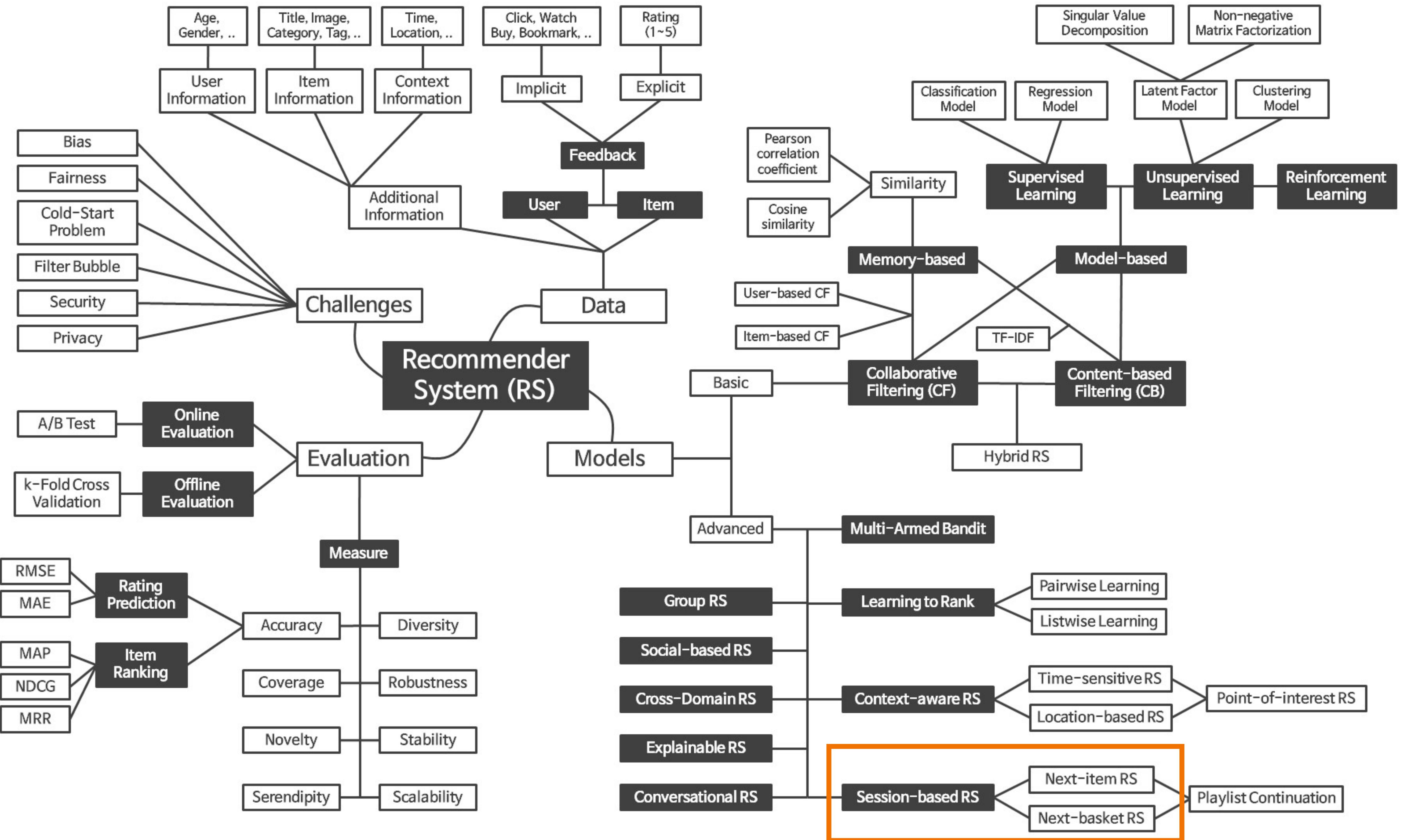


Sequential-based подходы

Краснов Александр, 14.04.2025, AI masters



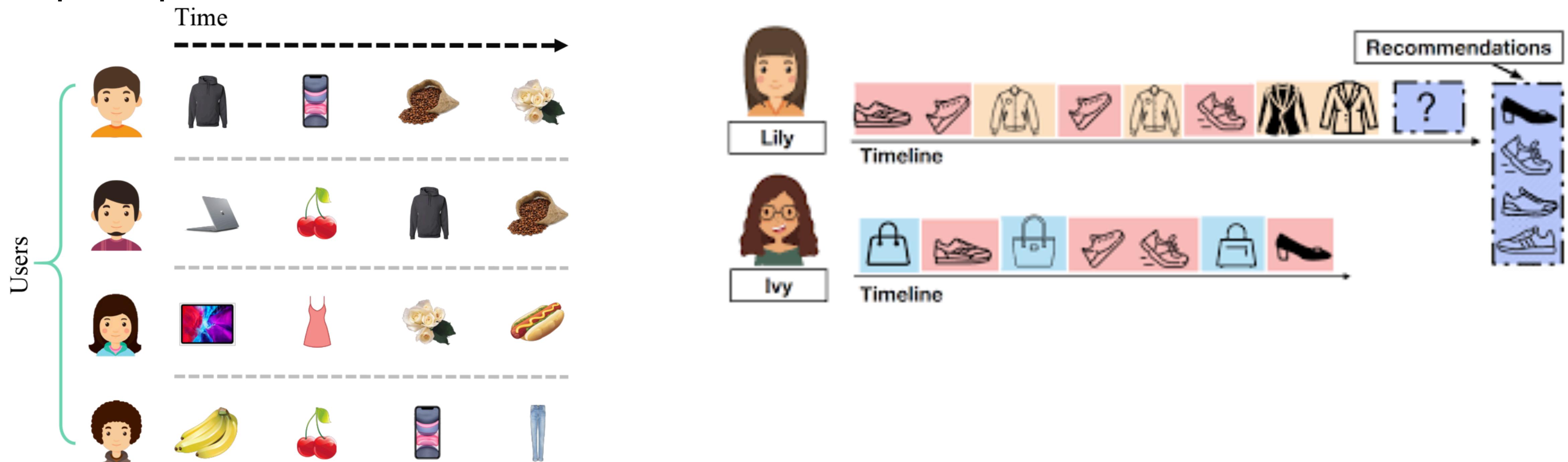
Последовательности

Откуда?

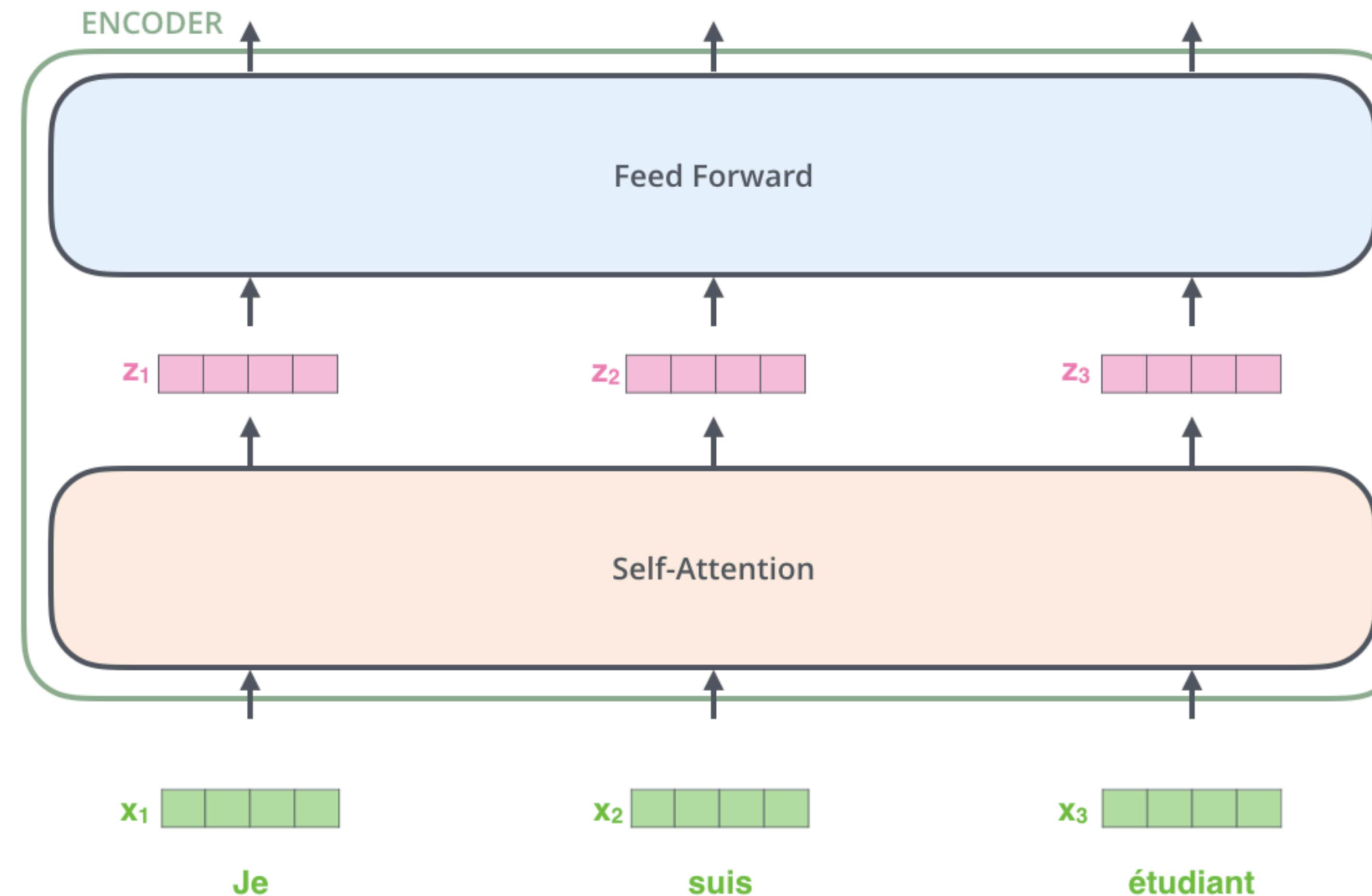
- История просмотров/кликов/покупок (например, для видео или товаров)
 - видео, товары, музыка, прочий контент
- Формат данных
 - [user_id, [item_1, item_2, ... item_N]]
 - [user_id, [(item_1, timestamp_1), (item_2, timestamp_2)...]]
 - [user_id, [(item_1, timestamp_1, action_1), (item_2, timestamp_2, action_2)...]]
 - добавить rating
 - и так далее

