

УДК 004.891.2

Рекомендательные системы, основанные на последовательностях

Захаров Даниил Сергеевич

Кузьмина Инна Анатольевна (научный руководитель)

МГТУ им. Н.Э. Баумана, Москва, Россия

Моделирование предпочтений пользователей на основе их поведения — это сложная и важная задача из области рекомендательных систем. Современный метод, показывающий один из лучших результатов, основан на глубоком двунаправленном внимании. Этот механизм был заимствован из области обработки естественного языка и адаптирован для задачи рекомендательных систем. Несмотря на его эффективность, авторы полагают, что современные вариации механизма могут также быть адаптированы под задачу рекомендаций с целью улучшения точности, персонализации и производительности.

Ключевые слова: последовательные рекомендации, двунаправленное моделирование последовательностей, механизм внимания, разделенное внимание, улучшенная маска декодера

Введение. В современном мире стремительно растет объем информации. Поэтому многие бизнес-задачи вынуждены адаптироваться, предоставляя клиентам возможность быстро находить релевантные данные, удовлетворяющие его потребностям. Рекомендательные системы призваны справиться с этой задачей, путем предсказания последующих действий каждого пользователя.

В последние годы архитектура трансформеров [1] произвела прорыв в глубоком обучении. Множество задач в области обработки естественного языка и, в частности, моделирования последовательностей различной природы удалось решить с невозможной ранее точностью.

Поэтому подход не обошел стороной сферу рекомендательных систем основанных на последовательностях взаимодействий пользователей и предметов рекомендаций. Почти сразу же после появления модели Bidirectional Encoder Representations from Transformers (BERT) [2], которая основана на двунаправленном механизме внимания, появилась ее адаптация способная решать задачу рекомендаций BERT для рекомендаций (BERT4rec) [3].

Появление множества вариаций архитектуры BERT, вызванное ее популярностью, предоставляет возможность обобщения этих вариаций на задачу рекомендаций с целью увеличения точности, уровня персонализации и эффективности рекомендаций.

Так, архитектура Decoding-enhanced BERT with disentangled attention (DeBERTa) [4] улучшает оригинальный BERT, используя две новые техники. Первая техника — это распутанный механизм внимания, который представляет каждый элемент обрабатываемой последовательности в виде векторов, кодирующих последовательность и содержание, а затем использует отдель-

ные матрицы весов внимания для обработки каждого. Вторая техника представляет собой использование декодера, который учитывает абсолютные позиции элементов для получения предсказаний.

Формальная постановка задачи. Для формальной постановки задачи необходимо ввести ряд определений, которые будут использованы в работе в дальнейшем.

Пусть $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$ обозначает множество пользователей, где u — идентификатор пользователя; $|\mathcal{U}|$ — количество уникальных идентификаторов пользователей в системе.

Аналогично обозначим $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$ множество товаров в системе, где v — идентификатор товара; $|\mathcal{V}|$ — количество уникальных идентификаторов в системе.

Комбинируя эти два понятия, обозначим набор взаимодействий пользователя и товара как $\mathcal{S}_u = [v_1^{(u)}, \dots, v_t^{(u)}, \dots, v_{n_u}^{(u)}]$ последовательность, где $u \in \mathcal{U}$ пользователи из их множества; $v_t^{(u)} \in \mathcal{V}$ товары, с которыми u взаимодействовал на временном шаге t ; n_u — длина последовательности взаимодействий с товаром для пользователя.

Важно отметить, что индекс времени t используется как номер взаимодействий, это упрощает учет порядка в последовательности.

Тогда, основываясь на введенных понятиях, можно формально поставить задачу, решаемую рекомендательными системами.

Дана история взаимодействий $\mathcal{S}_u$, цель последовательной рекомендательной системы предсказать товар, с которым пользователь провзаимодействует на временном шаге $n_u + 1$.

Также это может быть формализовано как модель вероятности для всех возможных товаров для пользователя u на временном шаге $n_u + 1$:

$$P(v_{n_u+1}^{(u)} = v | \mathcal{S}_u).$$

Обзор существующих подходов. Одним из классических подходов для решения поставленной задачи является коллаборативная фильтрация [5, 6]. Наиболее популярной вариацией этого подхода является матричная факторизация, которая делает прогнозы на основании скалярного произведения векторов пользователя и товара из одного векторного пространства [5, 7, 8].

