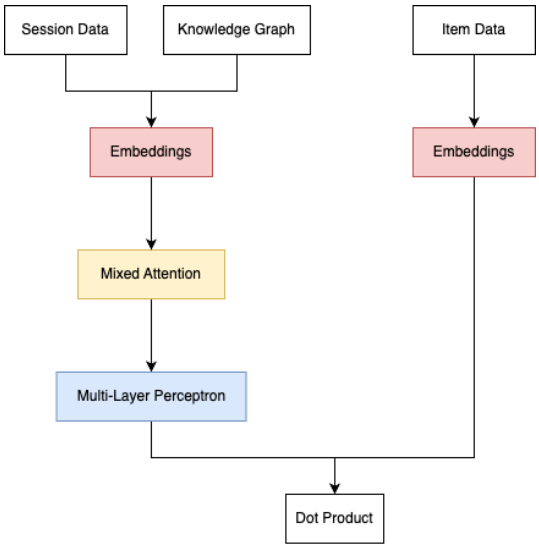


Figure 2 – The proposed Mixed Attention-based model architecture

KG is obtained by choosing only pairs of items where PMI value is greater than α. The selected threshold is defined at the 25-th percentile of all PMI values.

PMI between two items $i$ and $j$ is defined as:

$$\text{PMI}(i, j) = \log\left(\frac{\Pr(i, j)}{\Pr(i)\Pr(j)}\right),$$

where $\Pr(i, j)$ is the joint probability of items $i$ and $j$ co-occurring, $\Pr(i)$ and $\Pr(j)$ are the individual probabilities of items $i$ and $j$ occurring independently [20].

Item embeddings could be chosen in two ways – random initialization or leveraging pre-trained embeddings. To achieve the latter from the obtained KG, it is proposed to apply Node2Vec algorithm [21].

The main purpose of Node2Vec is to capture both the local and global network structure of a given graph. It does this by performing biased random walks on the graph, by balancing between breadth-first search (BFS) and depth-first search (DFS). This allows Node2Vec to generate node embeddings that reflect both the community structure (via BFS) and functional similarity (via DFS) within the graph [21]. For the experiment purposes, the only adjusted hyper-parameter is embedding dimension, which should align with the corresponding hyper-parameter in all proposed architectures.

The next step is to define the student model, parameters of which will be optimized via temperature scaling. This model is utilizing MHA module, thus eliminating the need for constantly traversing KG for each recommendation suggestion. The schematic representation of proposed model is shown in Fig. 3.

The student model learns the probability distribution of the teacher; thus, it is proposed to minimize KL divergence (7) between student model and teacher model.

Concluding, the following hyper-parameters are set for baseline model:

1. Embedding dimension – 64.
2. Multi-Layer Perceptron layer number – 2.
3. LSTM cell size – 64.

Consequently, the following hyper-parameters are set for teacher model:

1. Embedding dimension – 64.
2. Multi-Layer Perceptron layer number – 2.
3. Query, Key and Value matrix size – 64.
4. Causal mask – applied to guarantee that current decisions don't affect previous ones.

On the other side, since student model utilizes convolutional neural networks (CNN) instead of MLP, the following hyper-parameters are picked:

1. Stride size – 1.
2. Padding – "same".
3. Dropout rate – 50%.
4. Number of heads in MHA – 2.
5. Embedding dimension – 64.
6. Query, Key and Value matrix size – 64.
7. Temperature $T$ – [0.5, 1, 2, 5].