Последовательности

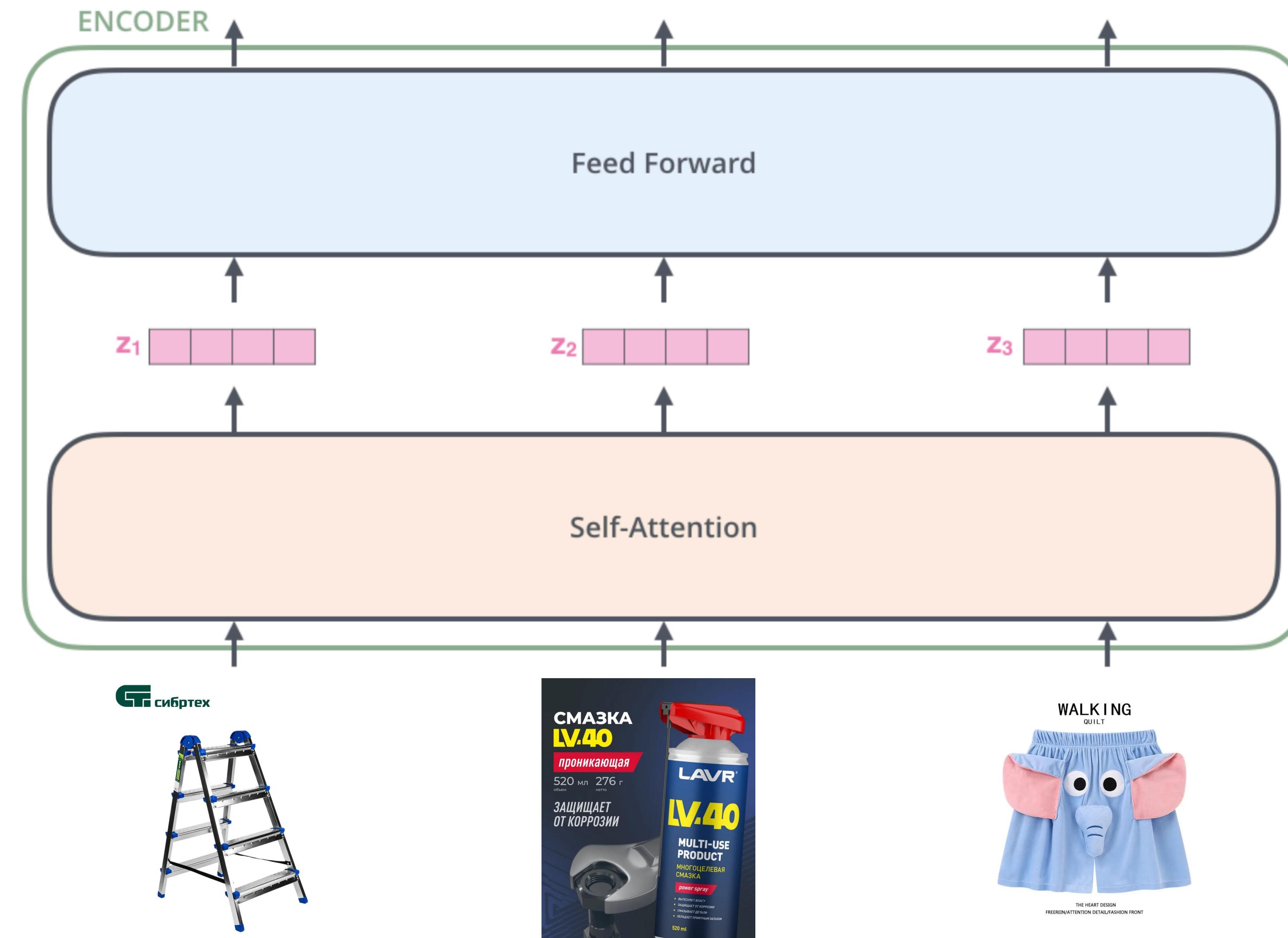
Примеры



NLP/IR -> RS

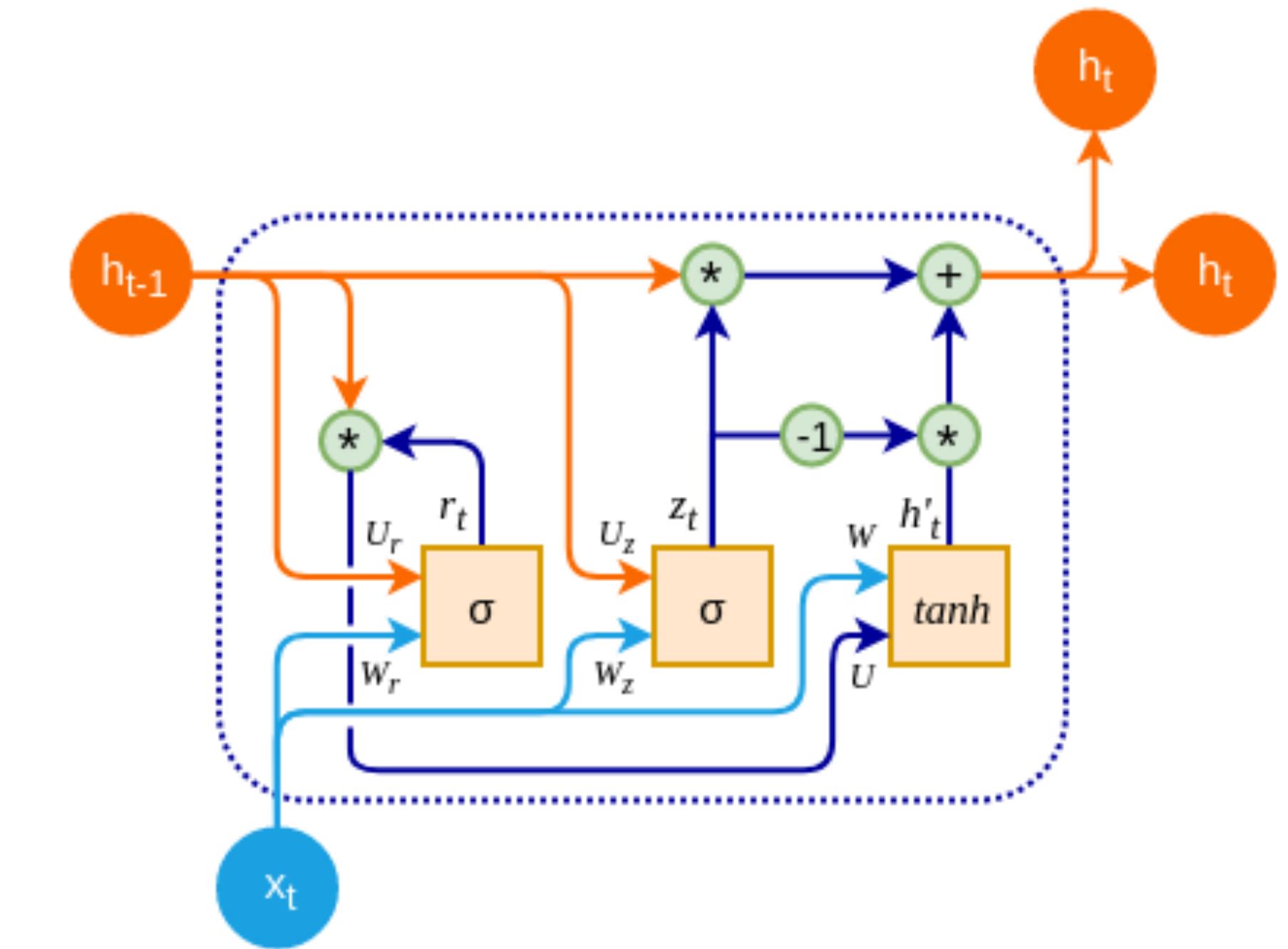
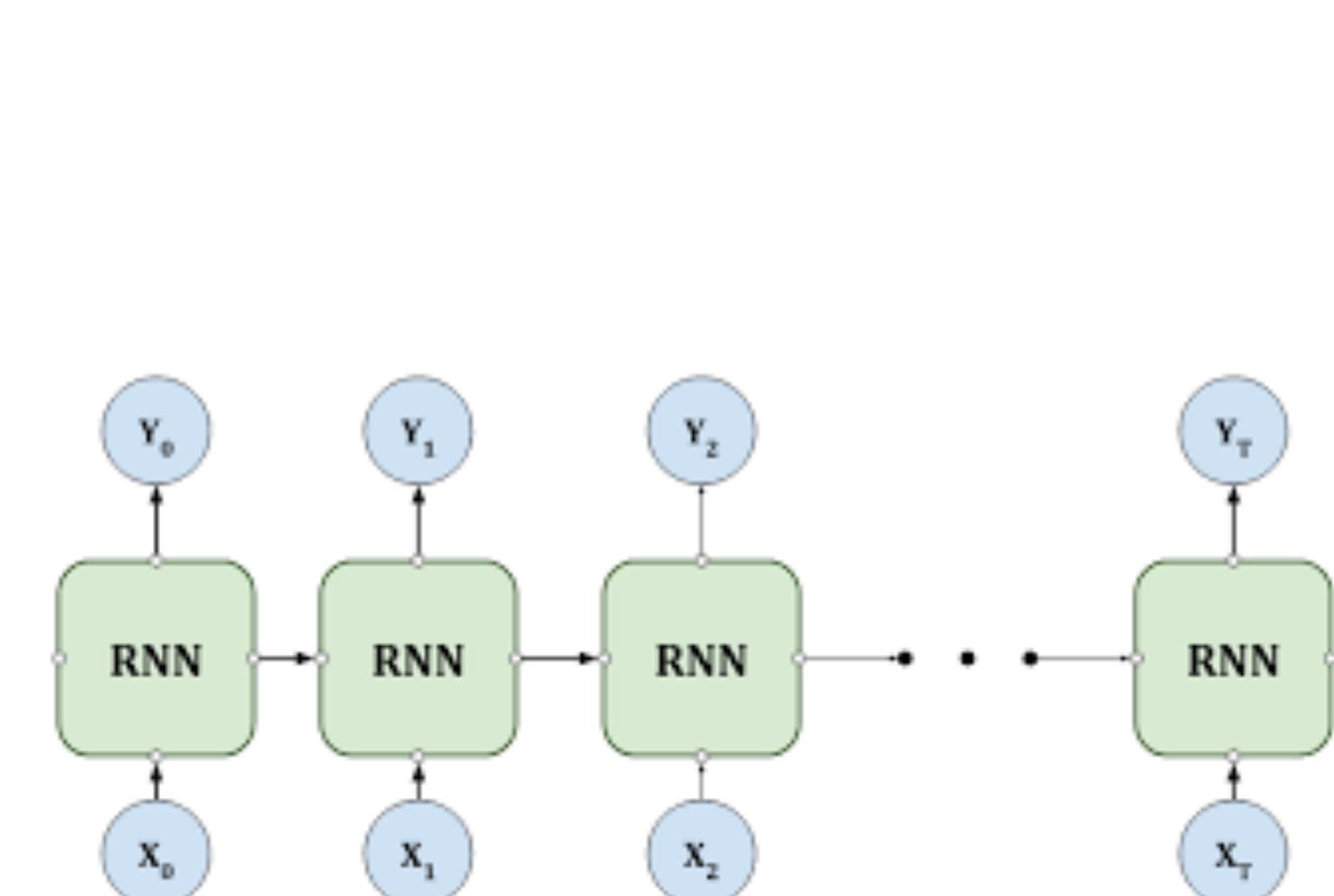
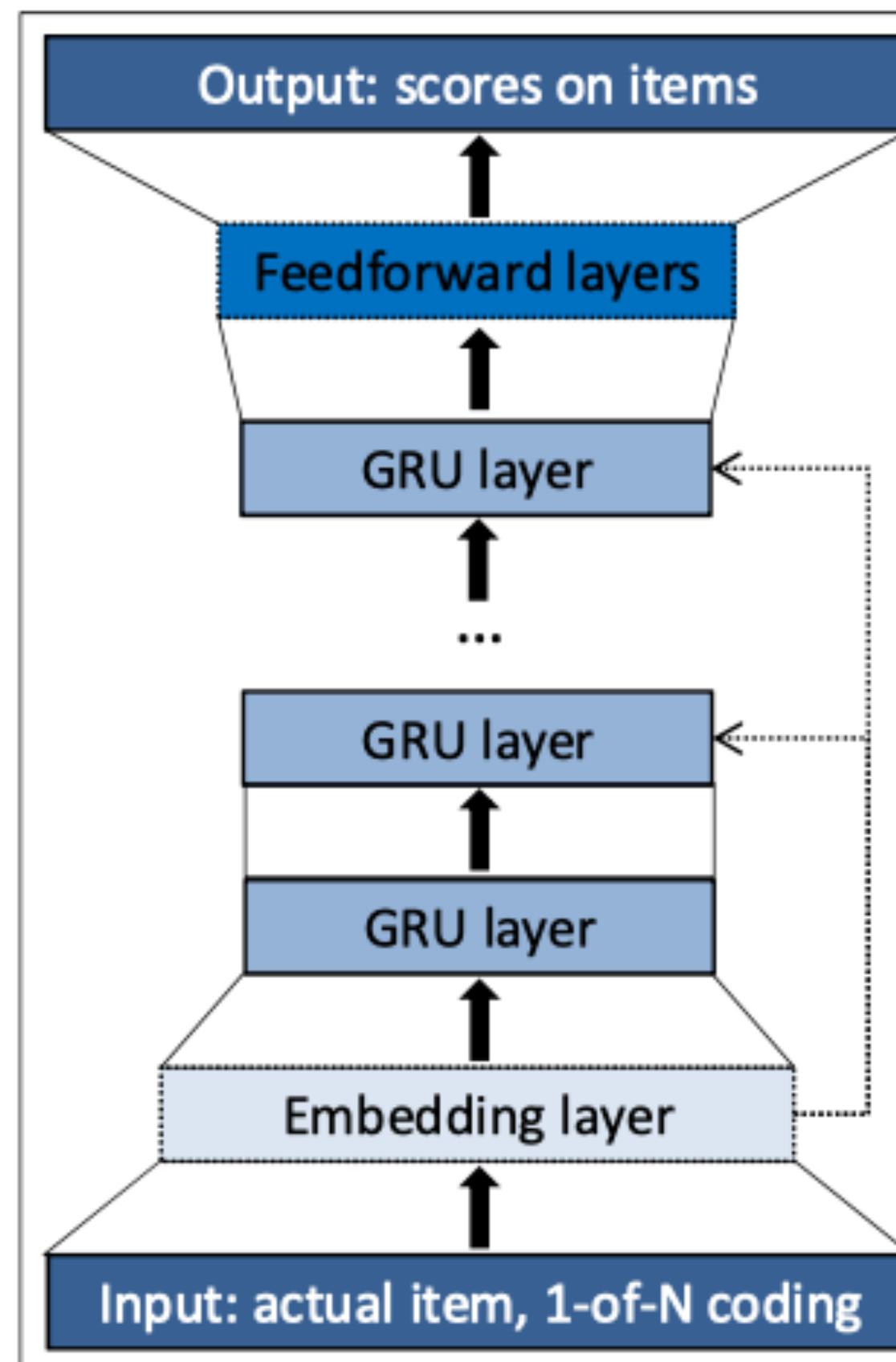


NLP/IR -> RS



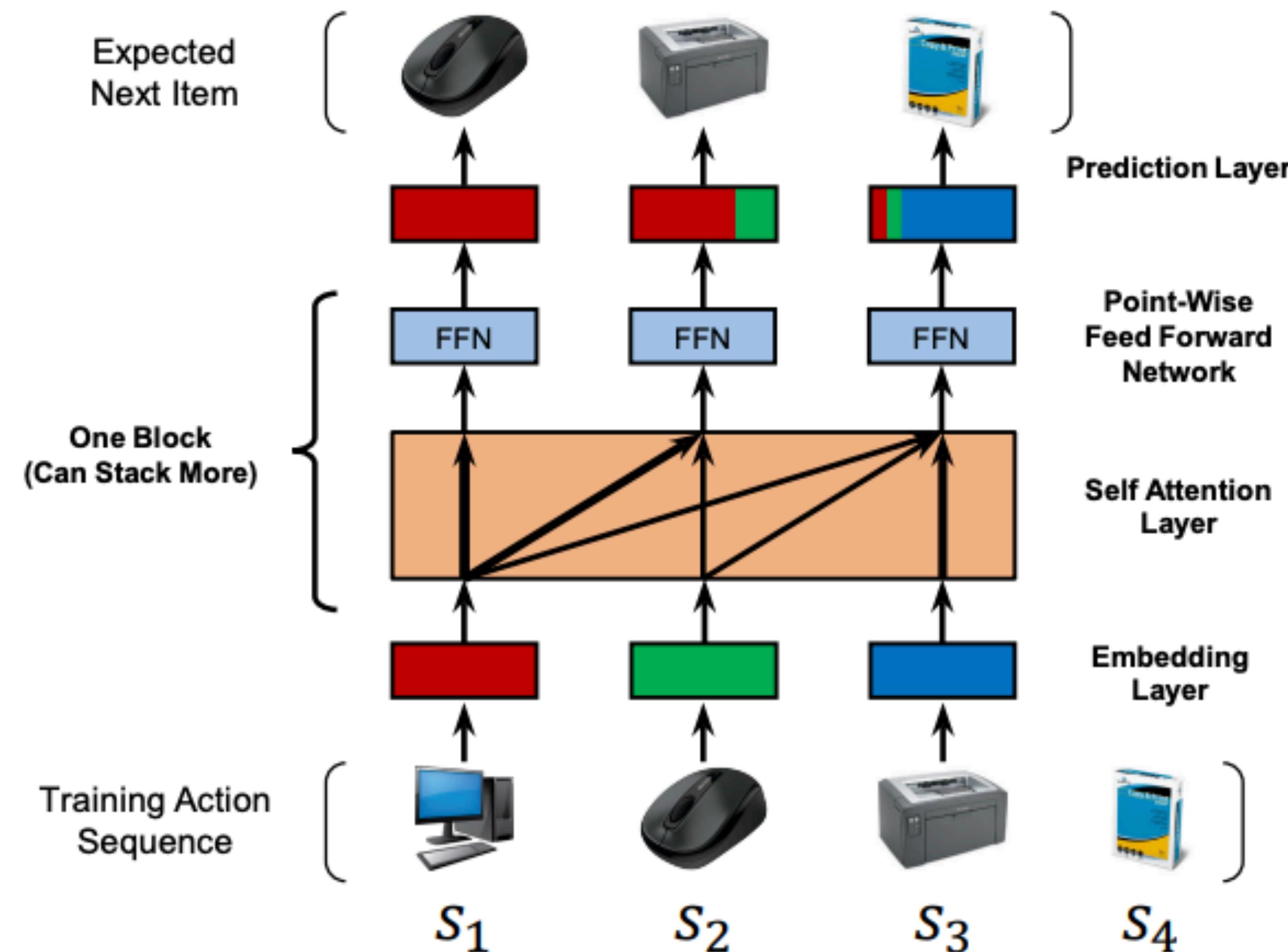
Спасибо за внимание!

Шаг назад, GRU4Rec (2015)



Session-based Recommendations with Recurrent Neural Networks

SASRec (2018)



Self-Attentive Sequential Recommendation

SASRec

Детали

- Causality: не смотрим в будущее
- Shared Item Embedding: на входе и выходе подаются одинаковые эмбеддинги айтемов
- Обучаемый positional embedding

SASRec

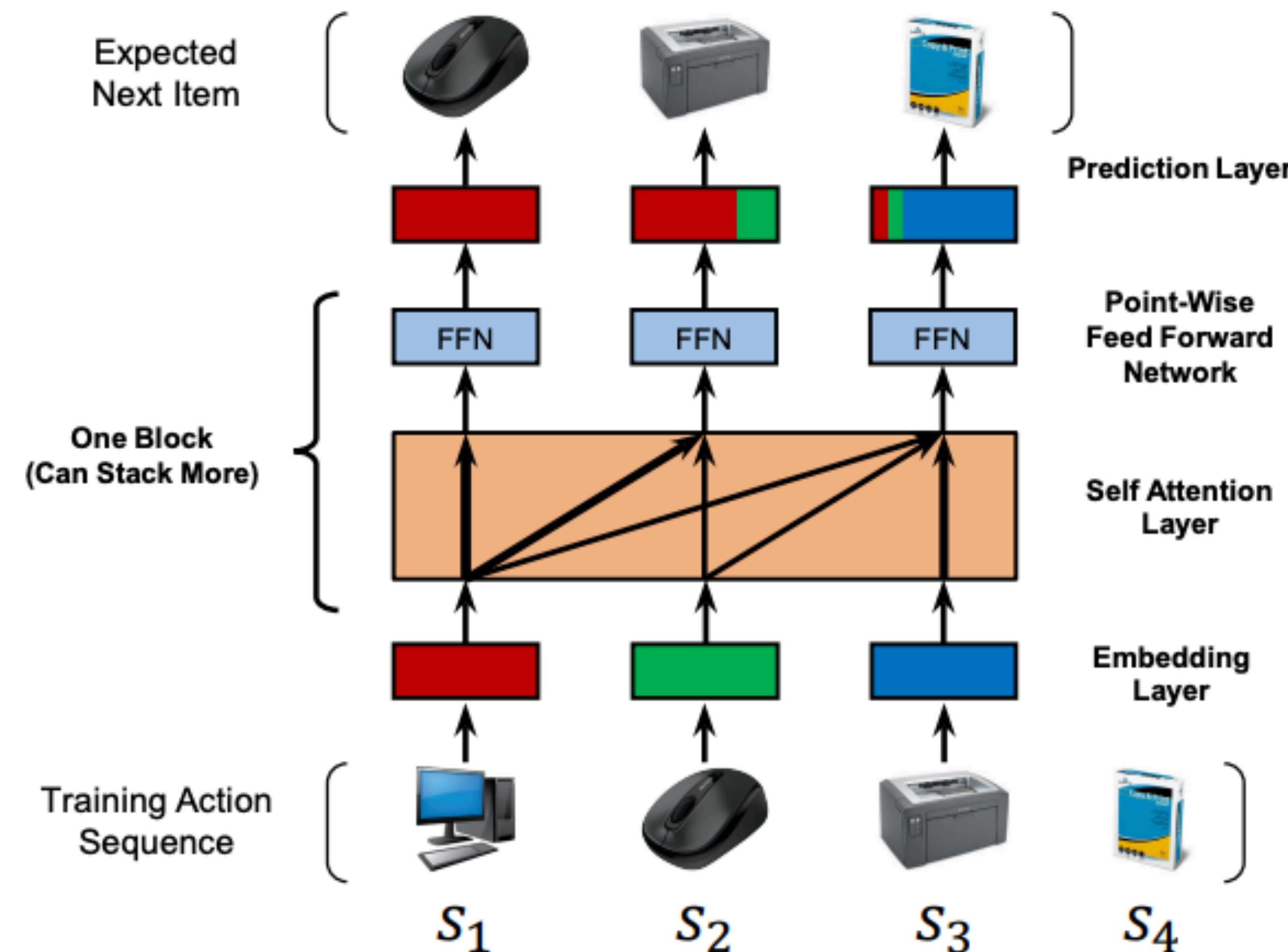
Лосс

$$-\sum_{\mathcal{S}^u \in \mathcal{S}} \sum_{t \in [1, 2, \dots, n]} \left[\log(\sigma(r_{o_t, t})) + \sum_{j \notin \mathcal{S}^u} \log(1 - \sigma(r_{j, t})) \right].$$

Note that we ignore the terms where $o_t = \text{<pad>}$.

