Sequential recommender model

Student: Pavel Fakanov
Supervisor: Vadim Filippovich

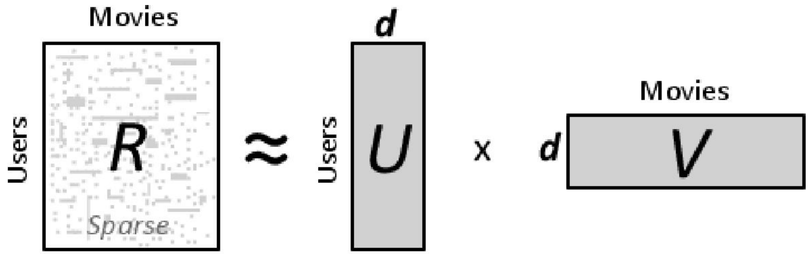
Programming project

pafakanov@edu.hse.ru

Sequential Recommender Models

Model and understand **sequential** user behaviours to increase the quality of recommendations

Recommender systems are extremely popular in our world



$$\hat{r}_{ui} = \langle \boldsymbol{p}_u, \boldsymbol{q}_i \rangle$$

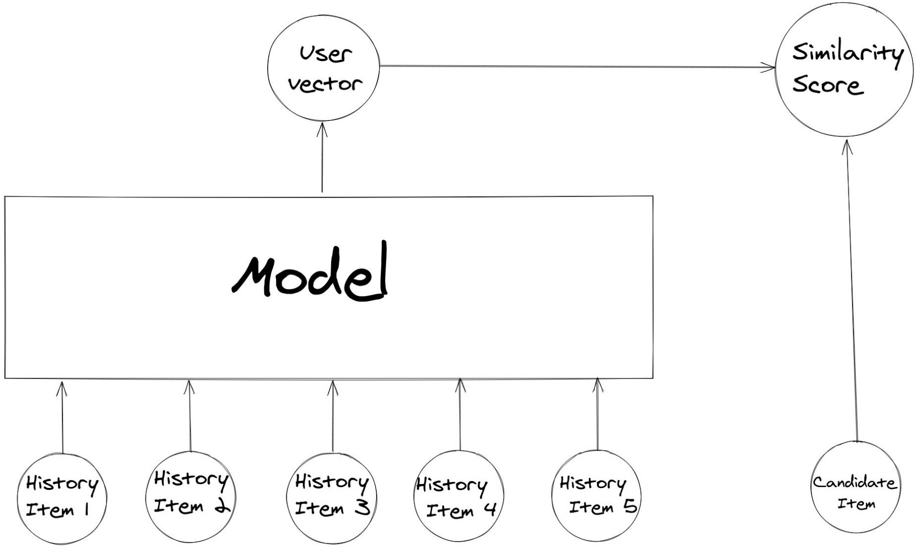
Traditional approaches ignore consideration of sequential dependencies between user interactions

Objective:

Increase the quality of recommendations using sequential models

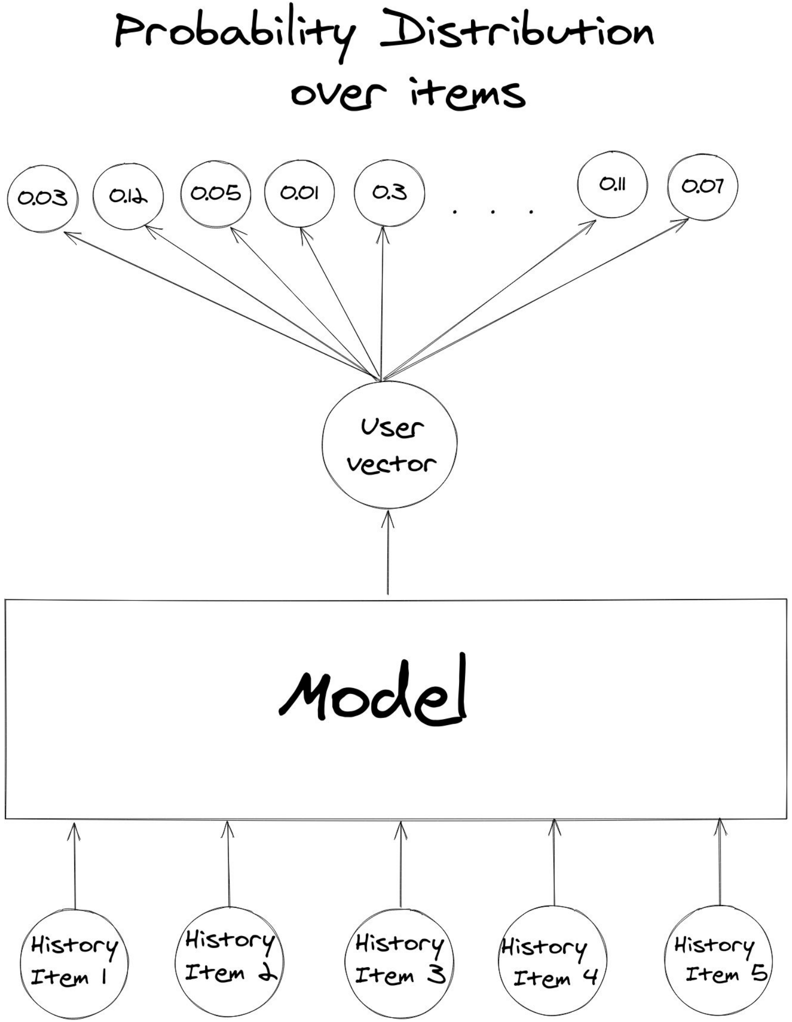
Tasks:

- Implement baseline LSTM-based model
- Implement BERT4Rec model
- Compare models performance on two different tasks



Relevance prediction

Learn a function F, to predict the probability of user liking the *given item* given user's interaction history

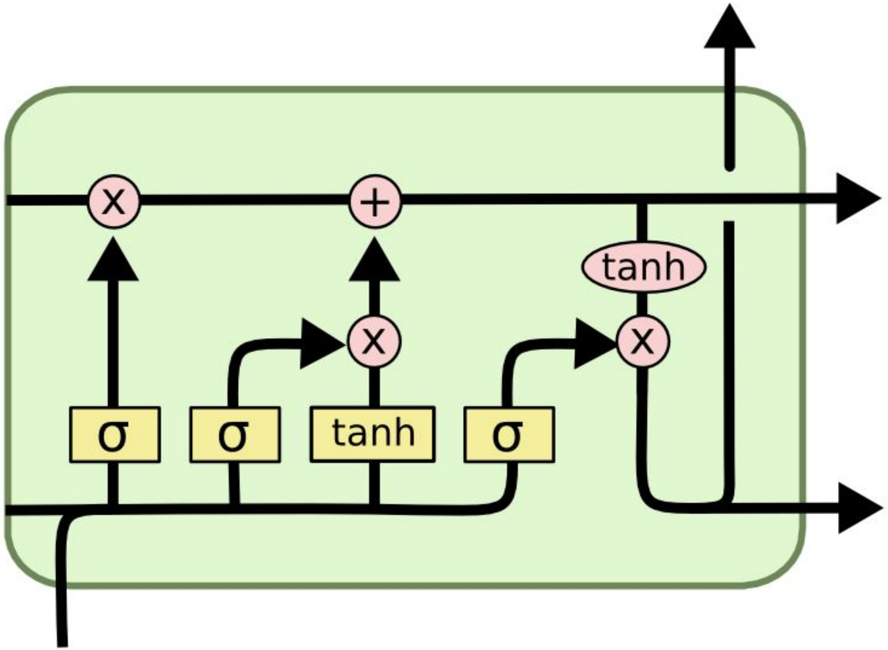


Next Item prediction

Model the probability of user liking the item over *all possible items* given user's interaction history