DOI 10.15588/1607-3274-2025-3-12

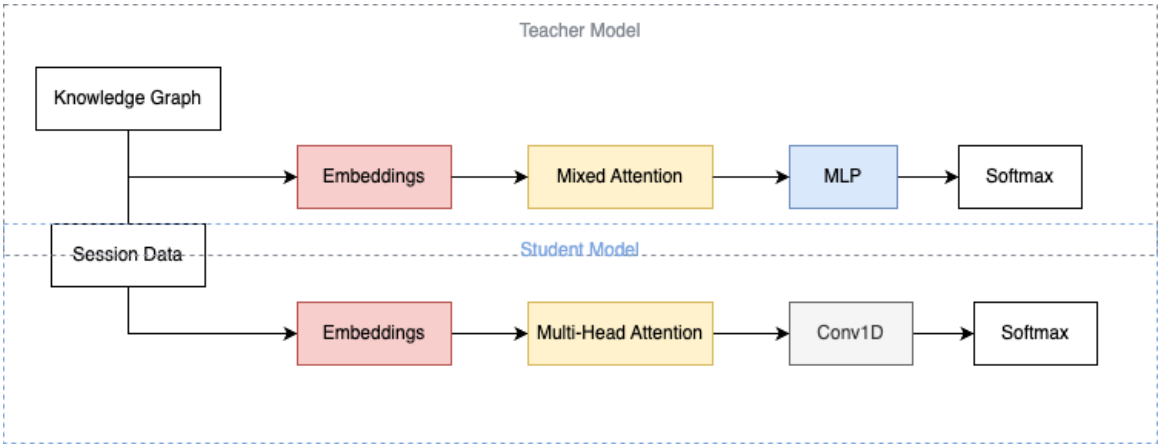


Figure 3 – The proposed teacher-student architecture

The choice of padding type and stride size is influenced by the constraint on output of NN to be an embedding of next item, given users' previous interactions data.

In order to examine the ability of models to retrieve relevant items, the following ranking metrics were chosen:

1. Mean average precision @ k (MAP @ k) – ranking metric used to evaluate the accuracy of a ranked list of items up to a cutoff rank k.

2. Normalized discounted cumulative gain @ k (NDCG @ k) – measure of ranking quality that considers the position of relevant items in the ranked list up to a cutoff rank k and applies penalty to relevant items that appear lower in the list by applying a logarithmic discount.

For both metrics, k values 1, 10 and 100 are considered.

Also, since the complexity of system affects recommendation candidate calculation time, mean retrieval time is proposed as the auxiliary metric to consider along with ranking ones.

## 5 RESULTS

In the Tables 2–3 MAP@k and NDCG@k metrics for given k ranking cut-off for LSTM baseline and Mixed Attention model statistics are provided. The best Mixed Attention model is selected as the teacher model.

Table 2 – Results of Baseline and Mixed Attention models benchmarking by MAP metric

| Model | MAP@1 | MAP@10 | MAP@100 |
|---|---|---|---|
| Baseline | 0.1492 | 0.2766 | 0.2859 |
| Baseline + Node2Vec | 0.1453 | 0.2740 | 0.2824 |
| Mixed Attention | 0.1769 | 0.3003 | 0.3082 |
| Mixed Attention + Node2Vec | 0.2 | 0.3316 | 0.3378 |

Table 3 – Results of Baseline and Mixed Attention models benchmarking by NDCG metric

| Model | NDCG@1 | NDCG@10 | NDCG@100 |
|---|---|---|---|
| Baseline | 0.1529 | 0.3296 | 0.3767 |
| Baseline + Node2Vec | 0.1509 | 0.3276 | 0.374 |
| Mixed Attention | 0.1755 | 0.3487 | 0.3865 |
| Mixed Attention + Node2Vec | 0.2025 | 0.3794 | 0.417 |

On the other hand, Table 4 and Table 5 reflect the values of MAP@k and NDCG@k results of KD for different temperature values, respectively. Table 6 summarizes models time performance. Figures 4–7 depict the evolution of ranking metrics with each epoch.

Table 4 – Results of KD benchmarking by MAP metric

| *T* | MAP@1 | MAP@10 | MAP@100 |
|---|---|---|---|
| 0.5 | 0.1977 | 0.2727 | 0.2773 |
| 1 | 0.1952 | 0.325 | 0.3315 |
| 2 | 0.0832 | 0.1441 | 0.1487 |
| 5 | 0.076 | 0.1352 | 0.1789 |

Table 5 – Results of KD benchmarking by NDCG metric

| *T* | NDCG@1 | NDCG@10 | NDCG@100 |
|---|---|---|---|
| 0.5 | 0.198 | 0.3041 | 0.33 |
| 1 | 0.1948 | 0.3745 | 0.374 |
| 2 | 0.0831 | 0.1708 | 0.1956 |
| 5 | 0.0368 | 0.1180 | 0.1417 |