SASRec

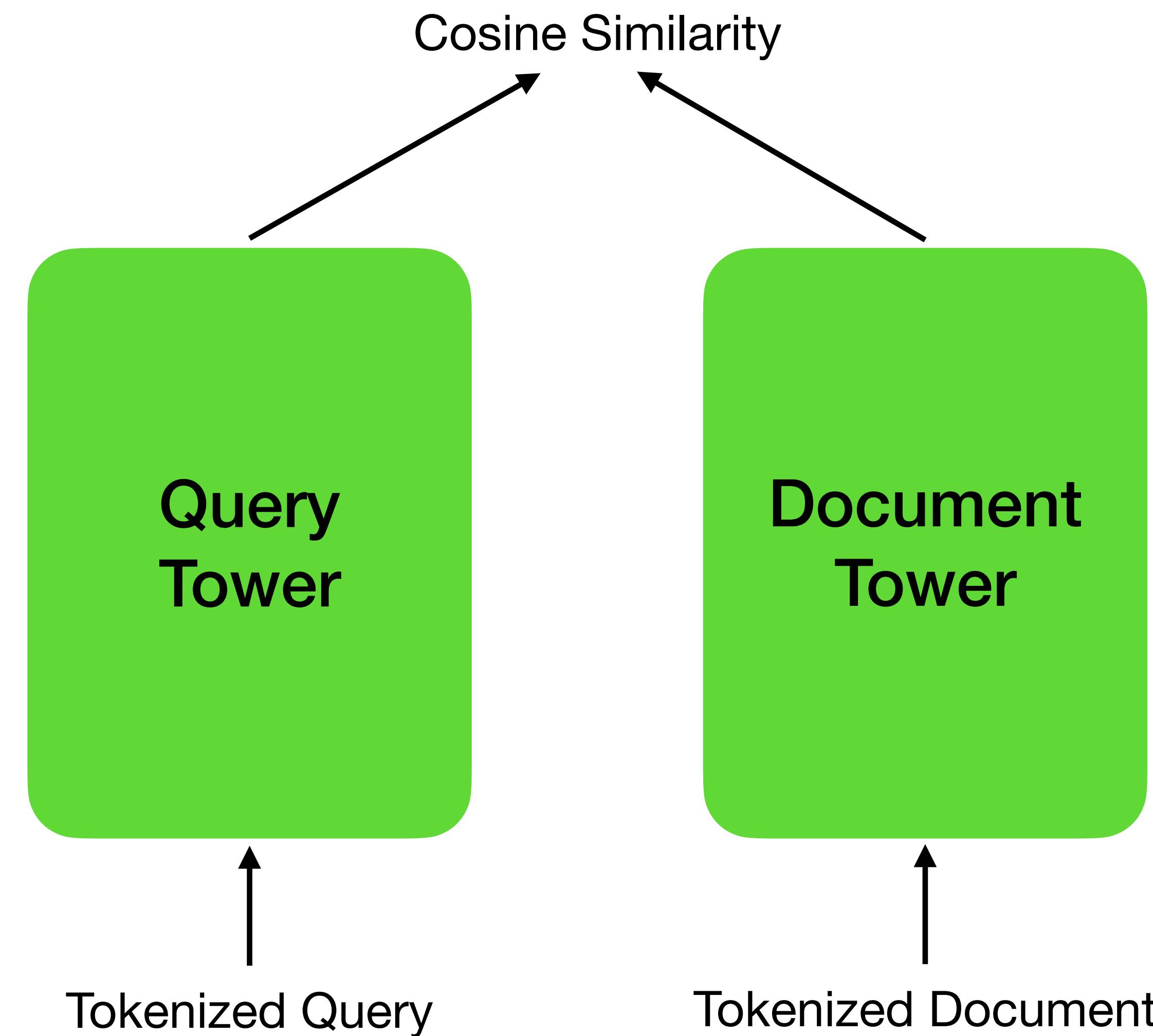
Predict



Self-Attentive Sequential Recommendation

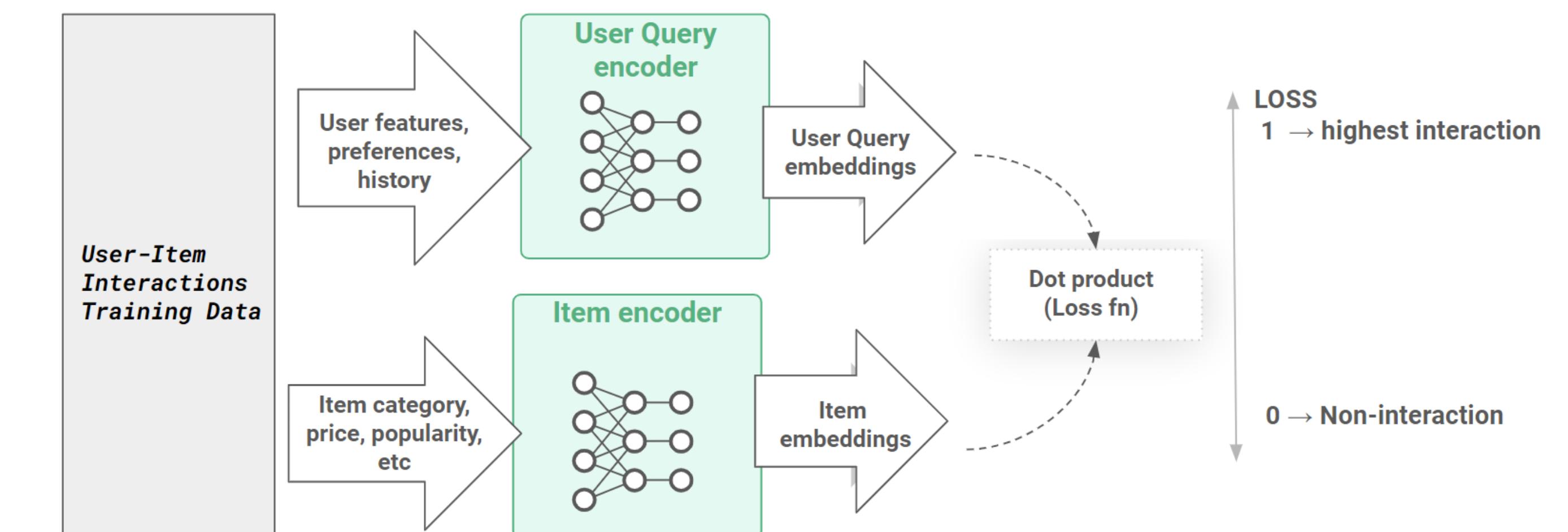
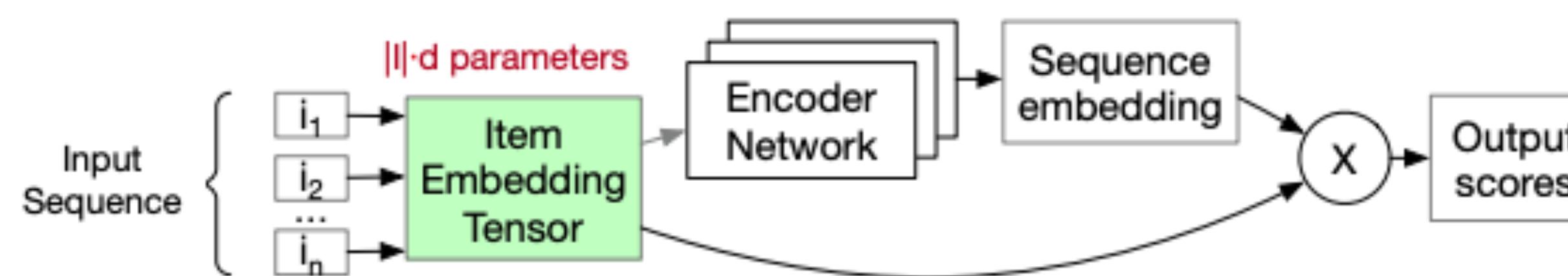
Two tower

Поиск



Two tower

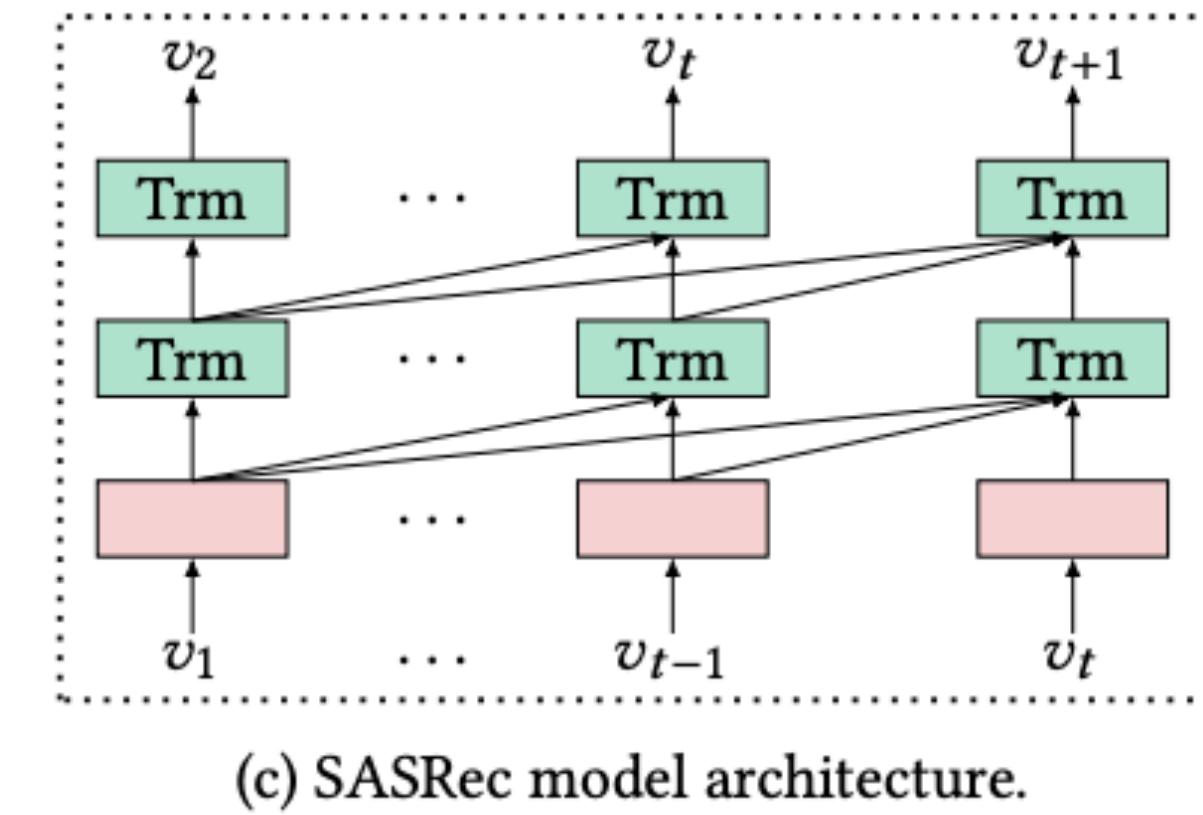
Рекомендации



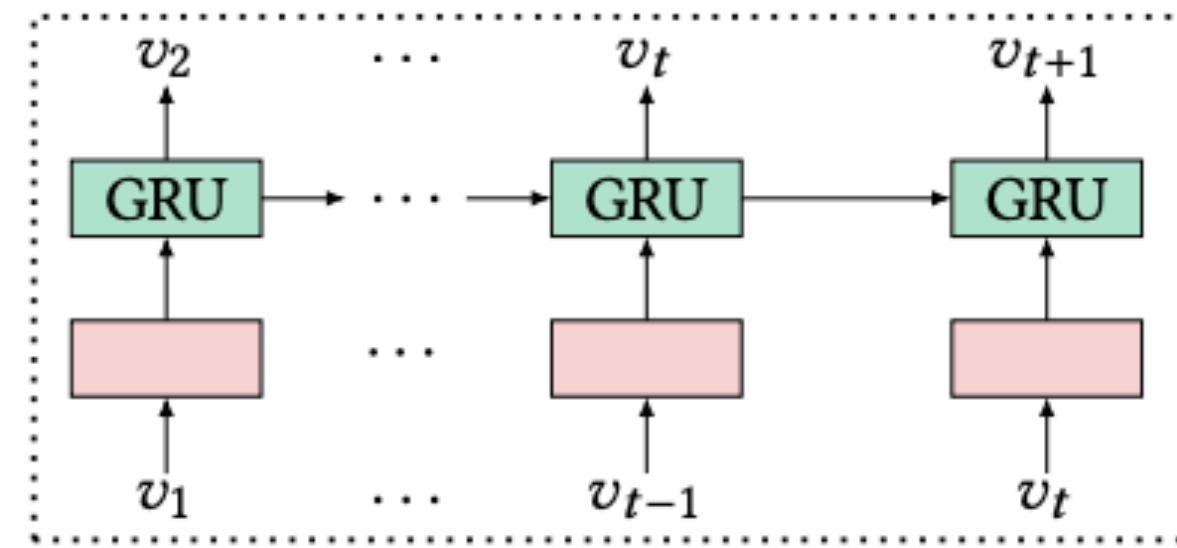
[RecJPQ: Training Large-Catalogue Sequential Recommenders](#)

<https://www.hopworks.ai/dictionary/two-tower-embedding-model>

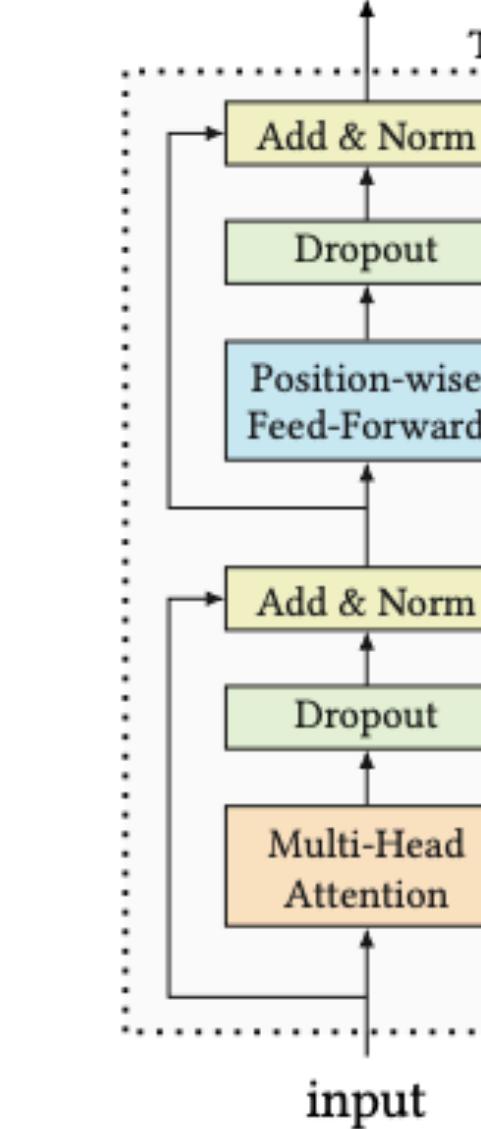
BERT4Rec (2019)



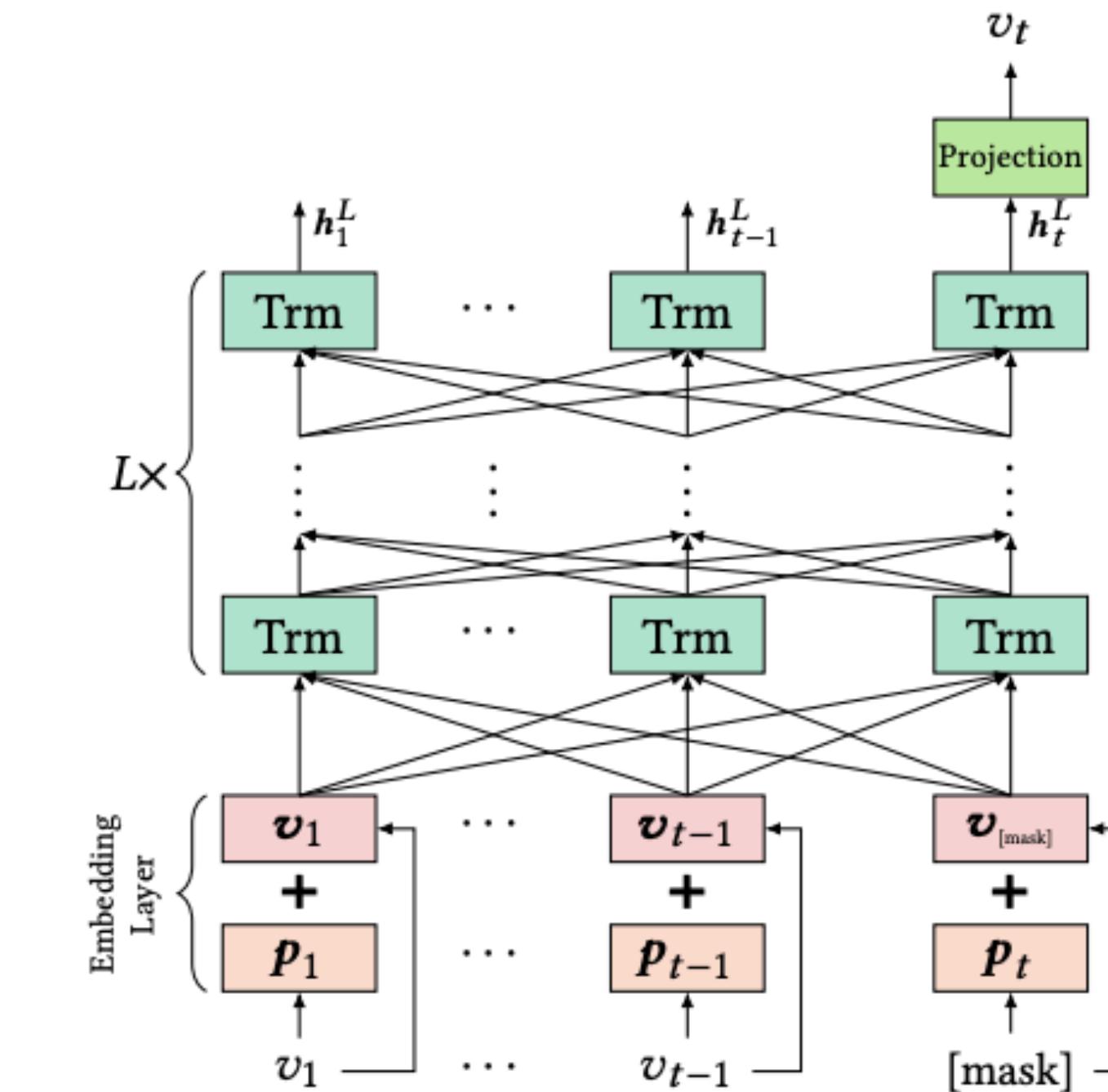
(c) SASRec model architecture.



(d) RNN based sequential recommendation methods.



(a) Transformer Layer.



(b) BERT4Rec model architecture.

BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

BERT4Rec

Лосс

Input: $[v_1, v_2, v_3, v_4, v_5] \xrightarrow{\text{randomly mask}} [v_1, [\text{mask}]_1, v_3, [\text{mask}]_2, v_5]$

Labels: $[\text{mask}]_1 = v_2, [\text{mask}]_2 = v_4$

$$\mathcal{L} = \frac{1}{|S_u^m|} \sum_{v_m \in S_u^m} -\log P(v_m = v_m^* | S'_u)$$

BERT4Rec

Детали

- Вся последовательность в attention
- Predict - добавляем маску в конце
- Softmax в конце

BERT4Rec

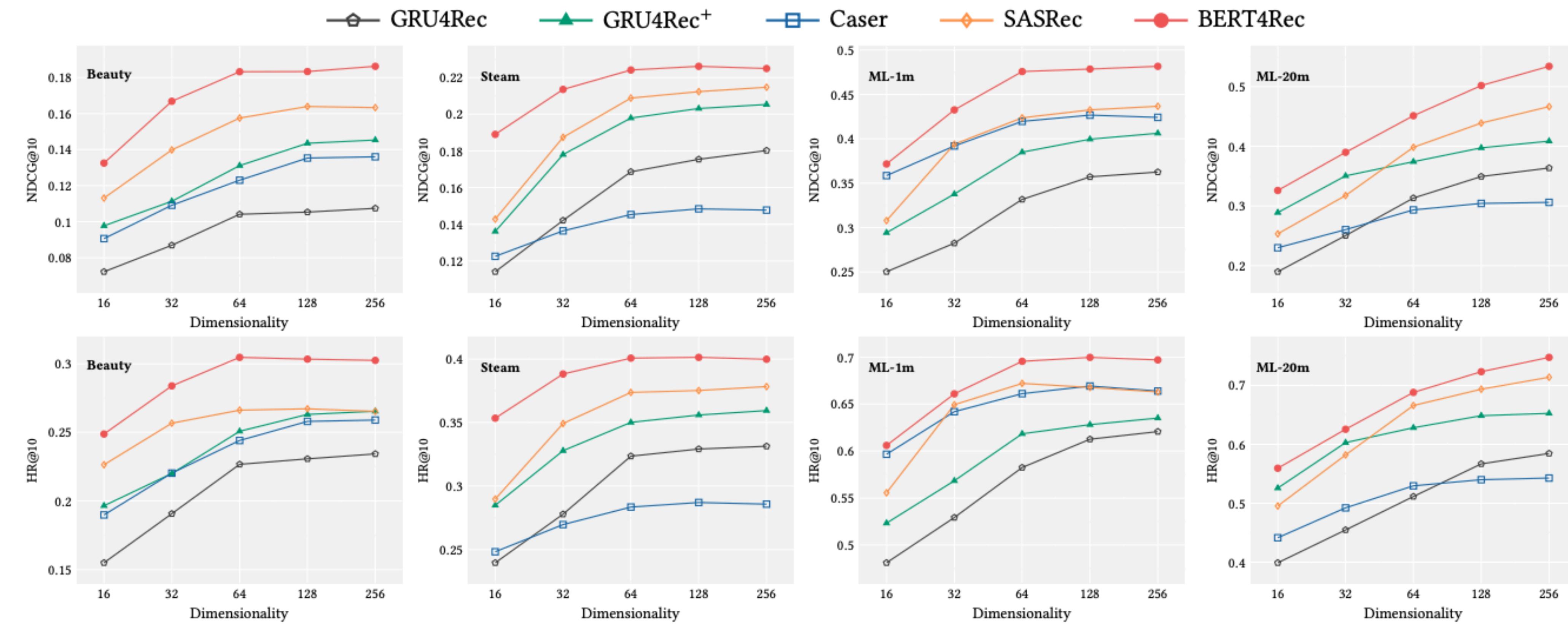


Figure 3: Effect of the hidden dimensionality d on HR@10 and NDCG@10 for neural sequential models.

gSASRec (2023)

$$\mathcal{L}_{\text{gBCE}}^{\beta} = -\frac{1}{|I_k^-| + 1} \left(\log(\sigma^{\beta}(s_{i^+})) + \sum_{i \in I_k^-} \log(1 - \sigma(s_i)) \right).$$

Table 2: Effects of model architecture and negative sampling on NDCG@10, for the MovieLens-1M (ML-1M) and Steam datasets. * denotes a significant change ($pvalue < 0.05$) in NDCG@10 caused by negative sampling (comparing horizontally) or model architecture (comparing vertically).