LSTM-based model

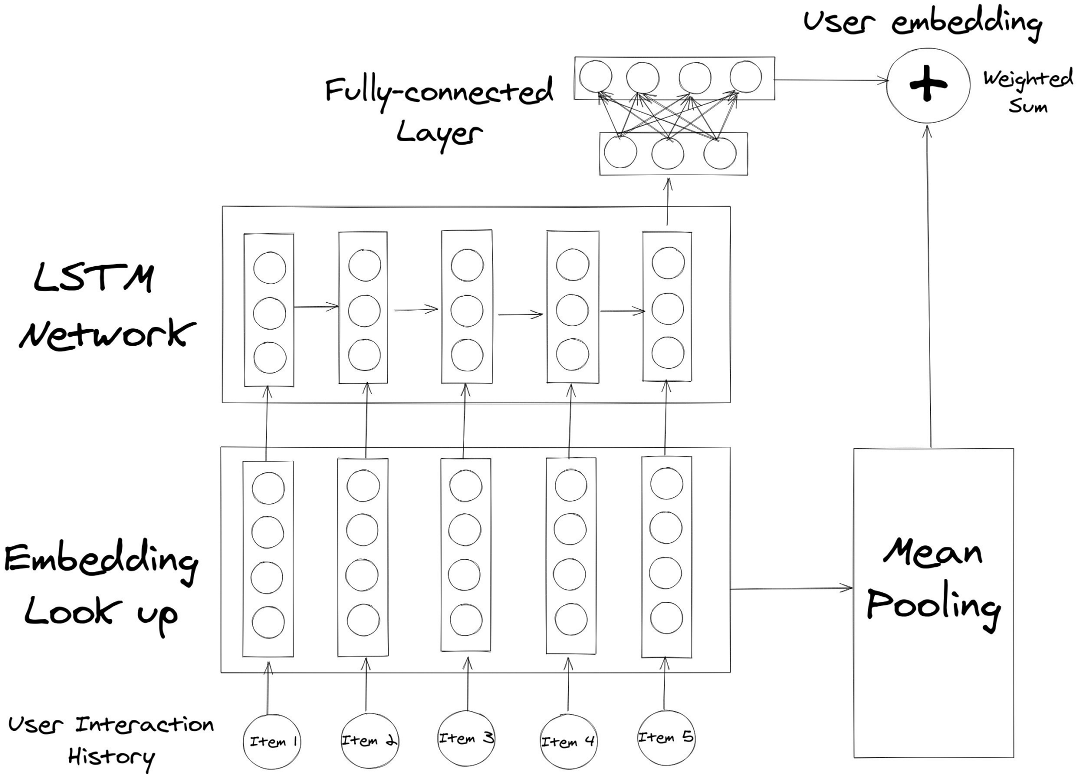
LSTM - recurrent neural network capable of learning order dependence in sequence prediction problems



LSTM-based model

User representation

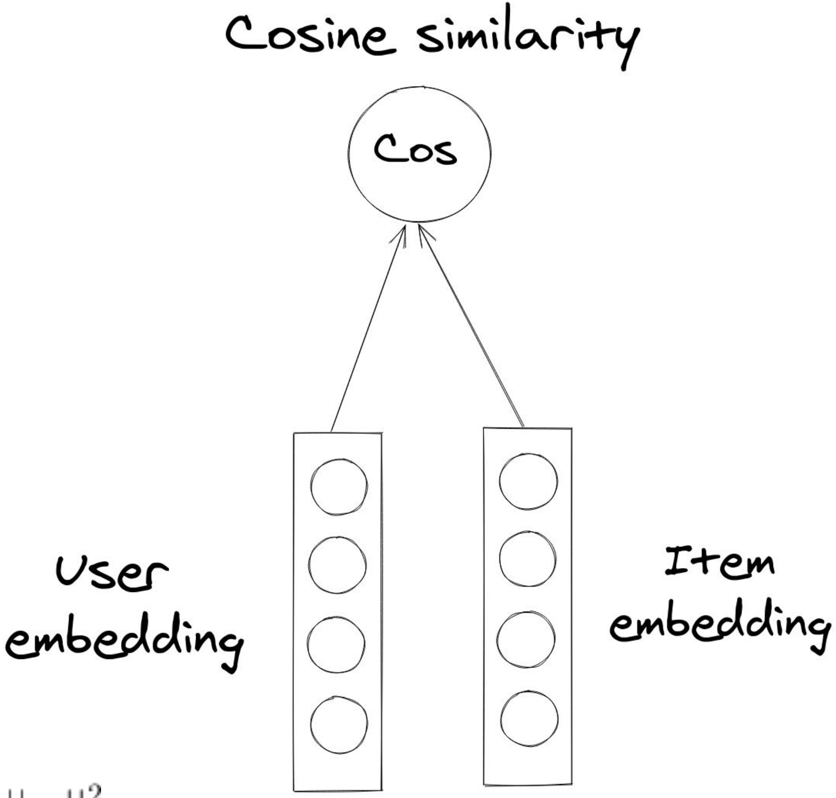
User embedding is a weighted sum of LSTM output and mean pooling output over input items' vectors