Table 6 – Average time per 1000 requests per model

| Model | Average time per 1000 requests, s |
|---|---|
| Baseline | 1.01 |
| Mixed Attention + Node2Vec | 2.67 |
| Proposed student model | 0.189 |


DOI 10.15588/1607-3274-2025-3-12

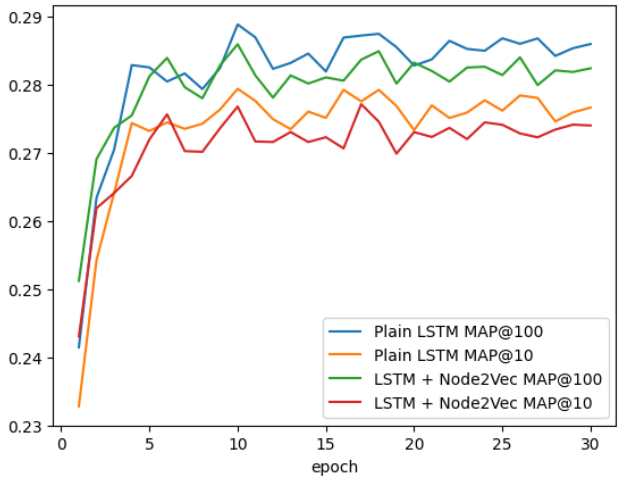


Figure 4 – MAP@k per epoch for baseline models

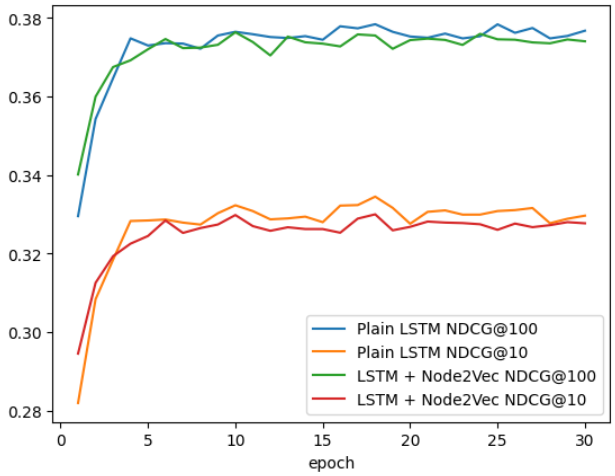


Figure 5 – NDCG@k per epoch for baseline models

## 6 DISCUSSION

As follows from Table 2 and Table 3, proposed Mixed Attention approach strongly outperforms LSTM-based baseline models. However, the fact that Node2Vec pre-trained embeddings cause little-to-no impact on ranking metrics for non-graph-based model but significantly enhances predictive capabilities of models that utilize KGs, is quite surprising and contradicts the initial suggestion that "implicit" knowledge, reflected solely in pre-trained embeddings could enhance sequential models.

It is worth noticing that this behavior persists for each epoch, as shown on Fig. 4 and 5 for Baseline models and Fig. 6 and 7 for Mixed Attention models, respectively.

Since the best model by all ranking metrics is Mixed Attention model with Node2Vec pre-trained embedding, this model is used as a teacher model.