Dataset	Negative sampling and loss function → Architecture ↓	1 negative per positive; BCE Loss (as SASRec)	No negative sampling; Softmax Loss (as BERT4Rec)	Negative sampling and loss effect
ML-1M	SASRec	0.131	0.169	+29.0%*
	BERT4Rec	0.123	0.161	+30.8%*
	Architecture effect	-6.1%	-4.7%	
Steam	SASRec	0.0581	0.0721	+24.1%*
	BERT4Rec	0.0513	0.0746	+45.4%*
	Architecture effect	-11.7%*	+3.4%*	

Вопросы

Индустрия

- онлайн АБ тесты с метриками
- Не только выбить топ метрику, а придумать как все завести в проде

Wide & Deep Learning for Recommender Systems (2016)

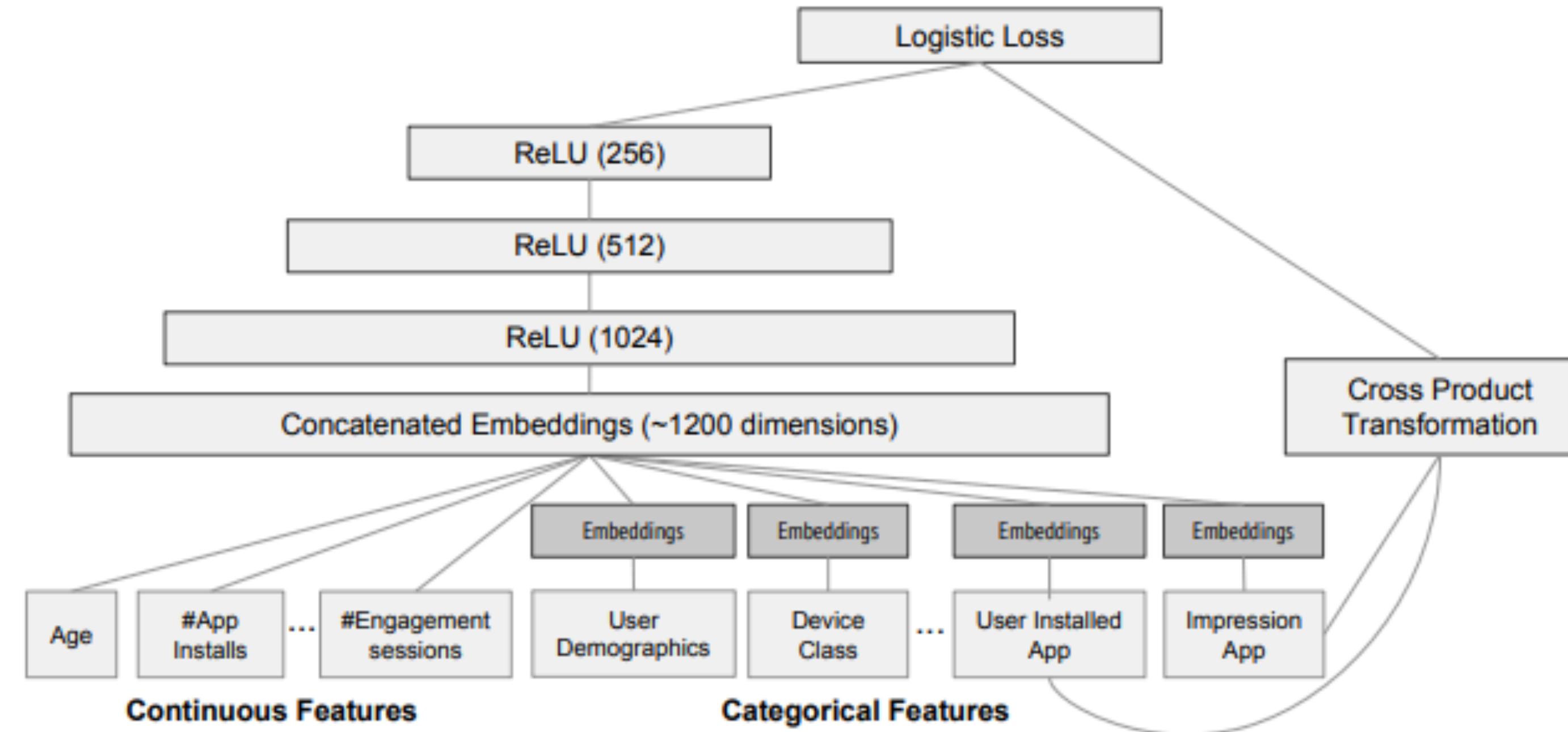


Figure 4: Wide & Deep model structure for apps recommendation.

Wide & Deep Learning for Recommender Systems

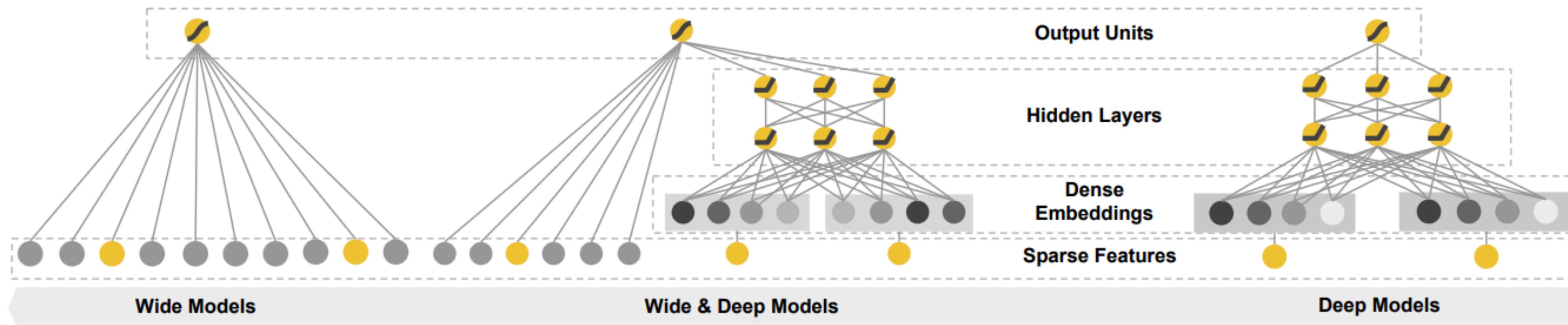
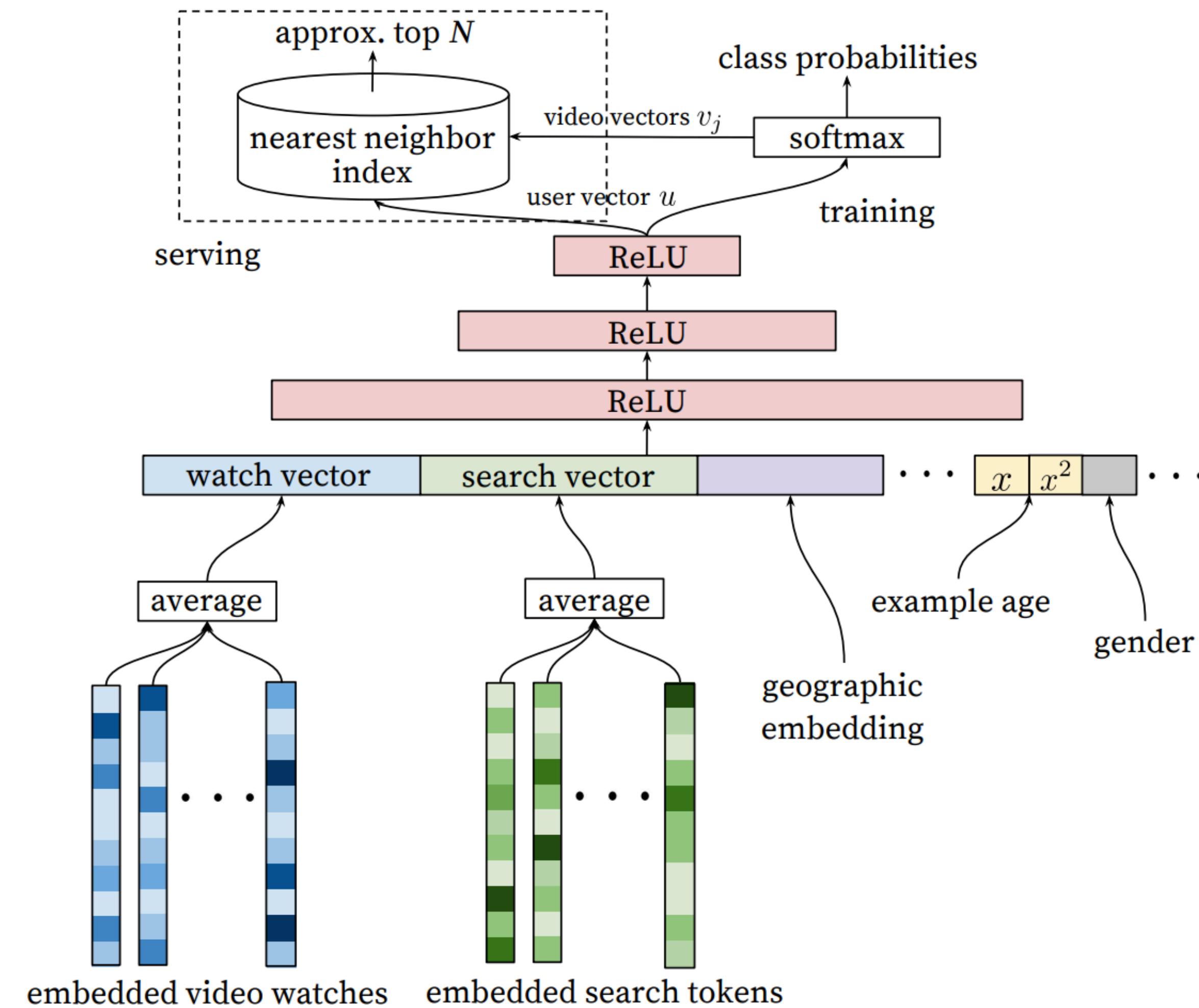


Figure 1: The spectrum of Wide & Deep models.

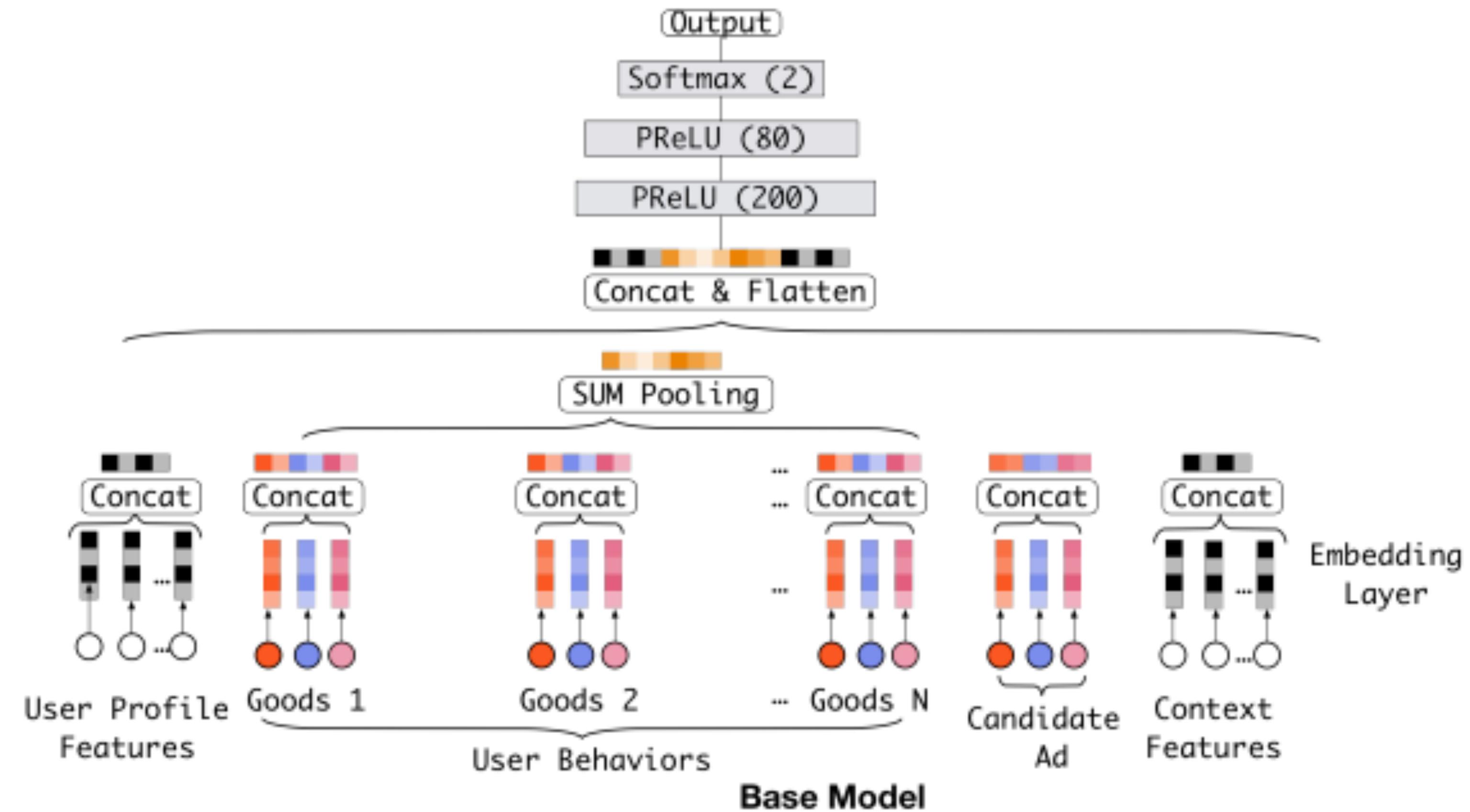
Wide & Deep Learning for Recommender Systems

- Memorization - запоминание частых совместных сигналов; получаем более актуальные рекомендации
- Generalization - исследование новых комбинаций фичей и наблюдений; получаем более разнообразные и новые рекомендации

Deep Neural Networks for YouTube Recommendations (2016)

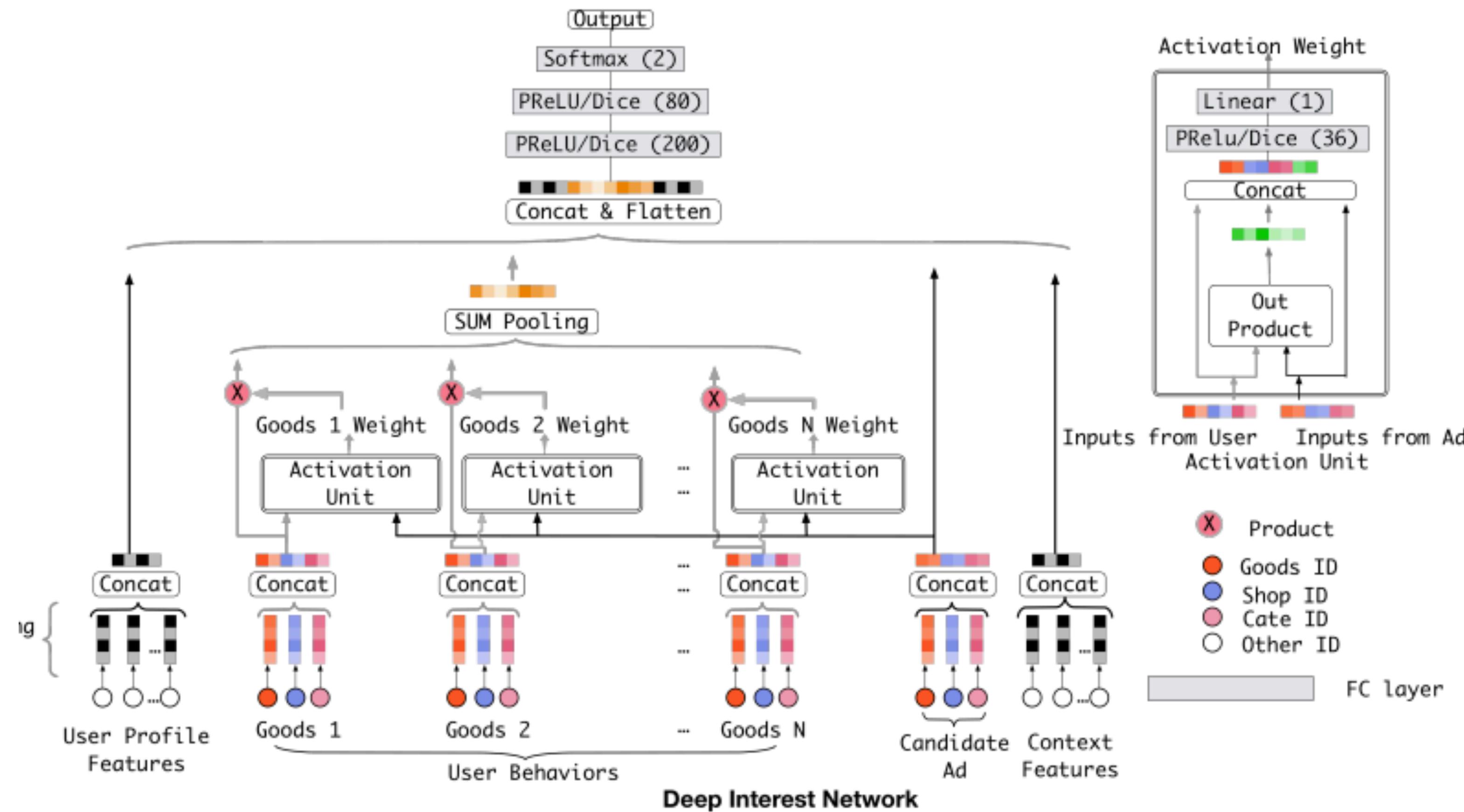


Deep Interest Network for Click-Through Rate Prediction (2017)