С развитием глубокого обучения появились более сложные методы. Одним из первых подходов этого поколения была ограниченная машина Больцмана для коллаборативной фильтрации [9].

Однако подходы, описываемые в рассмотренных работах, объединяют в себе существенный недостаток. Порядок, с которым пользователь взаимодействовал с предметом рекомендаций не учитывается.

Первые попытки решить проблему предпринимались с использованием цепей Маркова. Подробно их исследование представлено в статье [10]. Хотя этот подход решал поставленную задачу, у него были значительные несовершенства. Он полагает что будущее состояние зависит только от текущего,

алгоритм не способен учитывать дополнительную информацию о предмете рекомендаций, а сложность растет экспоненциально при масштабировании.

Модели, основанные на рекуррентных нейронных сетях, смогли устранить выявленные недостатки. Работа архитектур такого типа основана на кодировании истории взаимодействий в вектор, который используется для предсказаний. Ключевой особенностью рекуррентных сетей является учет состояния на предыдущем шаге, что дает преимущества перед Марковскими цепями. Наиболее известные модели этого типа Gated Recurrent Unit (GRU) [11] и Long Short-Term Memory (LSTM) [12]. Ввиду способности моделировать последовательности они стали очень популярными в рекомендательных системах и рассмотрены в следующих статьях.

Несмотря на эффективность существовавших на момент появления в 2017 году архитектуры BERT4rec, она смогла показать лучшие результаты на задаче рекомендаций. Добиться это удалось благодаря механизму внимания, модификации которого будут рассмотрены в рамках этой статьи на задаче рекомендаций.

Описание модели BERT4rec. Модель основывается на механизме двунаправленного внимания.