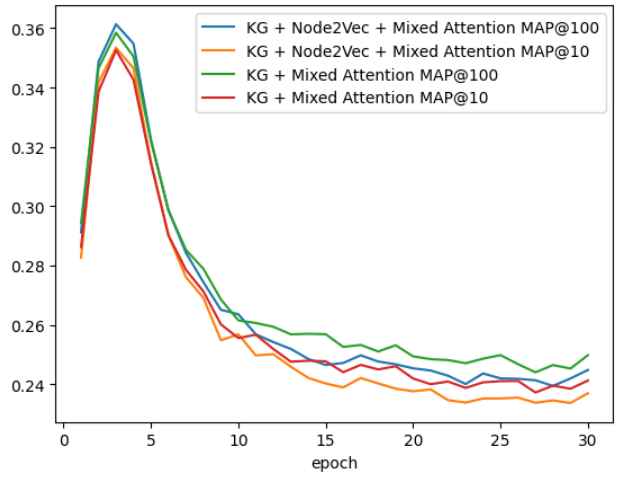


Figure 6 – MAP@k per epoch for Mixed Attention models

After performing temperature scaling with different values of parameter *T*, various student models were obtained. Since the bigger temperature, the more output distribution is uniform-like. Whilst very low value can introduce overconfidence to model decisions, it was predictable that both high and low *T* values could decrease predictive capabilities of model. The overall dependencies between temperature scaled outputs of teacher model and student model performance are listed in Tables 4 and 5.

The best model was obtained without performing temperature scaling of teacher model outputs. It is also worth noticing that results are only slightly worse than teacher's model ones, namely Mixed Attention + Node2Vec models in Tables 2 and 3.

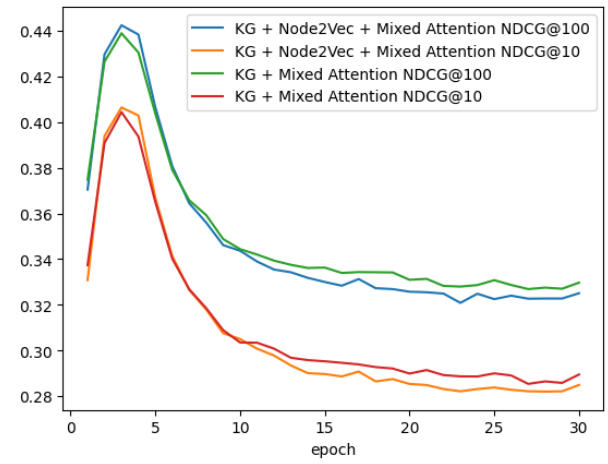


Figure 7 – NDCG@k per epoch for Mixed Attention models

It is also worth noticing the graph representation quality of the recommender system, obtained with the proposed KG and distillation method (Fig. 3). On Fig. 8 one can see the top-5 recommendations given that user has put Nike shoes to the basket or purchased this item. As one can see, the proposed model captures associations from the KG with a decent accuracy, grasping relationships between sport shoes and fitness vehicles and equipment, although the model itself gives irrelevant recommendation to buy desktop.