LSTM-based model

Relevance Prediction

- Randomly sample negative candidates
- Build user embedding
- Binary cross-entropy loss

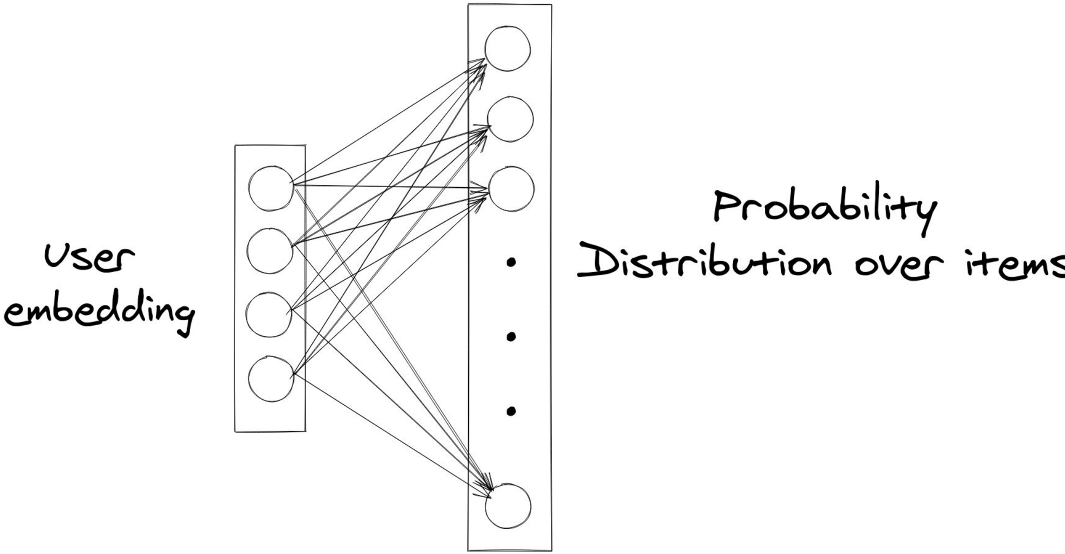


$$\mathcal{L} = -\frac{1}{|\mathcal{D}|} \sum_{x,y \in |D|} (y \log p(x) + (1-y) \log(1-p(x))) + \lambda \sum_{i=1}^{N} ||x_i||^2$$

LSTM-based model

Next Item Prediction

- Build user embedding
- Cross-entropy loss



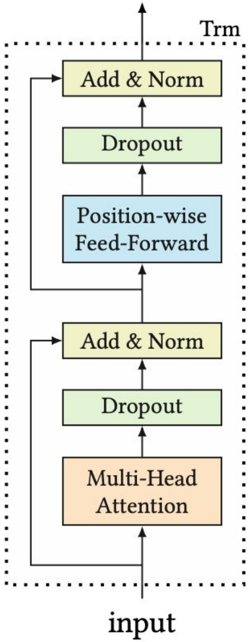
$$\mathcal{L} = -\frac{1}{|\mathcal{D}|}\sum_{x, v^x_{n_x+1}\in|D|}\sum_{v\in\mathcal{V}}[v = v^x_{n_x+1}]\log p(v|x) + \lambda\sum_{i=1}^{N}||x_i||^2$$