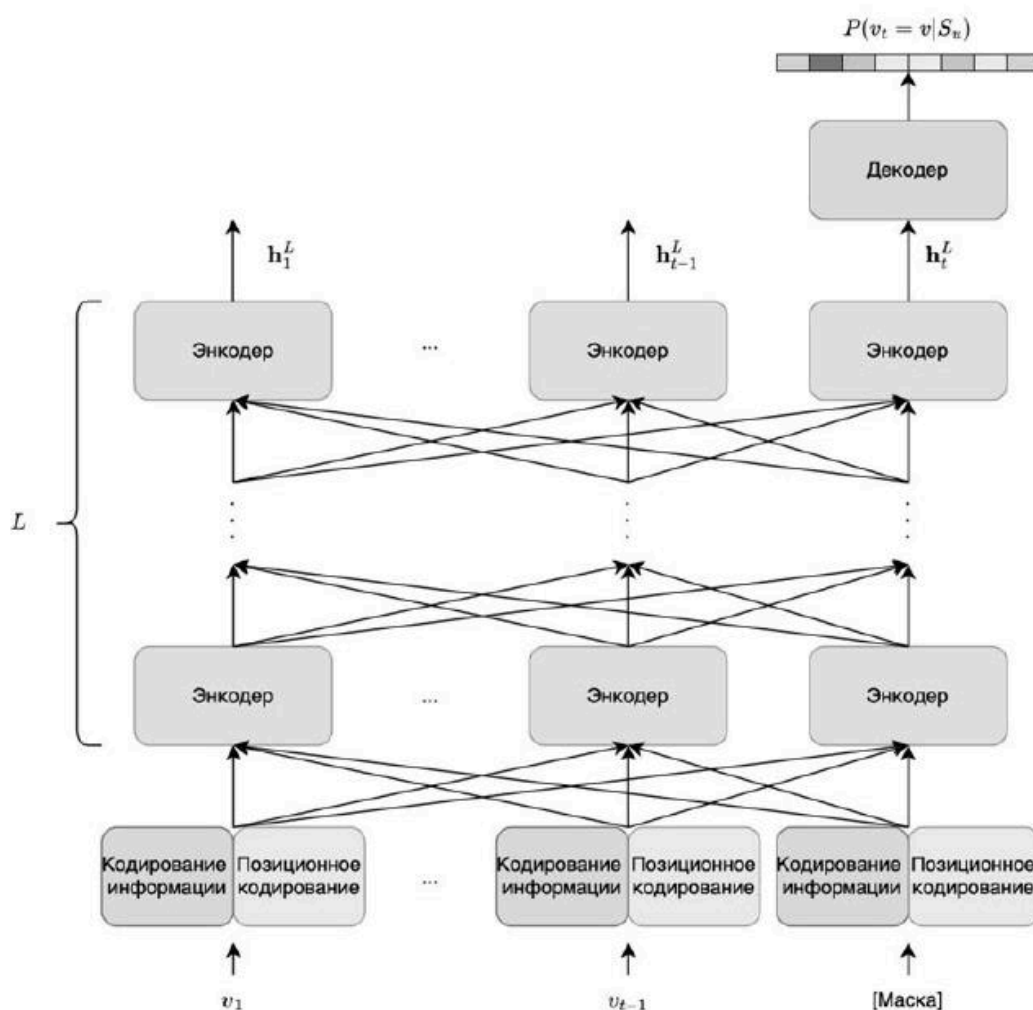


Рис. 1. Схема архитектуры BERT4rec

Как показано на рис. 1, модель состоит из составленных друг на друга L энкодеров. На каждом слое ℓ модель итеративно уточняет представление каждой позиции $\mathbf{h}_i^\ell$, обмениваясь информацией между всеми позициями предыдущего слоя, — это происходит параллельно благодаря механизму Transformer.

На вход передается последовательность $\mathcal{S}_u$ длиной t . По ней вычисляется скрытое представление для первого слоя формируется по формуле

$$\mathbf{h}_i^0 = \mathbf{v}_i + \mathbf{p}_i,$$

где $\mathbf{v}_i \in \mathbf{E}$ — векторное представление информации о предмете рекомендаций; $\mathbf{p}_i \in \mathbf{P}$ — векторное представление, кодирующее информацию о позиции товара.

Позиционное кодирование. Формирование позиционных векторных представлений необходимо поскольку энкодер устроен так, что архитектурно не учитывает информацию о позиции. Закладывая в скрытое представление такие вектора, проблема устраняется. Однако позиционные векторные представления накладывают ограничение на длину моделируемой последовательности.

Максимальная длина последовательности ограничена размерностью $\mathbf{P} \in \mathbb{R}^{N \times d}$, где N — максимальная длина последовательности; d — длина вектора позиционного представления.

Вектора $\mathbf{v}_i$ формируются с использованием стандартных подходов. Подробнее рассмотрим формулы для $\mathbf{p}_i$. Эти векторы формируются с использованием периодических функций:

$$\mathbf{p}_i(k, j) = \sin\left(\frac{k}{n^{\frac{j}{d}}}\right);$$

$$\mathbf{p}_i(k, 2j + 1) = \cos\left(\frac{k}{n^{\frac{j}{d}}}\right),$$

где k — индекс позиции входного элемента; j — индекс в формируемом векторе, принимает значения от 0 до $d - 1$; n — гиперпараметр, зависящий от количества пользователей в системе.

После того как вектора $\mathbf{h}_i^\ell$ скрытых представлений сформированы они попадают в энкодеры, общая структура каждого из которых представлена на рис. 2.

Каждый отдельный энкодер включает в себя слой многоголового внимания, позиционно зависимую полносвязную сеть и слой нормализации. Слой нормализации стандартный и не представляет интереса для исследования в рамках работы, поэтому описан не будет. Стоит отдельно отметить, что стрелки, соединяющие нормализацию с предыдущим слоем, означают сложение текущего состояния с предшествующим.



Рис. 2. Энкодер

Слой многоголового внимания (Multi-Head Self-Attention). Вектора скрытых представления каждого временного шага t объединяются в матрицу $\mathbf{H}^\ell \in \mathbb{R}^{t \times d}$. После чего матрица $\mathbf{H}^\ell$ обрабатывается функцией внимания (1), которая содержит собственные обучаемые параметры $\mathbf{W}$ для каждой головы. Функция (1), внимания возвращает head_i — выходные представления. $\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V})$ представлен формулой (2). Далее все head_i объединяются и еще раз умножаются на обучаемые веса $\mathbf{W}^O$:

$$\text{head}_i = \text{Attention}(\mathbf{H}^\ell \mathbf{W}_i^Q, \mathbf{H}^\ell \mathbf{W}_i^K, \mathbf{H}^\ell \mathbf{W}_i^V); \quad (1)$$

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d/h}}\right)\mathbf{V}; \quad (2)$$

$$\text{MH}(\mathbf{H}^\ell) = [\text{head}_1; \text{head}_2; \dots; \text{head}_h] \mathbf{W}^O, \quad (3)$$

где h — количество голов внимания; $\mathbf{Q}, \mathbf{K}, \mathbf{V}$ — проекции одной матрицы матрицы $\mathbf{H}^\ell$ с разными обучаемыми параметрами; $\mathbf{W}_i^Q \in \mathbb{R}^{d \times \frac{d}{h}}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times \frac{d}{h}}$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times \frac{d}{h}}$ и $\mathbf{W}^O \in \mathbb{R}^{d \times d}$ являются обучаемыми параметрами. Для простоты опущен индекс слоя ℓ , но важно отметить, что параметры проекций не разделяются между слоями.