DOI 10.15588/1607-3274-2025-3-12

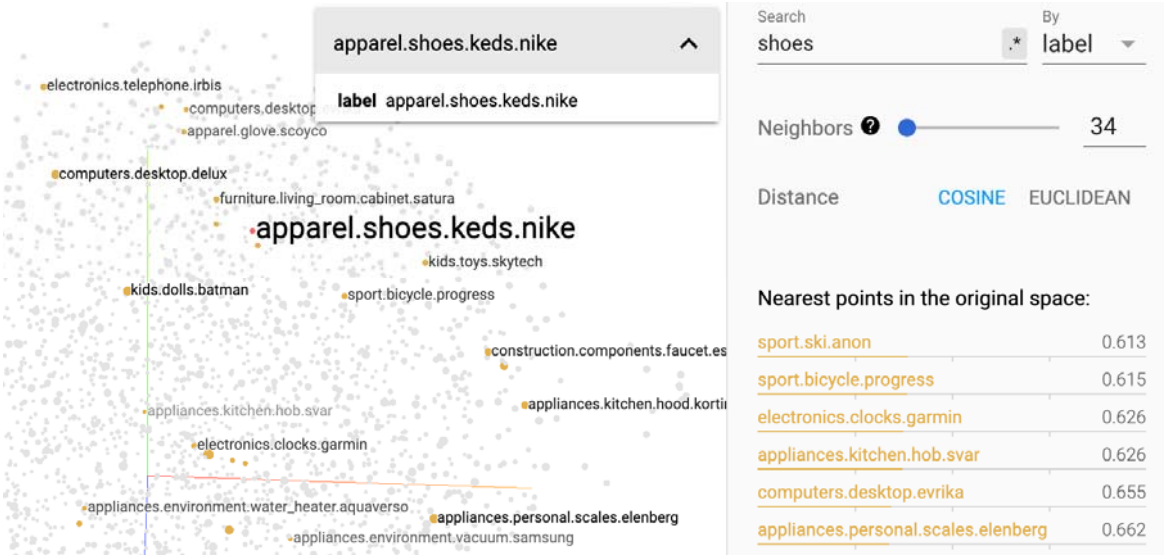


Figure 8 – Top-5 recommendations, obtained using the proposed model

The final assessment is conducted between best models in each category – LSTM, Mixed Attention and distilled student model. The main objective is to measure average time per 1000 requests for retrieving 100 most relevant items, given user sessions data. The received measures, recorded into Table 6, show that the proposed architecture (Fig. 3) and method significantly outperforms baseline solution by offering the main benefit of transformer-like architecture over RNNs – high degree of computations parallelization.

## CONCLUSIONS

The problem of next best offer prediction is solved in this work using multiple deep learning-based approaches.

**The scientific novelty** of obtained results shows that by combining learning on graphs and knowledge distillation it is feasible to build scalable, fast and precise recommendations systems.

**The practical significance** of current work and its results is that implemented models could be applied to forecast users next interactions on the enterprise scale.

**Prospects for further research** are to examine other architectural approaches, different from decoder-only models, and propose alternatives to Attention networks.

## ACKNOWLEDGEMENTS

This study was funded and supported by National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute" (NTUU KPI) in Kyiv (Ukraine), and also financed in part of the NTUU KPI Science-Research Work by the National Academy of Sciences of Ukraine "Development of models and methods for solving predictive problems based on large amounts of poorly structured information in conditions of uncertainty" (State Reg. No. 0122U000671).

## REFERENCES

1. Falk K. Practical Recommender Systems. Shelter Island, Manning, 2019, 432 p.
2. Rasool A. Next Best Offer (NBO) / Next Best Action (NBA) – why it requires a fresh perspective? [Electronic resource]. Access mode: https://www.linkedin.com/pulse/next-best-offer-nbo-action-nba-why-requires-fresh-azaz-rasool/
3. Wang S., Wang Y., Hu L. et al. Modeling User Demand Evolution for Next-Basket Prediction, *IEEE Transactions on Knowledge and Data Engineering*, 2023, Vol. 35, Issue 11, pp. 11585–11598. DOI: 10.1109/TKDE.2022.3231018.
4. Eliyahu K. A. Achieving Commercial Excellence through Next Best Offer models. [Electronic resource]. Access mode: https://www.linkedin.com/pulse/achieving-commercial-excellence-through-next-best-offer-kisliuk/
5. Wang S., Hu L., Wang Y. et al. Sequential Recommender Systems: Challenges, Progress and Prospects, *International Joint Conference on Artificial Intelligence : Twenty-eighth international joint conference, IJCAI 2019, Macao, 10–16 August 2019 : proceedings*. Macao: International Joint Conference on Artificial Intelligence, 2019, pp. 6332–6338. DOI: 10.24963/ijcai.2019/883.
6. Garcin F., Dimitrakakis C., Faltings B. Personalized News Recommendation with Context Trees, *Recommender systems : Seventh ACM conference, RecSys'13, Hong-Kong, 12–16 October 2013 : proceedings*. New York, Association for Computing Machinery, 2013, pp. 105–112. DOI: 10.1145/2507157.2507166.
7. He R., McAuley J. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation, *ArXiv*, 2016. DOI: 1609.09152v1.
8. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow. Sebastopol, O'Reilly Media Inc., 2017, 760 p.
9. Hochreiter S., Schmidhuber J. Long short-term memory, *Neural computation*, 1997, Vol. 9, № 8, pp. 1735–1780.
10. Xia Q., Jiang P., Sun F. et al. Modeling Consumer Buying Decision for Recommendation Based on Multi-Task Deep Learning, *Information and Knowledge Management : Twenty-seventh ACM international conference, CIKM '18, Torino, 22–26 October 2018 : proceedings.* New York, Association for Computing Machinery, 2018, pp. 1703–1706. DOI: 10.1145/3269206.3269285.
11. Zhao C., You J., Wen X., Li X. Deep Bi-LSTM Networks for Sequential Recommendation, *Entropy (Basel)*, 2020, Vol. 22, Issue 8, P. 870. DOI: 10.3390/e22080870.