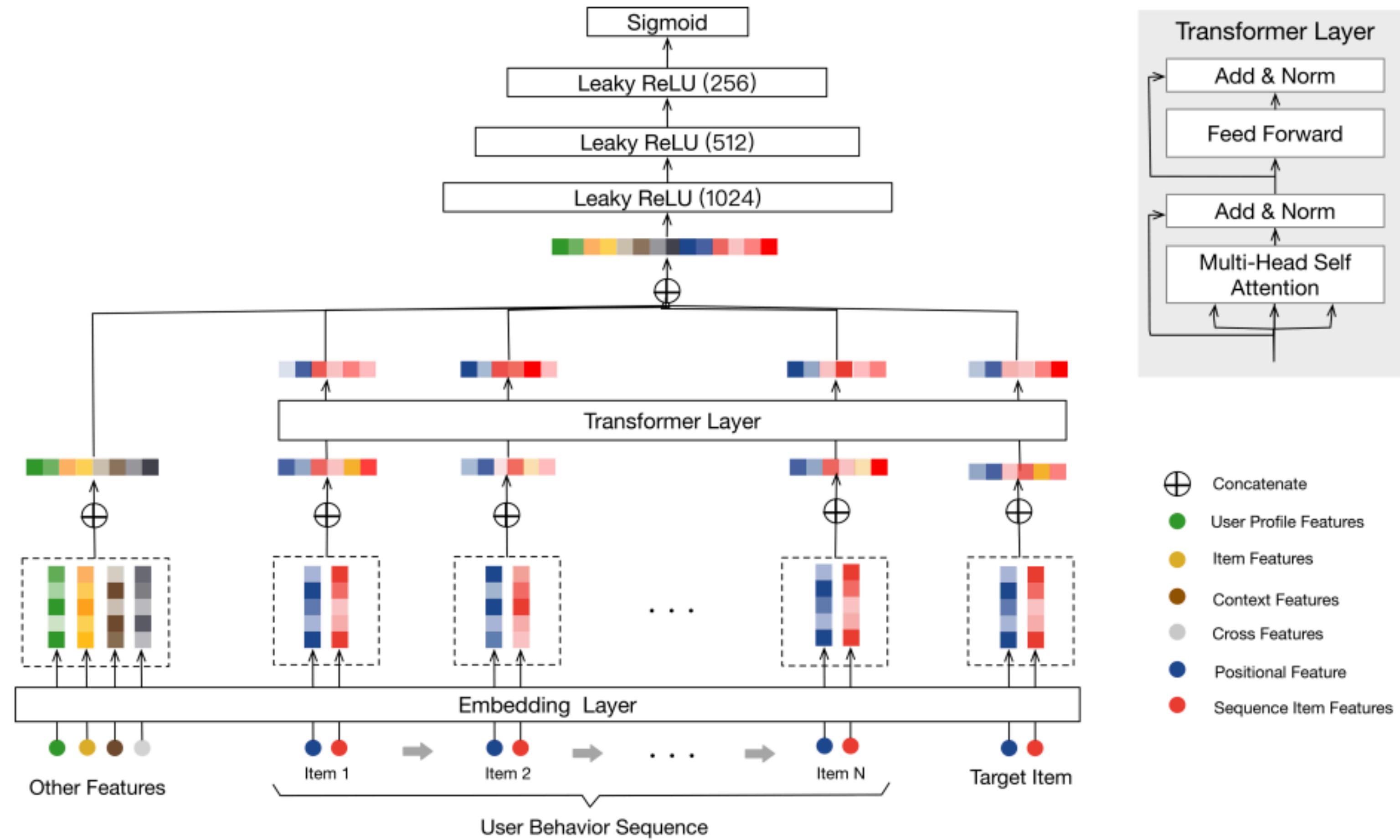
Deep Interest Network for Click-Through Rate Prediction

Deep Interest Network for Click-Through Rate Prediction



Deep Interest Network for Click-Through Rate Prediction

Behavior Sequence Transformer for E-commerce Recommendation in Alibaba (2019)



Behavior Sequence Transformer for E-commerce Recommendation in Alibaba

PinnerFormer: Sequence Modeling for User Representation at Pinterest (2022)

PINNERFORMER: Sequence Modeling for User Representation at Pinterest

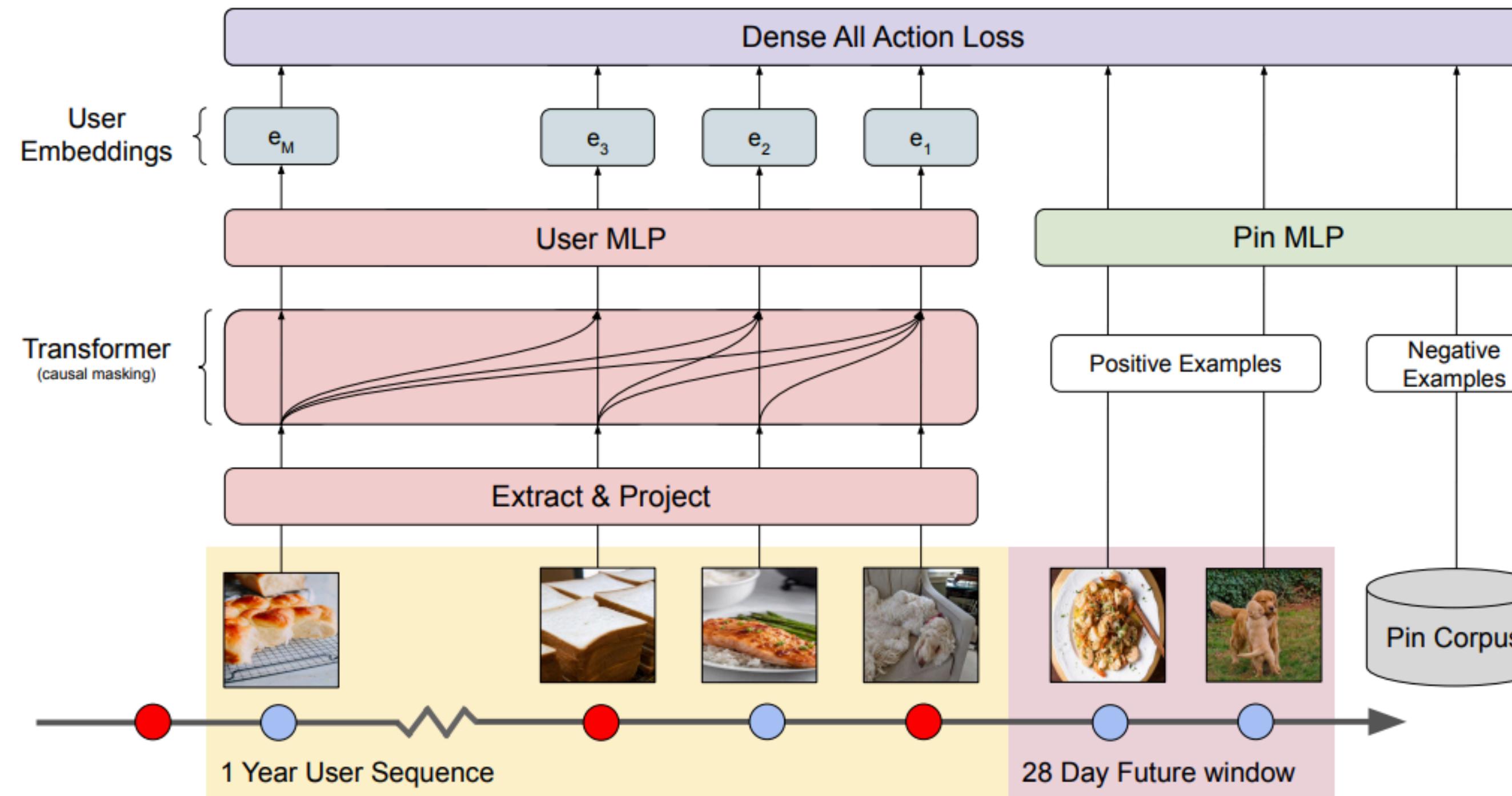
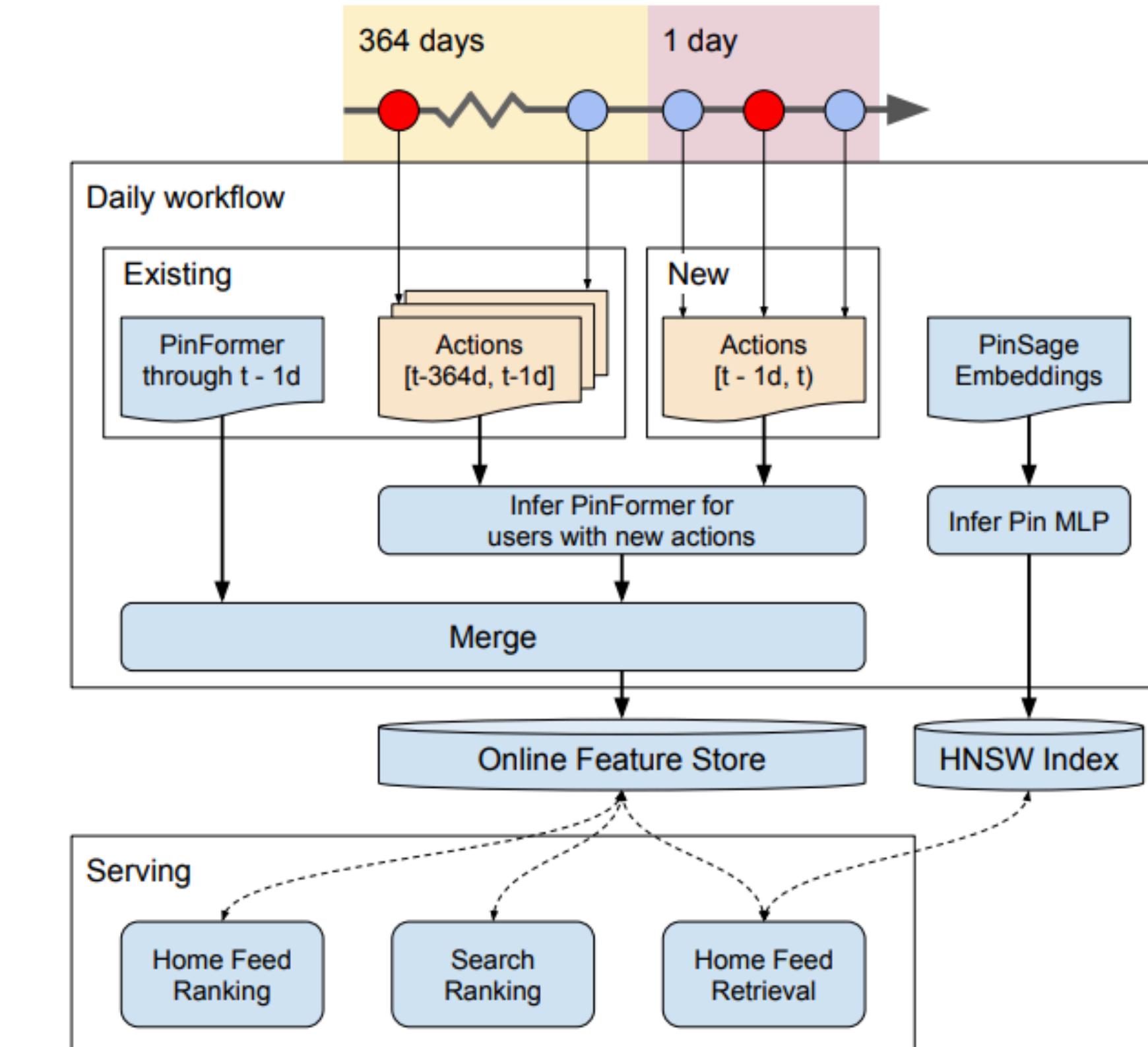
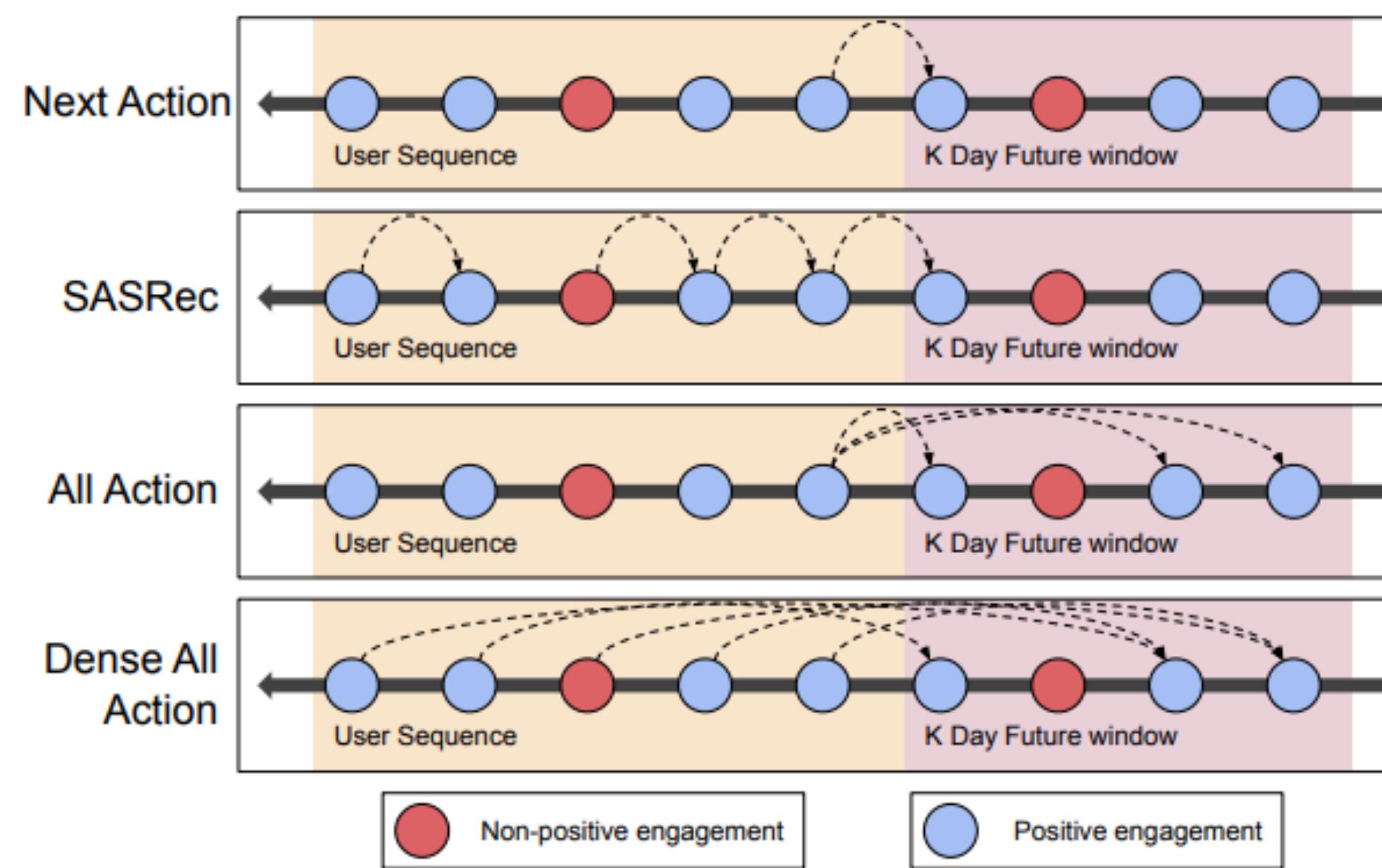


Figure 1: Overview of PINNERFORMER architecture. Features are passed through a transformer with causal masking, and embeddings are returned at every time step. Note that the training window (28d above) exceeds our future evaluation objective window (14d)

PinnerFormer: Sequence Modeling for User Representation at Pinterest



PinnerFormer: Sequence Modeling for User Representation at Pinterest

PinnerFormer: Sequence Modeling for User Representation at Pinterest

lower bound for stability. If we let $s(u, p) = \langle u, p \rangle / \tau$, a sampled softmax loss without sample probability correction would be defined as follows:

$$\mathcal{L}(u_i, p_i) = -\log \left(\frac{e^{s(u_i, p_i)}}{e^{s(u_i, p_i)} + \sum_{j=1}^N e^{s(u_i, n_j)}} \right) \quad (1)$$

When negatives are not uniformly distributed, A correction term $Q_i(v) = P(\text{Pin } v \text{ in batch} \mid \text{User } U_i \text{ in batch})$ should be applied to correct for sampling bias, where v may be a positive or negative example. The softmax loss with sample probability correction for a single pair is then defined as follows:

$$\mathcal{L}(u_i, p_i) = -\log \left(\frac{e^{s(u_i, u_i) - \log(Q_i(p_i))}}{e^{s(u_i, u_i) - \log(Q_i(p_i))} + \sum_{j=1}^N e^{s(u_i, n_j) - \log(Q_i(n_j))}} \right) \quad (2)$$

PinnerFormer: Sequence Modeling for User Representation at Pinterest

Лосс и негативы

- Сэмлирование негативов - in batch негативы + рандомные из коллекции
- In batch негативы не "случайны" - добавляем logQ correction, отвечающий за "популярность" пары (user, item)