Масштабирующий коэффициент $\sqrt{d/h}$ (температура) смягчает распределение внимания, предотвращая слишком малые градиенты.

Слой нормализации имеет стандартную реализацию, поэтому в данной работе не представляет интереса для изучения.

Позиционно-зависимая полносвязная сеть (Position-wise Feed-Forward Network) основана на линейных проекциях. Чтобы добавить нелинейность и взаимодействие между измерениями, применяется позиционно-зависимая полносвязная сеть к выходам подслоя внимания. Она состоит из двух аффинных преобразований с активацией GELU между ними:

$$\begin{aligned} \text{PFFN}(\mathbf{h}^\ell) &= \left[\text{FFN}(\mathbf{h}_t^\ell)^\top; \dots; \text{FFN}(\mathbf{h}_t^\ell)^\top \right]^\top; \\ \text{FFN}(\mathbf{x}) &= \text{GELU}(\mathbf{x}\mathbf{W}^{(1)} + \mathbf{b}^{(1)})\mathbf{W}^{(2)} + \mathbf{b}^{(2)}; \\ \text{GELU}(\mathbf{x}) &= \mathbf{x}\Phi(\mathbf{x}), \end{aligned}$$

где $\Phi(\mathbf{x})$ — функция распределения стандартного нормального распределения; $\mathbf{W}^{(1)} \in \mathbb{R}^{d \times 4d}$; $\mathbf{W}^{(2)} \in \mathbb{R}^{4d \times d}$; $\mathbf{b}^{(1)}$, $\mathbf{b}^{(2)}$ — обучаемые параметры, общие для всех позиций.

Декодер. После L слоев получается итоговое представление $\mathbf{H}^L$. Для предсказания замаскированного $\mathbf{v}_t$, который замаскировали на входе, обозначен на рисунке 1 как [Маска], используется двухслойная сеть с GELU:

$$P(v_t = v | \mathcal{S}_u) = \text{softmax}(\text{GELU}(\mathbf{h}_t^L \mathbf{W}^P + \mathbf{b}^P) \mathbf{E}^\top + \mathbf{b}^O),$$

где $\mathbf{W}^P$, $\mathbf{b}^P$, $\mathbf{b}^O$ — обучаемые параметры; $\mathbf{E}$ — матрица эмбеддингов элементов (общая для входного и выходного слоев для уменьшения переобучения).

Описание улучшенной модели. Модель deBERTa реализует два новых улучшения: разделенное многоголовое внимание и улучшенный декодер маски.

Разделенное многоголовое внимание представляет из себя версию многоголового внимания, которая по отдельности обрабатывает векторные представления относительных позиций и контентной информации.

Обозначим k как максимальное относительно расстояние, $\delta(i, j) \in [0, 2k]$ как относительное расстояние от товара i до товара j в последовательности $\mathcal{S}_u$ как

$$\delta(i, j) = \begin{cases} 0 & \text{для } i - j \leq -k; \\ 2k - 1 & \text{для } i - j \geq k; \\ i - j + k & \text{иные случаи.} \end{cases} \quad (4)$$

Используя (4), разделенное внимание определяется формулами

$$\mathbf{Q}_c = \mathbf{H}\mathbf{W}_{q,c}, \quad \mathbf{K}_c = \mathbf{H}\mathbf{W}_{k,c}, \quad \mathbf{V}_c = \mathbf{H}\mathbf{W}_{v,c};$$

$$\mathbf{Q}_r = \mathbf{R}\mathbf{W}_{q,r}, \quad \mathbf{K}_r = \mathbf{R}\mathbf{W}_{k,r};$$