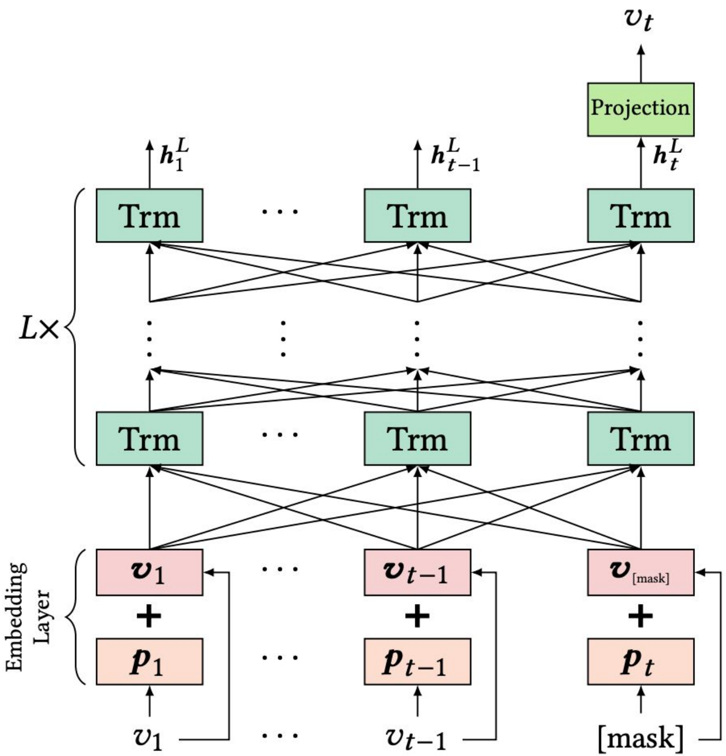
BERT4Rec

Model architecture

BERT4Rec is stacked by L bidirectional Transformer layers.

At each layer, it iteratively revises the representation of every position by exchanging information across all positions at the previous layer.

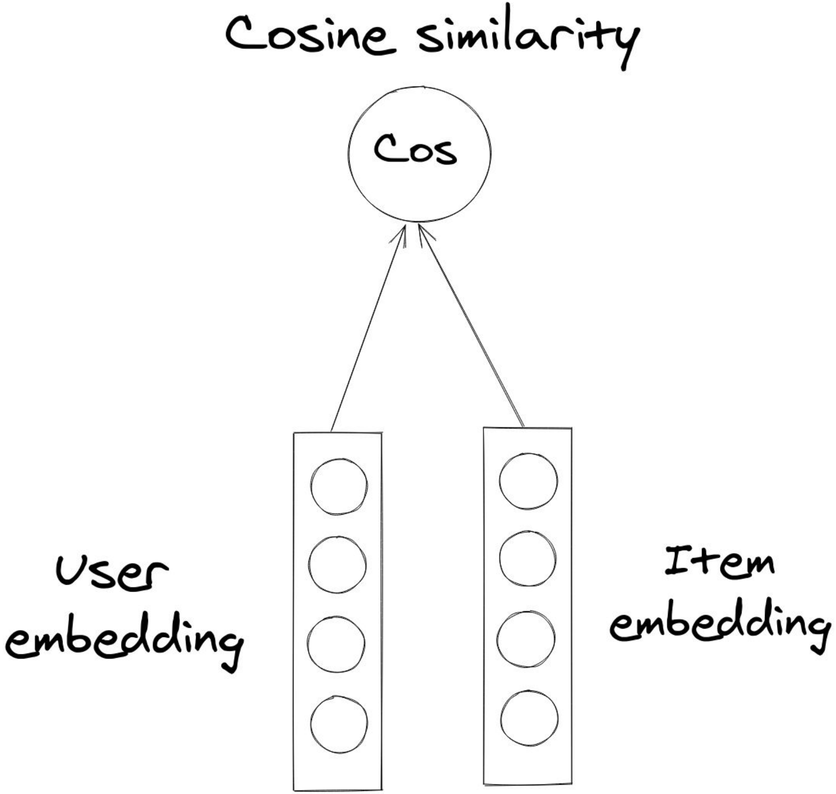




BERT4Rec

Relevence prediction

- Randomly sample negative candidates
- Apply L Transformer Layers
- Mean pooling over hidden vectors
- Binary cross-entropy loss



BERT4Rec

Next Item Prediction

- Randomly mask some positions in interaction history
- Apply L Transformer Layers
- Softmax for masked items
- Cross-entropy loss

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

$$\textbf{Labels}:\ [\texttt{mask}]_1 = v_2, \quad [\texttt{mask}]_2 = v_4$$

Evaluation Metrics

- Accuracy (Relevance Prediction)
- hit@10 (Next Item Prediction)
- ndcg@10 (Next Item Prediction)