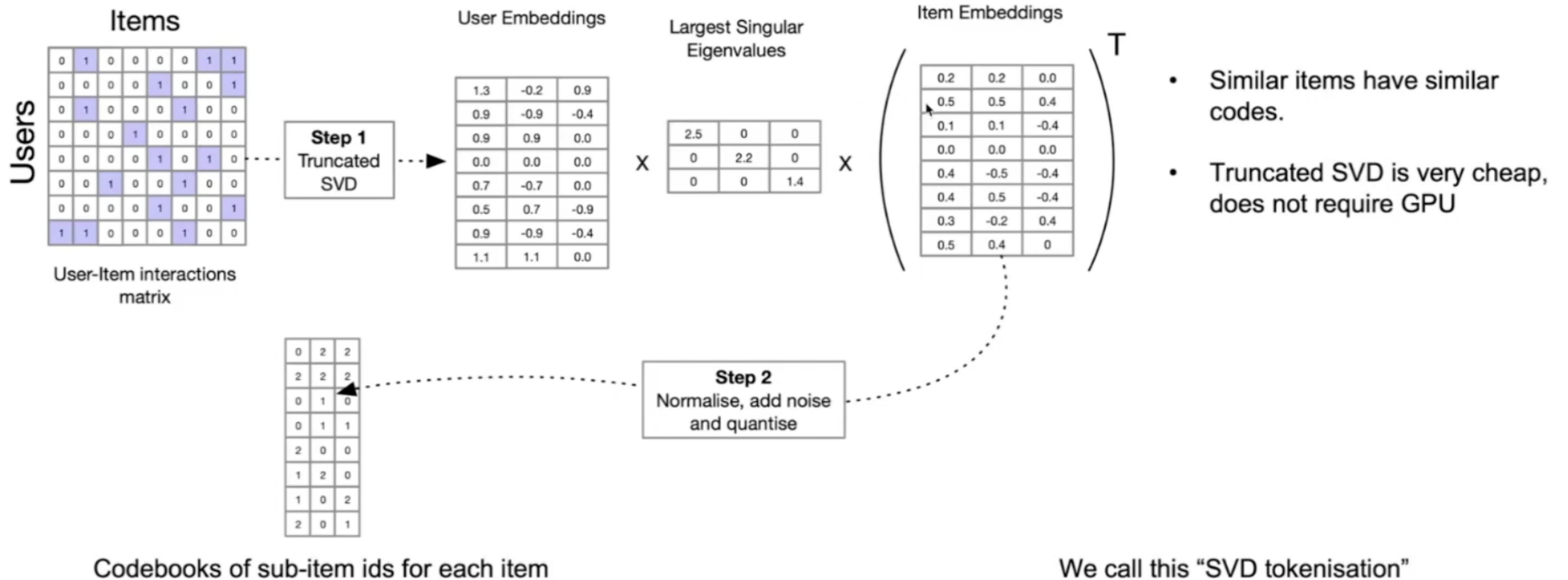
Вопросы

Проблемы

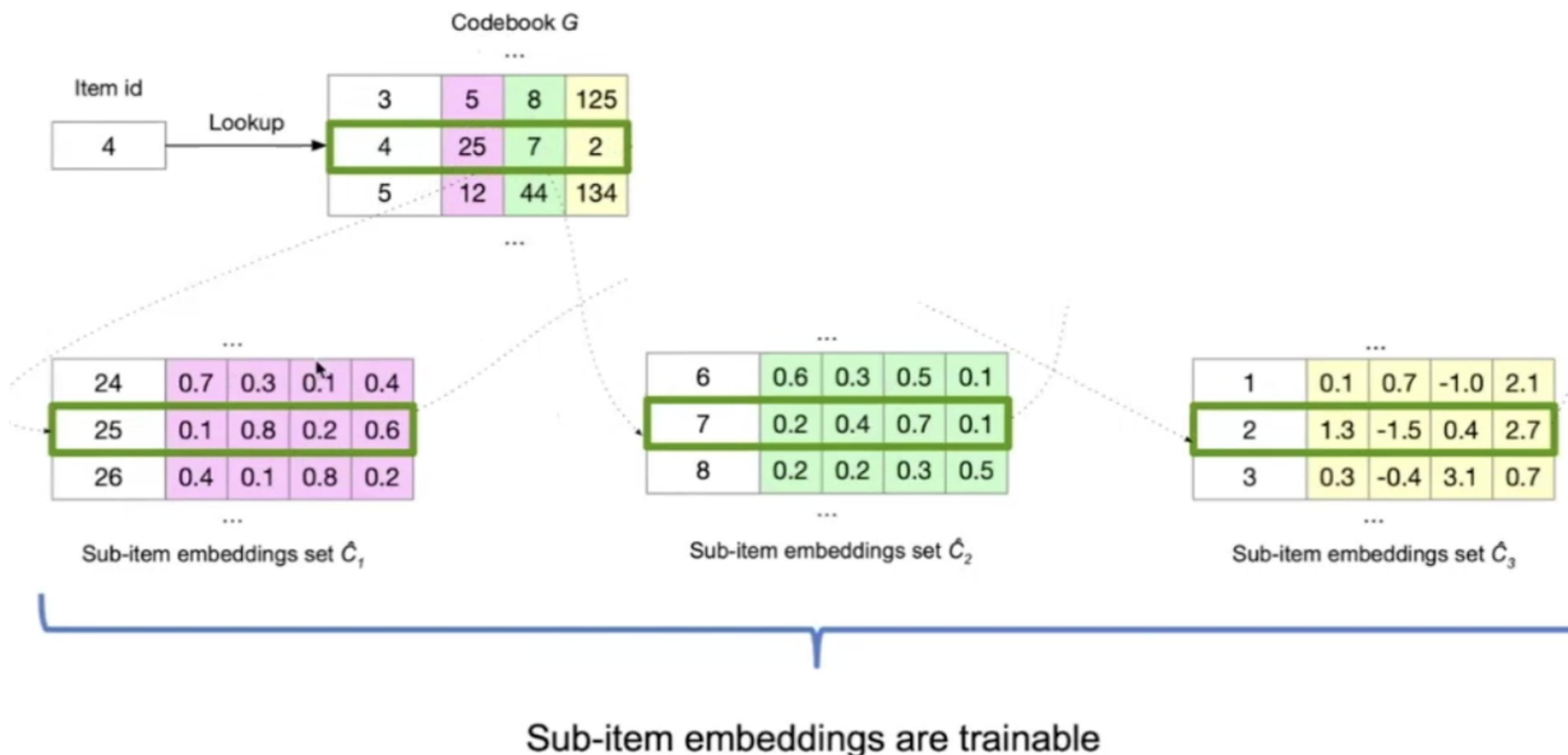
Размер словаря embedding

- в нlr моделях размер словаря 30-250k токенов
 - bert - 30k
 - GPT-3 - 50k
 - Llama 2 - 32k
- количество уникальных itemid может быть более 100млн id
- что делать?

RecJPQ: Training Large-Catalogue Sequential Recommenders (2023)



RecJPQ: Training Large-Catalogue Sequential Recommenders



Product Quantisation (PQ):

1. Instead of storing embeddings of items, we store embeddings of sub-items, (equivalent of tokens), which we have grouped into sets
2. Store code (list of sub-item ids) for items, instead of storing embeddings

Semantic IDs

- Идея: уйти от рандомных id в сторону векторных представлений товара на основе контента/смысловых характеристик
- Поговорим подробнее на следующих лекциях...

Вопросы

А что делают у нас?

Yandex

| Кодируем тип события через обучаемый вектор

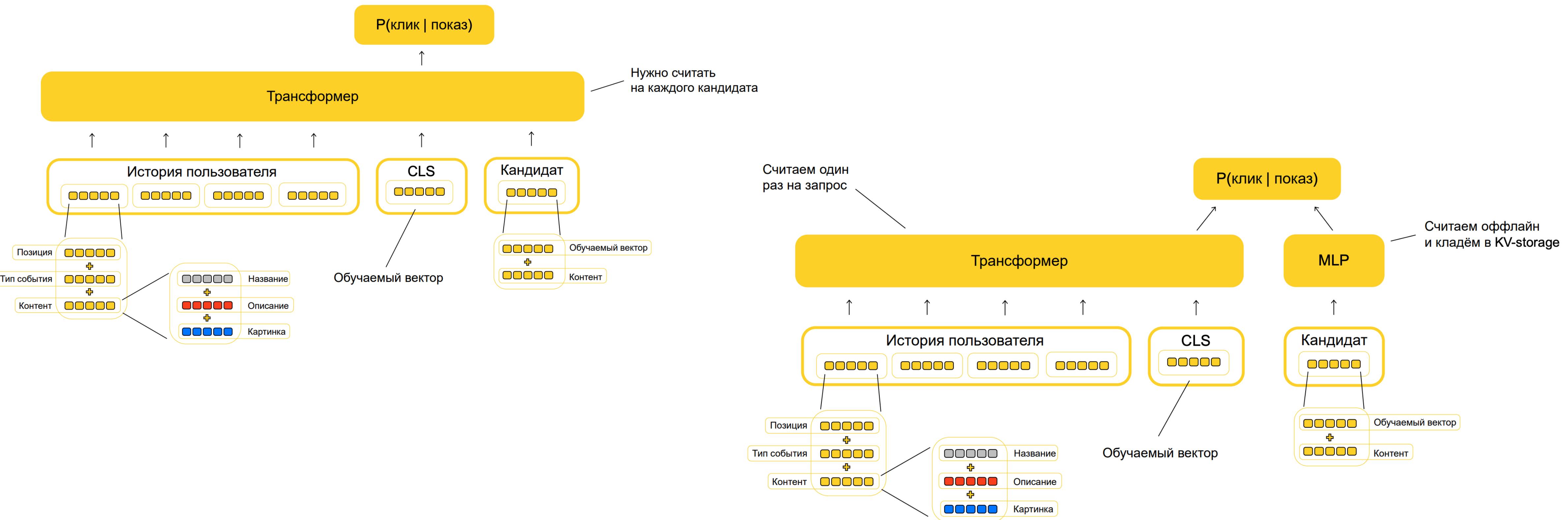
| Кодируем позиции в обратном порядке

- › Обучаемые абсолютные позиционные эмбеддинги
- › Относительные позиции и таймстемпы тоже можно использовать
- › При наличии каузальной маски позиции не всегда нужны
- › Оффлайн-модели менее чувствительны к ПОЗИЦИЯМ



Тайны трансформерной персонализации

Yandex

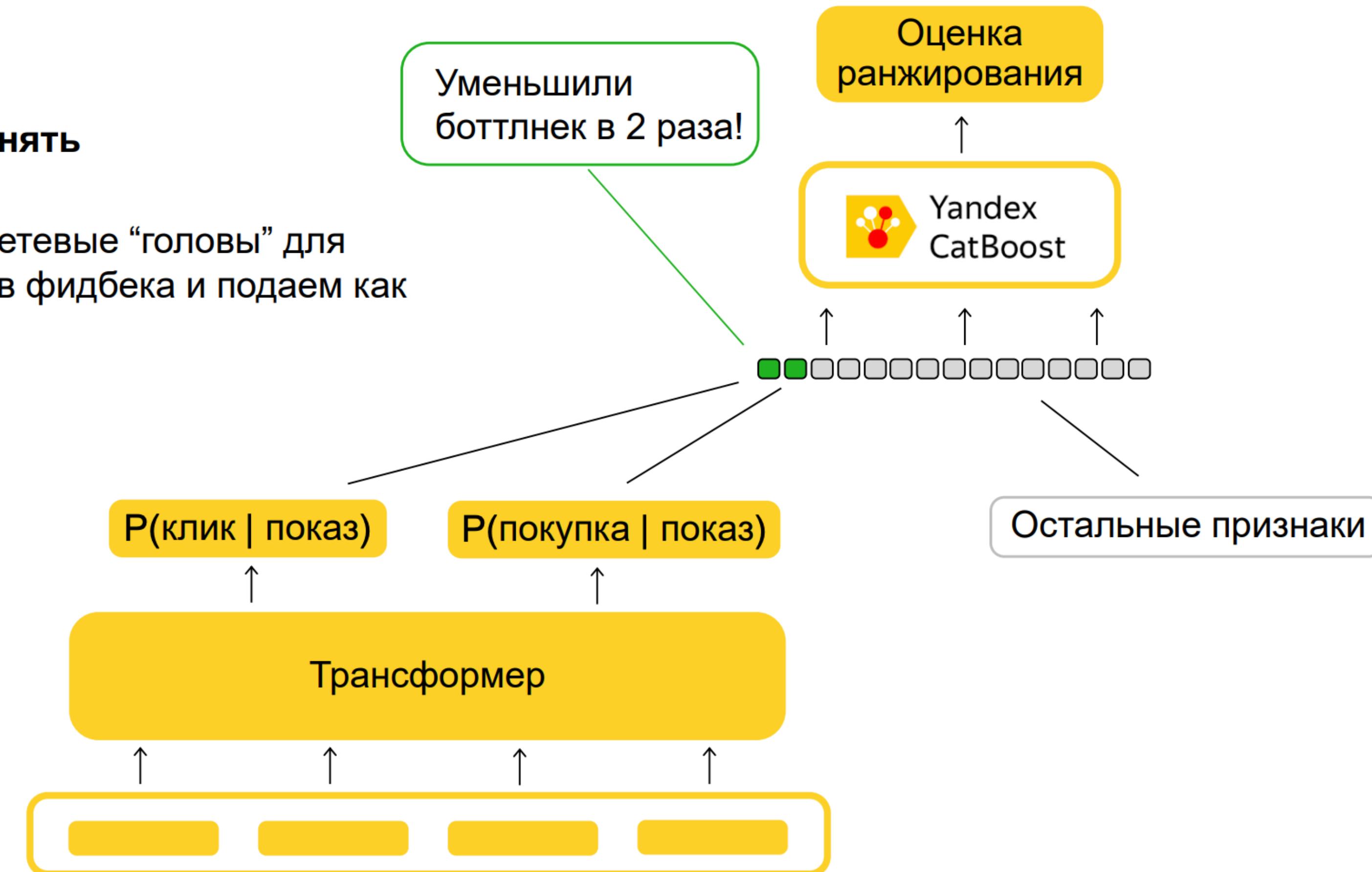


Тайны трансформерной персонализации

Yandex

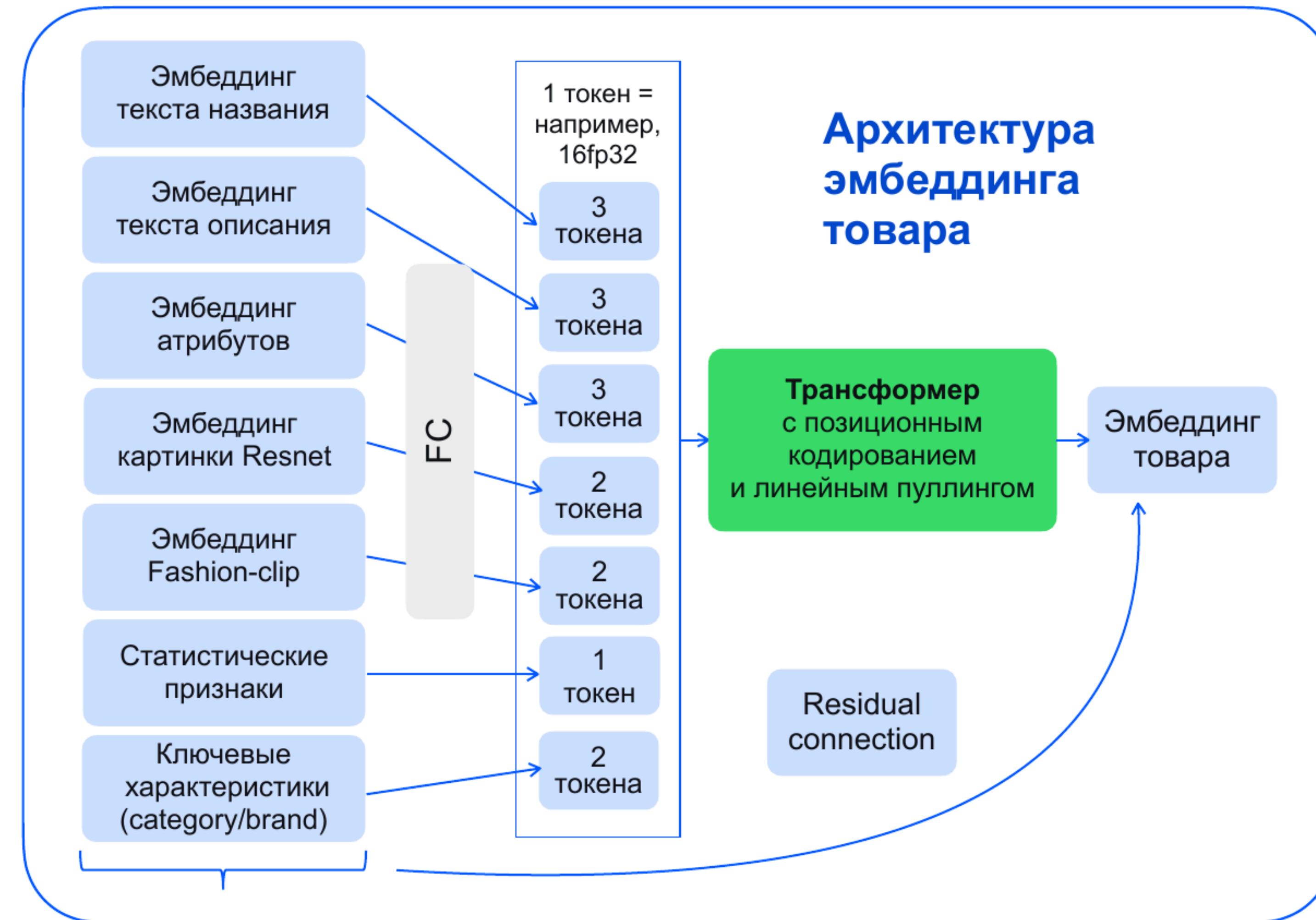
Сигналы можно разъединять

- › Делаем отдельные нейросетевые “головы” для предсказания разных типов фидбека и подаем как отдельные признаки