$$\widetilde{\mathbf{A}}_{i,j} = \underbrace{\mathbf{Q}_i^c \mathbf{K}_j^{cT}}_{(a)} + \underbrace{\mathbf{Q}_i^c \mathbf{K}_{\delta(i,j)}^r}_{(b)} + \underbrace{\mathbf{K}_j^c \mathbf{Q}_{\delta(j,i)}^r}_{(c)};$$

$$H_o = \text{softmax} \left(\frac{\tilde{A}}{\sqrt{3d}} \right) V_c,$$

где Q_c, K_c, V_c — проекции векторов контента; $W_{q,c}, W_{k,c}, W_{v,c} \in \mathbb{R}^{d \times d}$ — матрицы обучаемых параметров контента; $R \in \mathbb{R}^{2k \times d}$ — матрица векторных представлений относительных позиций; Q_r, K_r — проекции R ; $W_{q,r}, W_{k,r}$ — матрицы обучаемых параметров относительных позиций. Поскольку все остальные блоки неизменны, структура энкодера аналогична рис. 3 за исключением измененного блока внимания.

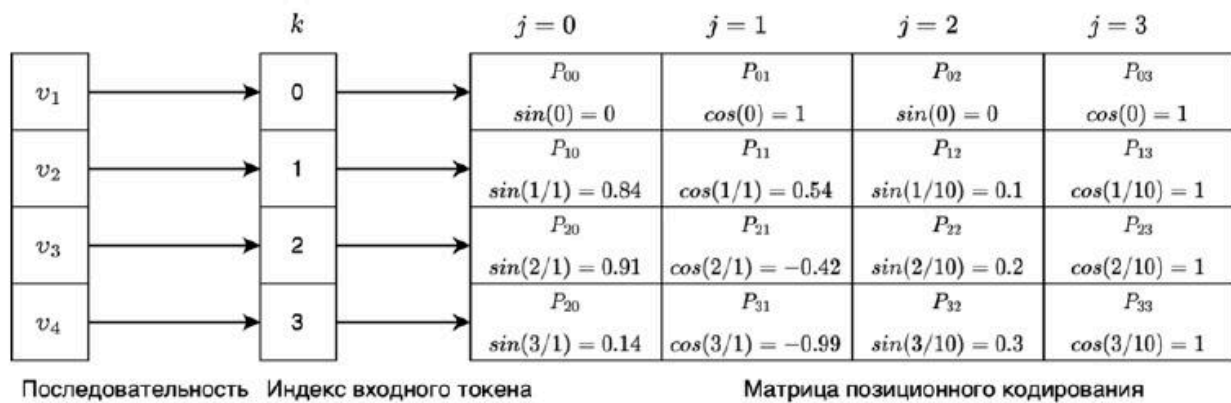


Рис. 3. Пример работы позиционного кодирования

Улучшенная маска декодера. Вместо двухслойной сети с GELU в качестве декодера используется энкодер с одним изменением. Контентная проекция Q_c вычисляется с использованием векторных представлений абсолютных позиций:

$$Q_c = (H + P)W_{q,c}.$$

Вычислительные эксперименты. Модели BERT4rec и DeBERTa4rec обучались и тестировались на двух эталонных наборах данных, которые используются в научной литературе для сравнения точности работы моделей. Модели обучались на 10 эпохах на каждом наборе данных, а разбиения обучающего набора производилось при помощи техники leave-one-out. Статистики для наборов данных представлены в табл. 1.

Таблица 1

Статистики наборов данных

Название	$ U $	$ V $	Количество действий	Средняя n_u	Плотность
ML-1m	6040	3416	1 мил-н	163,5	4,79%
ML-20m	138,493	26,744	20 мил-нов	144,4	0.54%

После каждой эпохи производилась валидация на метриках NDCG и Recall. Лучшая модель по NDCG сохранялась, а после завершения обучения измерялось качество на тестовой подвыборке. Результаты и сравнение реализованных архитектур представлены в табл. 2.