DOI 10.15588/1607-3274-2025-3-12

12. Vaswani A., Shazeer N., Parmar N. et al. Attention is all you need, *Neural Information Processing Systems : Thirty-first international conference, NIPS '17, Long Beach, California, 04–09 December 2017 : proceedings*. New York: Curran Associates Inc., 2017, pp. 6000–6010.
13. Ying H., Zhuang F., Zhang F. et al. Sequential Recommender System based on Hierarchical Attention Network, *International Joint Conference on Artificial Intelligence : Twenty-seventh international joint conference, IJCAI '18, Stockholm, 13–19 July 2018 : proceedings*. Menlo Park, AAAI Press, 2018, pp. 3926–3932. DOI: 10.24963/ijcai.2018/546.
14. Fan Z., Liu Z., Wang Y. et al. Sequential Recommendation via Stochastic Self-Attention, *ACM Web Conference 2022, WWW '22, Lyon, 25–29 April 2022 : proceedings*. New York, Association for Computing Machinery, 2022, pp. 2036–2047. DOI: 10.1145/3485447.3512077.
15. Kipf T. N., Welling M. Semi-Supervised Classification with Graph Convolutional Networks, *International Conference on Learning Representations : Fifth international conference, ICLR 2017, Toulon, 24–26 April 2017 : proceedings*. New York, Curran Associates Inc., 2017. DOI: 10.48550/arXiv.1609.02907.
16. Wu Z., Pan S., Chen F. et al. A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs, *IEEE Transactions on Neural Networks and Learning Systems*, 2022, Vol. 32, № 1, pp. 4–24. DOI: 10.1109/TNNLS.2020.2978386.
17. Hekmatfar T., Haratizadeh S., Razban P., Goliaei S.]Attention-Based Recommendation On Graphs, *ArXiv*, 2022. DOI: 2201.05499.
18. Hinton G., Vinyals O., Dean J. Distilling the knowledge in a neural network, *ArXiv*, 2015. DOI: 1503.02531.
19. Ba L. J., Caruana R. Do Deep Nets Really Need to be Deep? *Advances in Neural Information Processing Systems*, 2014, Vol. 27, pp. 2654–2662. DOI: 1312.6184.
20. Church K. W., Hanks P. Word association norms, mutual information, and lexicography, *Computational Linguistics*, 1990, Vol. 16, № 1, pp. 22–29.
21. Grover A., Leskovec J. Node2vec: Scalable Feature Learning for Networks, *ArXiv*, 2016. DOI: 1607.00653.

Received 11.05.2025.
Accepted 04.07.2025.

УДК 004.852

## ПРОСТІ, ШВИДКІ ТА МАСШТАБОВАНІ РЕКОМЕНДАЦІЙНІ СИСТЕМИ ЗАСНОВАНІ НА ФІЛЬТРАЦІЇ ЗНАНЬ ВІД ВЧИТЕЛЯ

**Андросов Д. В**. – аспірант кафедри штучного інтелекту Навчально-наукового Інституту прикладного системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», Київ, Україна.

**Недашківська Н. І.** – д-р техн. наук, професор кафедри математичних методів системного аналізу Навчально-наукового Інституту прикладного системного аналізу Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського», доцент, Київ, Україна.

### АНОТАЦІЯ

**Актуальність.** Системи рекомендацій – важливі інструменти для сучасного бізнесу, які дають змогу отримувати більший дохід за рахунок пропозиції клієнтам відповідних товарів та залучення більш лояльних відвідувачів. З появою глибокого навчання та розвитком апаратних можливостей стало можливим уловлювати моделі поведінки клієнтів на основі даних. Однак точність прогнозу залежить від складності системи, і ці фактори призводять до збільшення затримки виведення на основі моделі. Об'єктом дослідження є задача видачі послідовних рекомендацій, а саме – наступного найбільш релевантного товару в умовах наявності обмежень по часу відповіді системи.