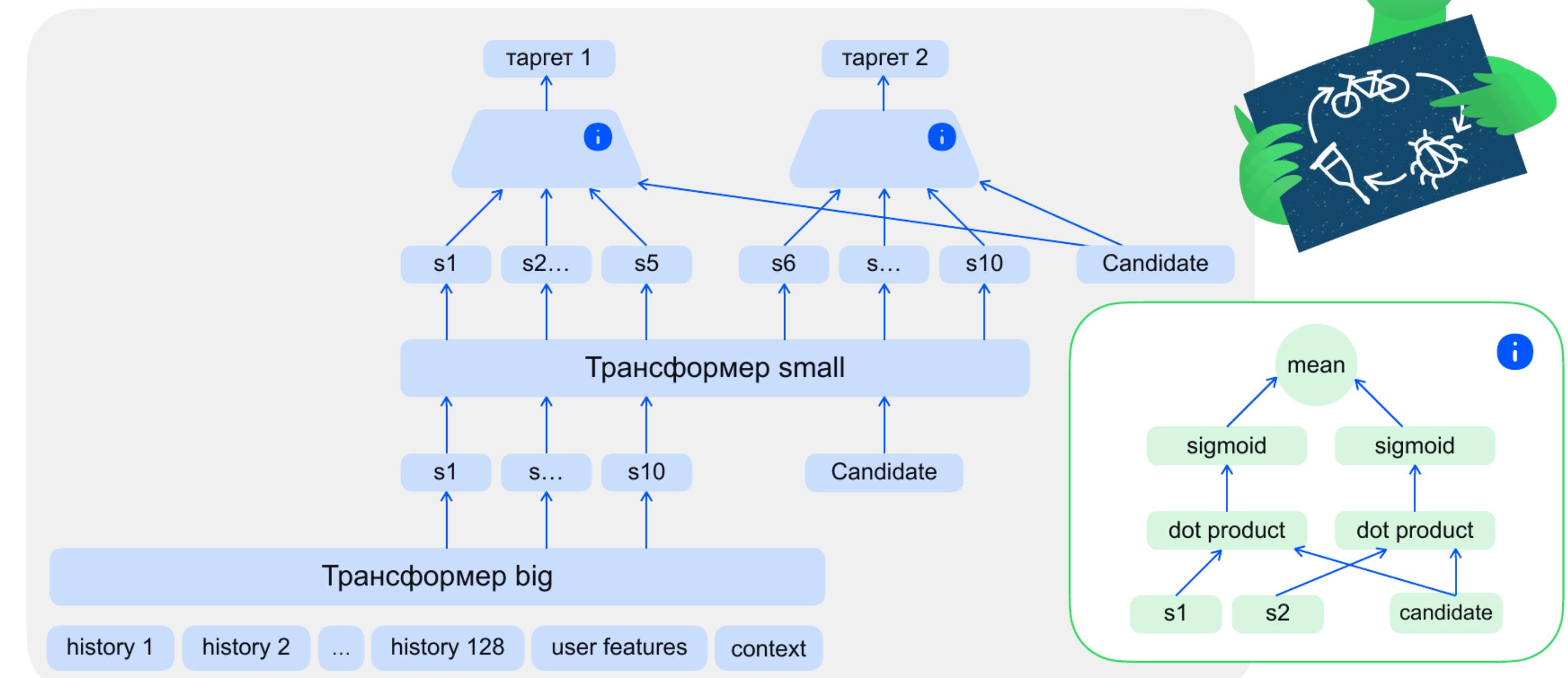
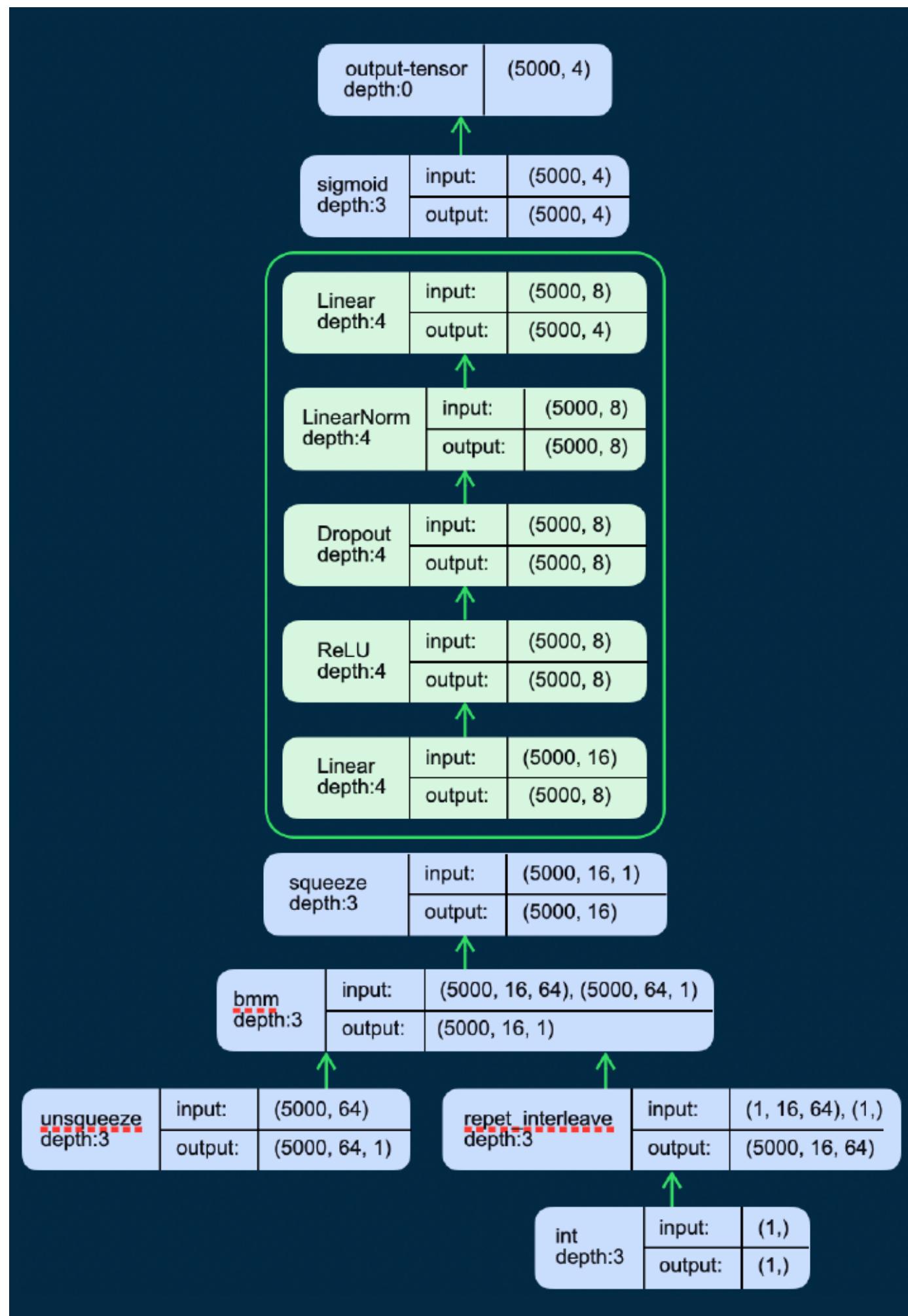
Тайны трансформерной персонализации

Ozon



Нейросети в рекомендациях: от идеи до продакшна

Ozon



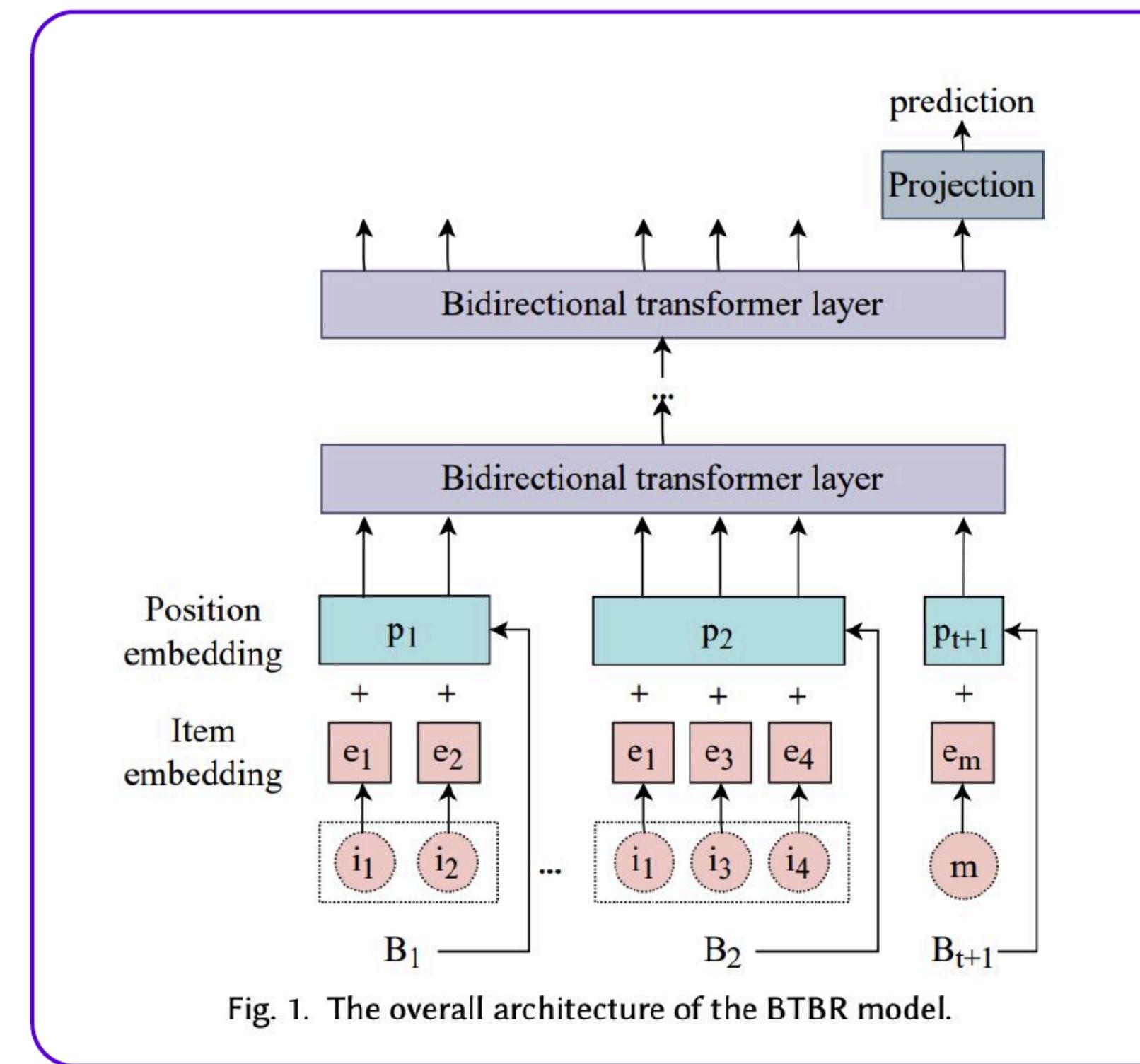
Нейросети в рекомендациях: от идеи до продакшна

Wildberries

Next basket prediction

Ai^{Conf}
wildberries

- BERT рассматривает товары в истории как отдельные транзакции — не E-commerce-сценарий
- Маскируя товары, выучиваем межアイテムные связи, но теряем связи между корзинами
- Давайте маскировать корзины!
- И адаптировать под них позиционный эмбеддинг



<https://arxiv.org/pdf/2308.01308>

WildBERT – развитие трансформерных архитектур для персонализации Wildberries

Wildberries

Позитивы по истории и квоты



→ Многие модели имеют свойство «зацикливаться» на историях, которые содержат в себе много однотипных товаров (например, одна частотная категория)

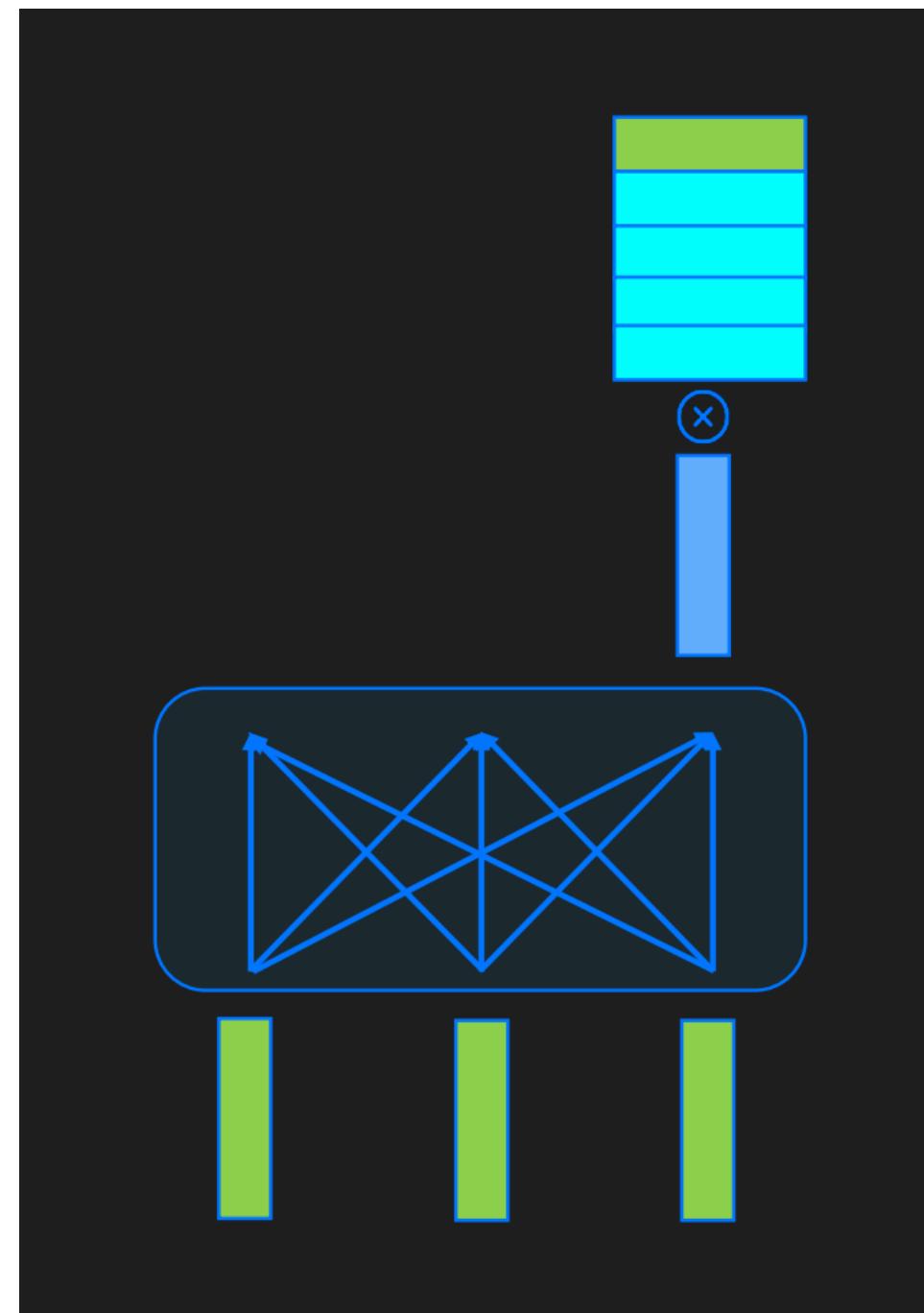
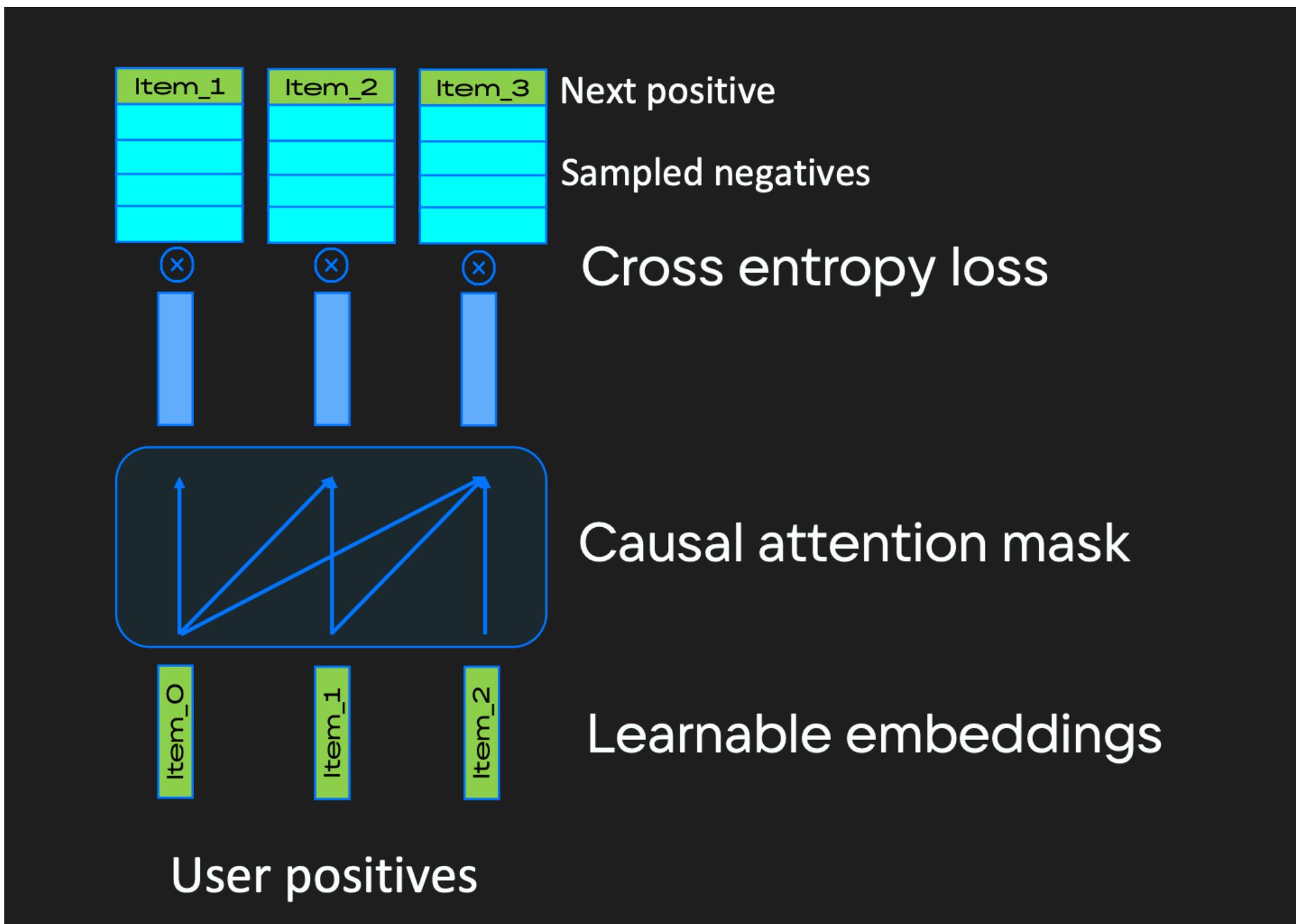
→ При этом это один из нормальных паттернов юзера