Таблица 2

Значения метрик

Набор данных	Метрика	BERT4rec	DeBERTa	Улучшение
ML-20m	Recall@100	0,985217	0,985104	-0,01 %
	Recall@50	0,682441	0,098572	2,31 %
	Recall@20	0,512476	0,554374	7,56 %
	Reccall@10	0,401283	0,443692	9,56 %
	Recall@5	0,298415	0,332517	10,26 %
	Recall@1	0,128346	0,139872	8,24 %
	NDCG@100	0,365219	0,382104	4,42 %
	NDCG@50	0,326845	0,348912	6,32 %
	NDCG@20	0,274639	0,301852	9,02 %
	NDCG@10	0,243872	0,268945	9,32 %
	NDCG@5	0,210456	0,231904	9,25 %
	NDCG@1	0,128346	0,139872	8,24 %
ML-1m	Recall@100	0,992105	0,992599	0,05 %
	Recall@50	0,752029	0,778088	3,35 %
	Recall@20	0,536933	0,591393	9,21 %
	Reccall@10	0,413322	0,463487	10,82 %
	Recall@5	0,309613	0,346199	10,57 %
	Recall@1	0,135727	0,142233	4,57 %
	NDCG@100	0,370047	0,387922	4,61 %
	NDCG@50	0,331627	0,353553	6,20 %
	NDCG@20	0,289221	0,316538	8,63 %
	NDCG@10	0,258098	0,284306	9,22 %
	NDCG@5	0,224589	0,246443	8,87 %
	NDCG@1	0,135727	0,142233	4,57 %

Заключение. Полученные результаты позволяют судить о улучшении точности работы модели, поскольку были улучшены почти все метрики на двух эталонных наборах данных. Однако ожидалось получить больший прирост метрик качества. Это может быть связано с неточностью в реализации метода. Дальнейшее выявление неточностей в реализации и масштабирование на другие наборы данных представляет перспективы развития работы.

Литература

- [1] Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A.N., Kaiser Ł., Polosukhin I. Attention Is All You Need. Proceedings of NIPS, 2017, pp. 6000–6010. <https://doi.org/10.48550/arXiv.1706.03762>
- [2] Devlin J., Chang M.-W., Lee K., Toutanova K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Proceedings of NAACL-HLT, 2019, pp. 4171–4186. Arxiv:1810.04805v2.
- [3] Sun F., Liu J., Wu J., Pei C., Lin X., Ou W., Jiang P. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM), 2019, pp. 1441–1450. Arxiv:1904.06690v1.
- [4] Pengcheng H., Xiaodong L., Jianfeng G., Weizhu C., DeBERT: Decoding-enhanced BERT with Disentangled Attention. Proceedings of ICLR, 2021. Arxiv:2006.03654v6.
- [5] Koren Y., Bell R. Advances in Collaborative Filtering. Recommender Systems Handbook. Springer US, Boston, 2011, pp. 145–186.
- [6] Sarwar B., Karypis G., Konstan J., Riedl J. Item-based Collaborative Filtering Recommendation Algorithms. Proceedings of WWW. New York, 2001, pp. 285–295.
- [7] Koren Y., Bell R., Volinsky C. Matrix Factorization Techniques for Recommender Systems. Computer, 2009, vol. 42, no 8, pp. 30–37.
- [8] Salakhutdinov R., Mnih A. Probabilistic Matrix Factorization. Proceedings of NIPS. Curran Associates Inc., USA, 2007, pp. 1257–1264.
- [9] Salakhutdinov R., Mnih A., Hinton G. Restricted Boltzmann Machines for Collaborative Filtering. Proceedings of ICML, 2007, pp. 791–798. <https://doi.org/10.1145/1273496.1273596>
- [10] Shani G., Heckerman D., Brafman R.I. An MDP-Based Recommender System. J. Mach. Learn. Res., 2005, vol. 6, pp. 1265–1295.
- [11] Cho K., van Merriënboer B., Gulcehre C., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Proceedings of EMNLP, 2014, pp. 1724–1734. <https://doi.org/10.48550/arXiv.1406.1078>
- [12] Hochreiter S., Schmidhuber J. Long Short-Term Memory. Neural Computation, 1997, vol. 9, no. 8, pp. 1735–1780.

Sequence - based recommendation systems

Zakharov Daniil Sergeevich

Kuzmina Inna Anatolievna (*scientific advisor*)

BMSTU, Moscow, Russia

Modeling users' preferences based on their behavior is a complex and important task in the field of recommendation systems. The modern method, which shows state-of-the-art results, is based on deep bidirectional self-attention. This mechanism was borrowed from the field of natural language processing and adapted for the task of recommendation systems. Despite its effectiveness, I believe that modern variations of the mechanism can also be adapted to the task of recommendations in order to improve accuracy, personalization and performance.

Keywords: sequential recommendation, bidirectional sequential model, attention mechanism, disentangled attention mechanism, enhanced mask decoder