**Ціль.** Метою дослідження є синтез глибокої нейронної мережі, яка з мінімальною затримкою може отримувати релевантні елементи для більшості користувачів.

**Метод.** Пропонований метод отримання систем рекомендацій, який використовує поєднання архітектур моделей глибокого навчання на основі уваги із застосуванням графів знань для підвищення якості прогнозування за допомогою явного збагачення пулу кандидатів для рекомендацій, демонструє переваги моделей декодування та структури дистиляційного навчання. Було доведено, що підхід дистиляції знань є надзвичайно продуктивним під час вирішення завдань пошуку рекомендацій, одночасно швидко реагуючи на пакетну обробку великих обсягів даних користувачів.

**Результати.** Запропоновано модель рекомендаційної системи та метод її навчання, що поєднує парадигму дистиляції знань та навчання на графах знань. Запропонований метод був реалізований через двобаштову глибоку нейронну мережу для вирішення проблеми пошуку рекомендацій. Побудовано систему прогнозування найбільш релевантних наступних пропозицій для користувача, яка включає пропоновану модель та метод її навчання, а також показники ранжування MAP@k та NDCG@k для оцінки якості роботи моделей. Розроблено програму, яка реалізує пропоновану архітектуру рекомендаційної системи, за допомогою якої досліджена проблема видачі найрелевантніших наступних пропозицій. Під час проведення експериментів на великій кількості реальних даних візитів користувачів до онлайн магазину роздрібної торгівлі було встановлено, що пропонований метод конструкції рекомендаційних систем гарантує високу релевантність виданих рекомендацій, є швидким та невибагливим до обчислювальних ресурсів на етапі отримання відповідей від системи.

**Висновки.** Проведені експерименти підтвердили, що запропонована система ефективно вирішує поставлену задачу за малий проміжок часу, що є вагомим аргументом на користь її застосування в реальних умовах для великих бізнесів, що оперують мільйонами візитів на місяць та тисячами товарів. Перспективи подальших досліджень в рамках заданої теми дослідження включають в себе використання інших методів дистиляції знань, таких як внутрішня або само-дистиляція, використання відмінних від механізму уваги архітектур глибинного навчання, а також оптимізація сховища векторів вкладень.

**КЛЮЧОВІ СЛОВА:** дистиляція знань, графи знань, декодувальні моделі, вкладення вершин графів, архітектури типу «трансформер», механізм уваги, рекурентні нейронні мережі, мережі довгострокової короткої пам'яті, глибинні нейронні мережі, персоналізовані послідовні рекомендації, прогнозування наступного найбільш релевантного товару, моделювання користувача.