→ Набираем больше интересов из истории и не даем «скатиться» в одну категорию



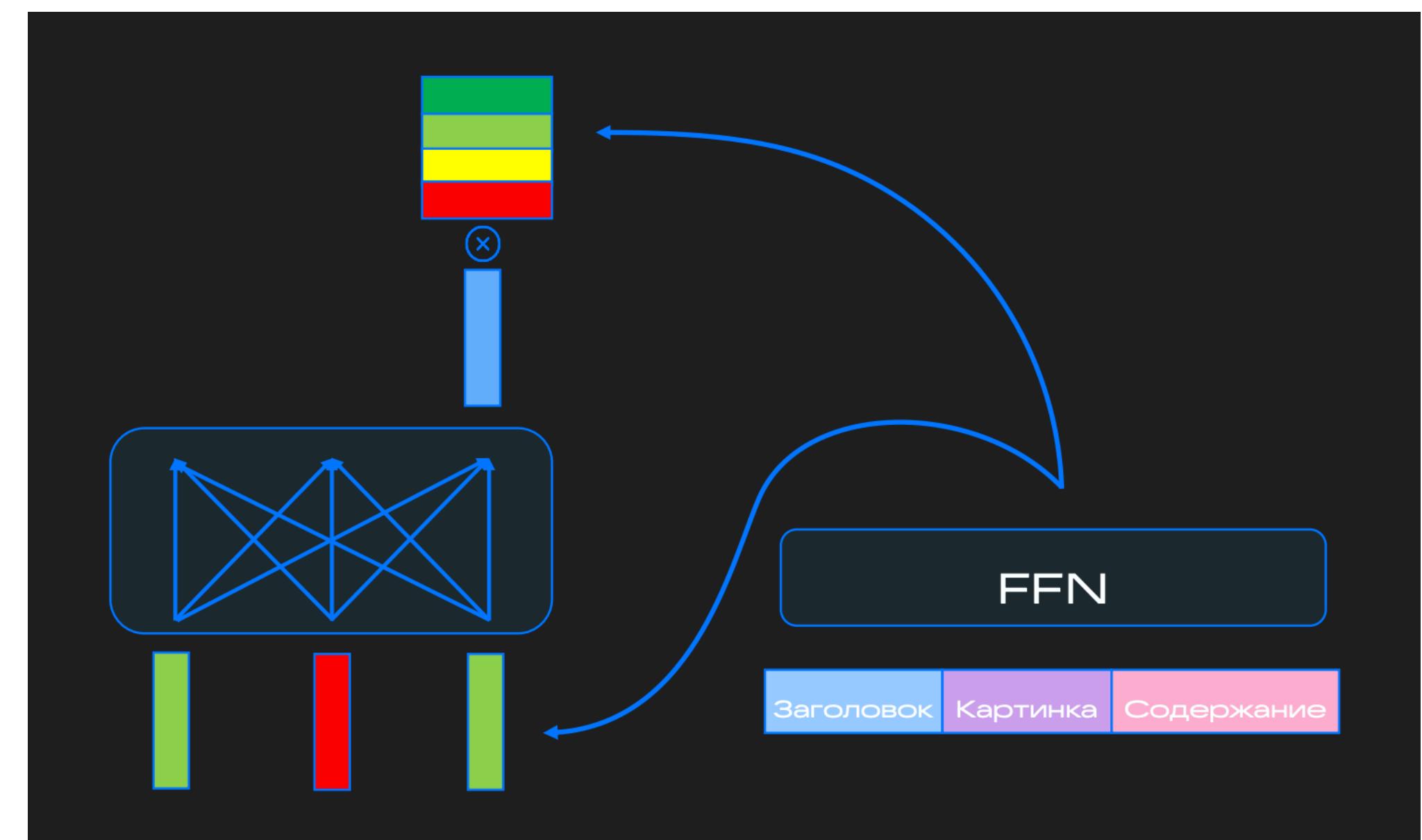
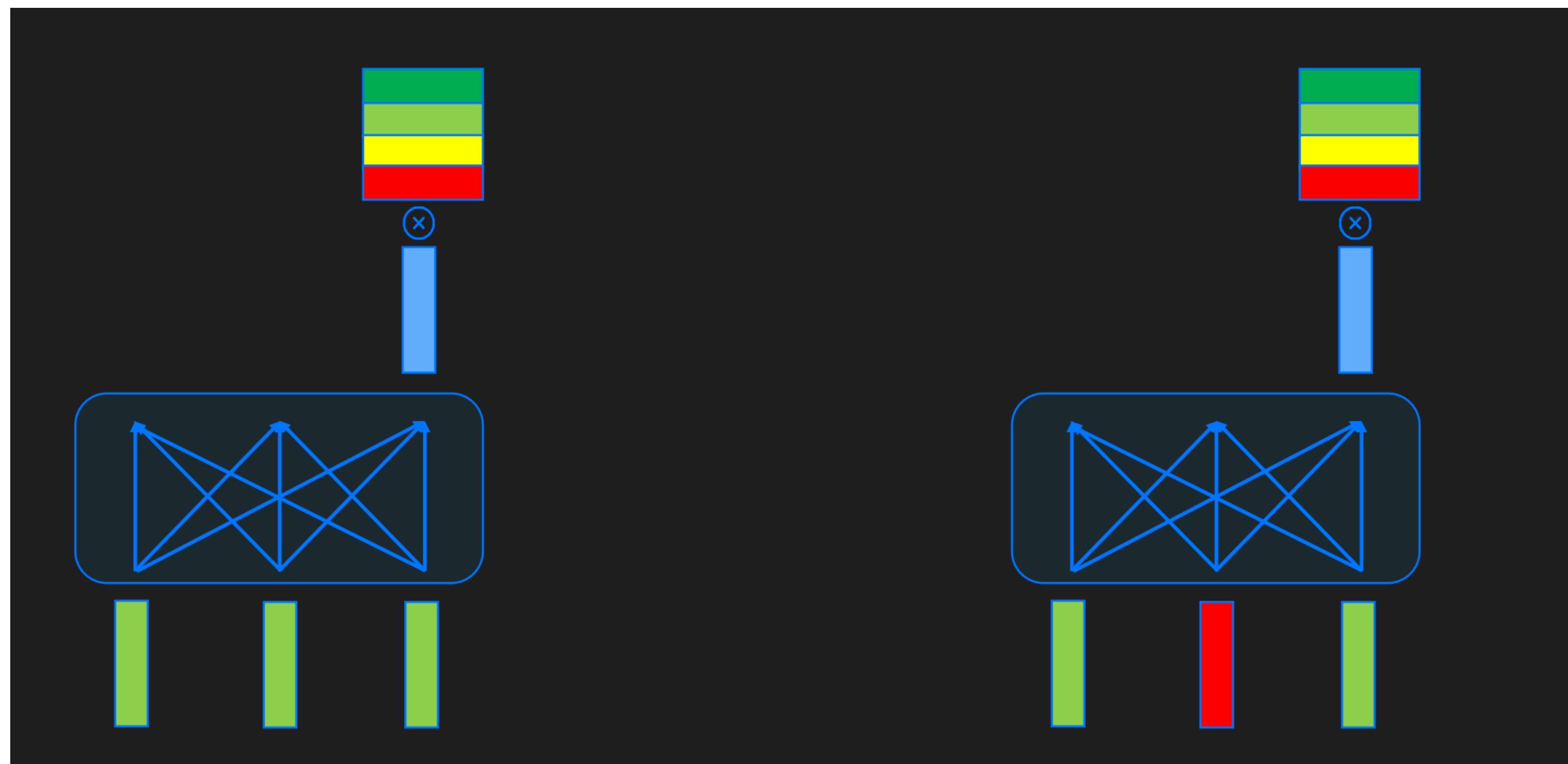
WildBERT – развитие трансформерных архитектур для персонализации Wildberries

VK



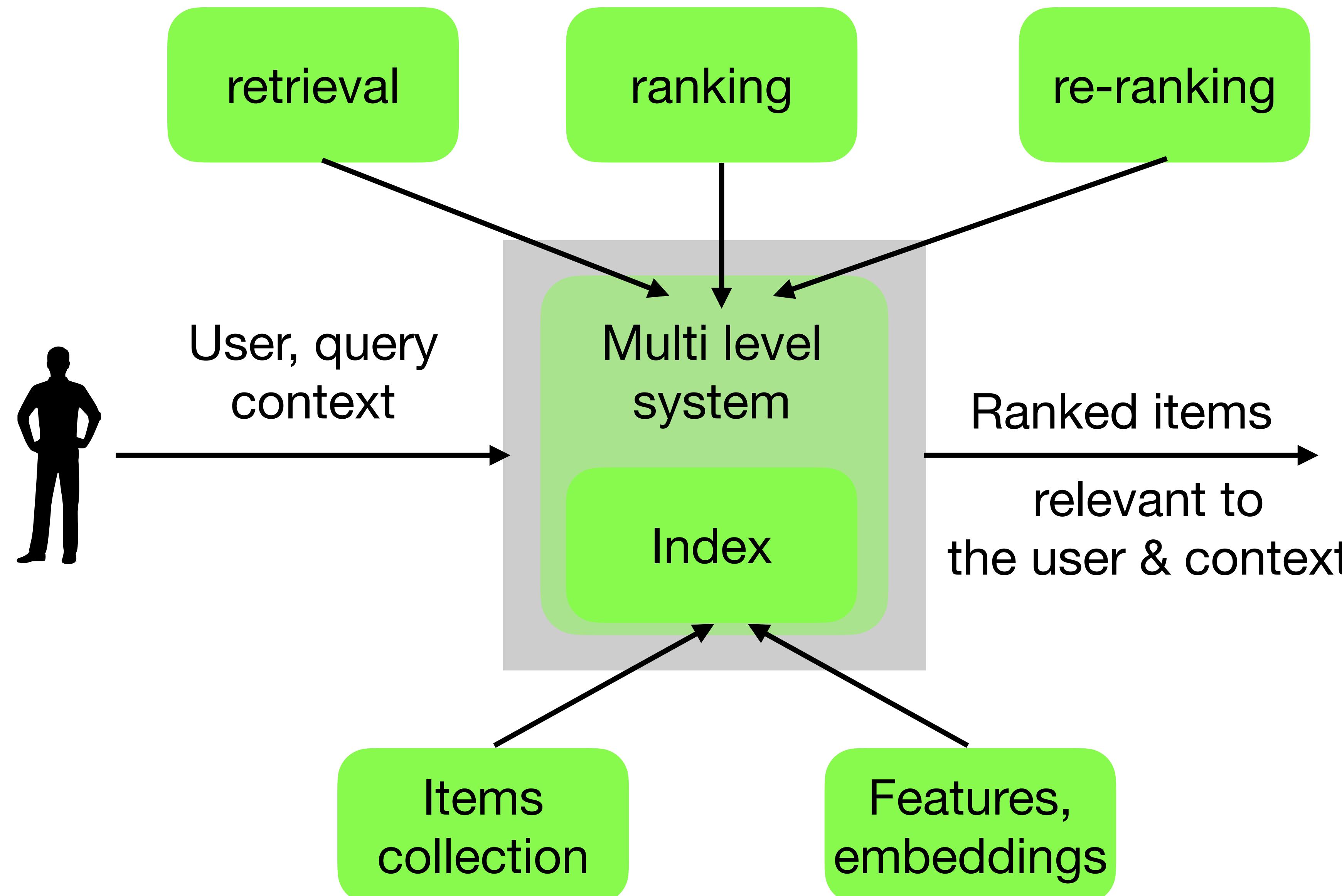
Трансформируем рекомендации

VK

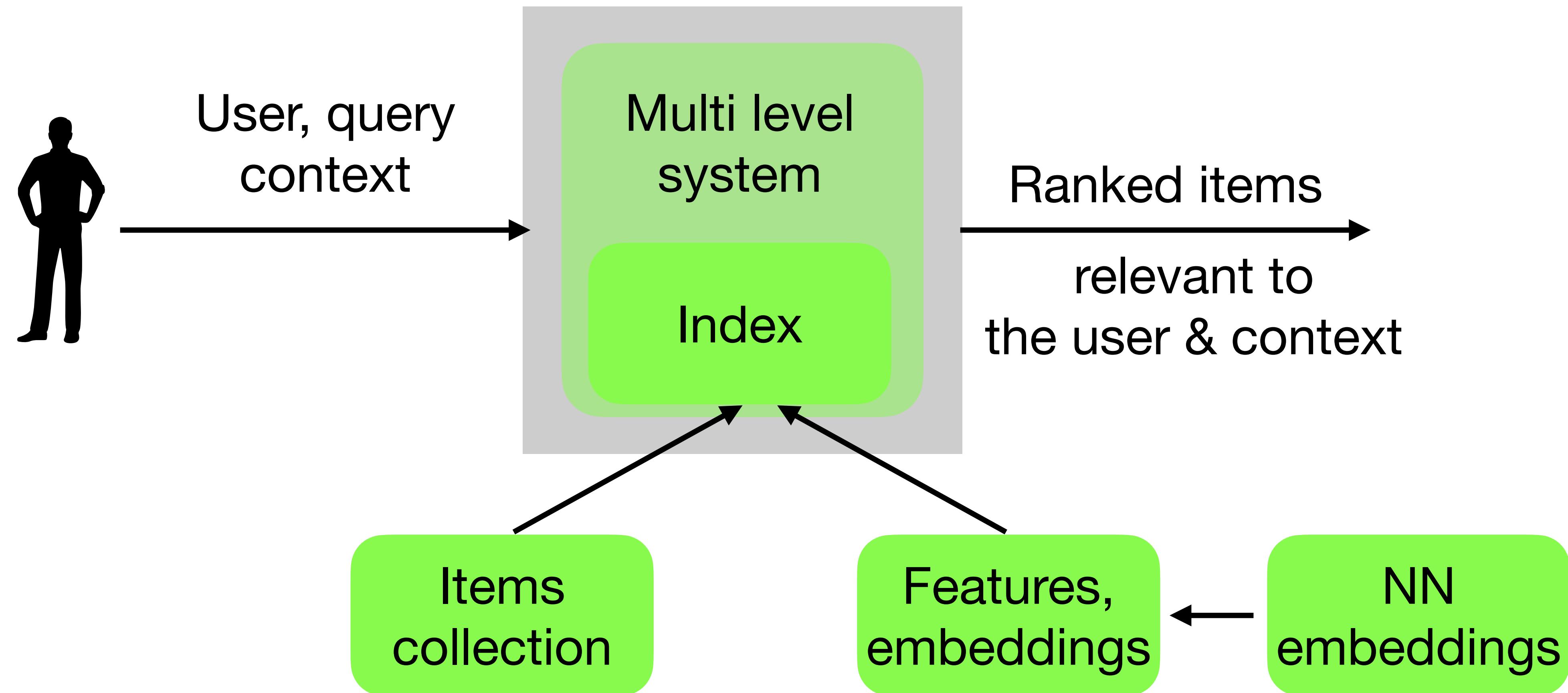


Трансформируем рекомендации

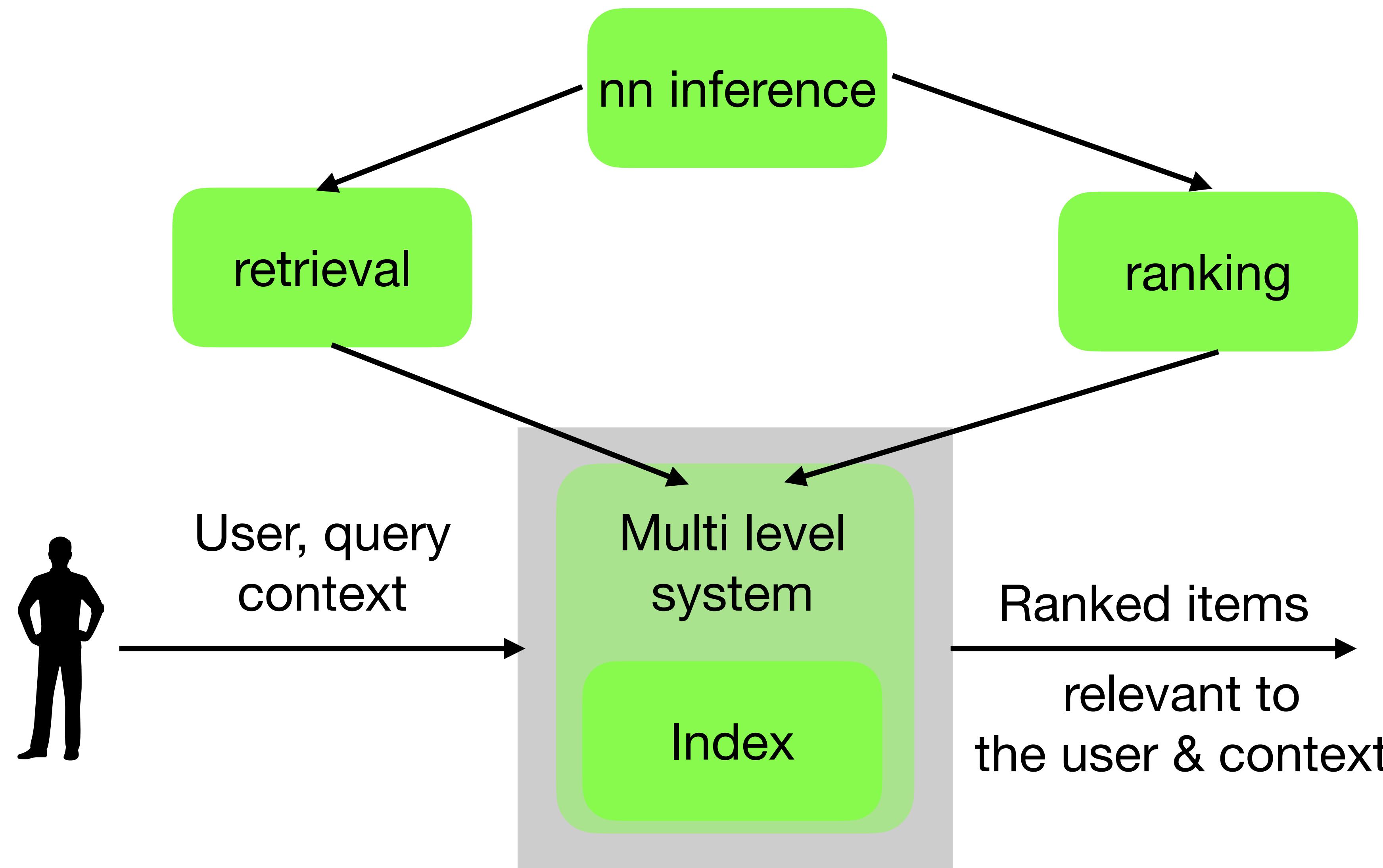
Многоуровневая система



Многоуровневая система



Многоуровневая система



Вопросы