Experiment Data

Datasets	#users	#items	#actions	Avg. length	Density
ML-1m	6040	3416	1.0m	163.5	4.79%
Yandex films	63727	25654	0.94m	14.7	0.0573%

Results

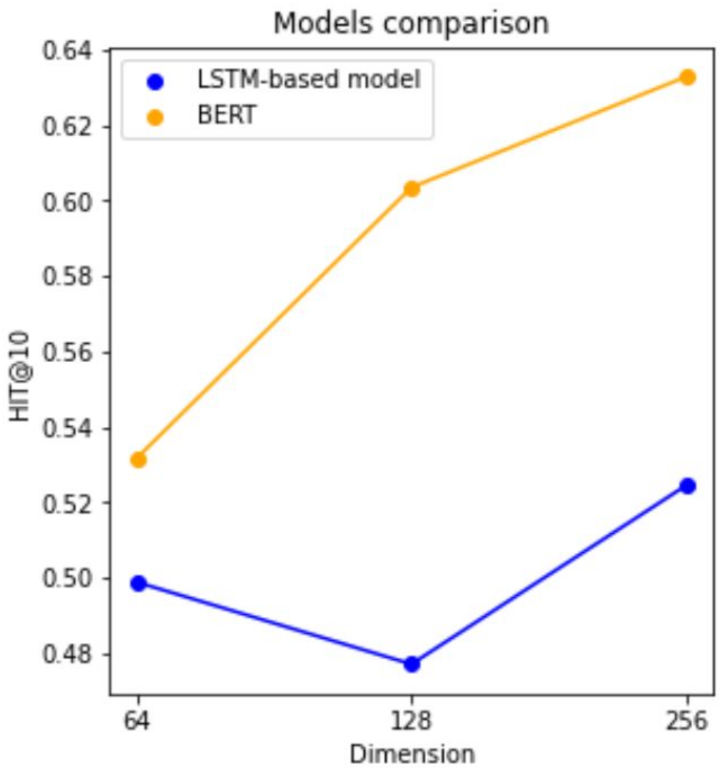
Relevance Prediction

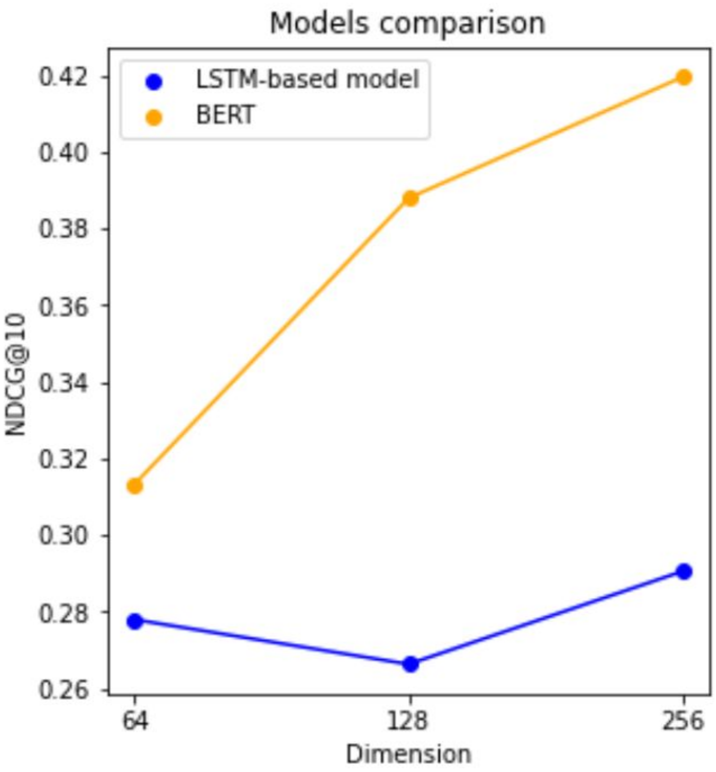
	LSTM-based model	BERT4Rec
Yandex small dataset	**0.908**	0.841*
Yandex full dataset	0.906	**0.912**

*Model overfitting

Results

Next Item Prediction





Conclusion

- Achieved **0.5%** quality boost in the final metrics
- Implemented baseline LSTM-based model
- Implemented BERT4Rec model
- Compared models performance on two different tasks