DOI 10.15588/1607-3274-2025-3-12

**ЛІТЕРАТУРА**

1. Falk K. Practical Recommender Systems / K. Falk. – Shelter Island: Manning, 2019. – 432 p.
2. Rasool A. Next Best Offer (NBO) / Next Best Action (NBA) – why it requires a fresh perspective? [Electronic resource]. – Access mode: https://www.linkedin.com/pulse/next-best-offer-nbo-action-nba-why-requires-fresh-azaz-rasool/
3. Modeling User Demand Evolution for Next-Basket Prediction / [S. Wang, Y. Wang, L. Hu et al.] // IEEE Transactions on Knowledge and Data Engineering – 2023. – Vol. 35, Issue 11. – P. 11585–11598. DOI: 10.1109/TKDE.2022.3231018.
4. Eliyahu K. A. Achieving Commercial Excellence through Next Best Offer models. [Electronic resource]. – Access mode: https://www.linkedin.com/pulse/achieving-commercial-excellence-through-next-best-offer-kisliuk/
5. Sequential Recommender Systems: Challenges, Progress and Prospects / [S. Wang, L. Hu, Y. Wang et al.] // International Joint Conference on Artificial Intelligence : Twenty-eighth international joint conference, IJCAI 2019, Macao, 10–16 August 2019 : proceedings. – Macao: International Joint Conference on Artificial Intelligence, 2019. – P. 6332–6338. DOI: 10.24963/ijcai.2019/883.
6. Garcin F. Personalized News Recommendation with Context Trees / F. Garcin, C. Dimitrakakis, B. Faltings // Recommender systems : Seventh ACM conference, RecSys'13, Hong-Kong, 12–16 October 2013 : proceedings. – New York: Association for Computing Machinery, 2013. – P. 105–112. DOI: 10.1145/2507157.2507166.
7. He R. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation / R. He, J. McAuley // ArXiv. – 2016. DOI: 1609.09152v1.
8. Geron A. Hands-On Machine Learning with Scikit-Learn and TensorFlow / A. Geron. – Sebastopol: O'Reilly Media Inc., 2017. – 760 p.
9. Hochreiter S. Long short-term memory / S. Hochreiter, J. Schmidhuber // Neural computation. – 1997. – Vol. 9, № 8. – P. 1735–1780.
10. Modeling Consumer Buying Decision for Recommendation Based on Multi-Task Deep Learning / [Q. Xia, P. Jiang, F. Sun et al.] // Information and Knowledge Management : Twenty-seventh ACM international conference, CIKM '18, Torino, 22–26 October 2018 : proceedings. – New York: Association for Computing Machinery, 2018. – P. 1703–1706. DOI: 10.1145/3269206.3269285.
11. Deep Bi-LSTM Networks for Sequential Recommendation / [C. Zhao, J. You, X. Wen, X. Li ] // Entropy (Basel). – 2020. – Vol. 22, Issue 8. – P. 870. DOI: 10.3390/e22080870.
12. Attention is all you need / [A. Vaswani, N. Shazeer, N. Parmar et al.] // Neural Information Processing Systems : Thirty-first international conference, NIPS'17, Long Beach, California, 04–09 December 2017 : proceedings. – New York : Curran Associates Inc., 2017. – P. 6000 – 6010.
13. Sequential Recommender System based on Hierarchical Attention Network / [H. Ying, F. Zhuang, F. Zhang et al.] // International Joint Conference on Artificial Intelligence : Twenty-seventh international joint conference, IJCAI '18, Stockholm, 13–19 July 2018 : proceedings. – Menlo Park: AAAI Press, 2018. – P. 3926–3932. DOI: 10.24963/ijcai.2018/546.
14. Sequential Recommendation via Stochastic Self-Attention / [Z. Fan, Z. Liu, Y. Wang et al.] // ACM Web Conference 2022, WWW '22, Lyon, 25–29 April 2022 : proceedings. – New York: Association for Computing Machinery, 2022. – P. 2036–2047. DOI: 10.1145/3485447.3512077.
15. Kipf T. N. Semi-Supervised Classification with Graph Convolutional Networks / T. N. Kipf, M. Welling // International Conference on Learning Representations : Fifth international conference, ICLR 2017, Toulon, 24–26 April 2017 : proceedings. – New York : Curran Associates Inc., 2017. – DOI: 10.48550/arXiv.1609.02907.
16. A Comprehensive Survey of Graph Neural Networks for Knowledge Graphs / [Z. Wu, S. Pan, F. Chen et al.] // IEEE Transactions on Neural Networks and Learning Systems. – 2022. – Vol. 32, № 1. – P. 4–24. DOI: 10.1109/TNNLS.2020.2978386.
17. Attention-Based Recommendation On Graphs / [T. Hekmatfar, S. Haratizadeh, P. Razban, S. Goliaei] // ArXiv. – 2022. DOI: 2201.05499.
18. Hinton G. Distilling the knowledge in a neural network / G. Hinton, O. Vinyals, J. Dean // ArXiv. – 2015. DOI: 1503.02531.
19. Ba L. J. Do Deep Nets Really Need to be Deep? / L. J. Ba, R. Caruana // Advances in Neural Information Processing Systems. – 2014. – Vol. 27. – P. 2654–2662. DOI: 1312.6184.
20. Church K. W. Word association norms, mutual information, and lexicography // K. W. Church, P. Hanks // Computational Linguistics. – 1990. – Vol. 16, № 1. – P. 22–29.
21. Grover A. Node2vec: Scalable Feature Learning for Networks / A. Grover, J. Leskovec // ArXiv. – 2016. DOI: 1607.00653.


DOI 10.15588/1607-3274-2025-3-12