Introduction to Recommendation System using BERT4Rec

Department of Statistics - IPB University
Jan 23rd 2021

tokopedia

About me

Dian Ayuningtyas

Data Scientist Tokopedia
2019 - now

Data Trainee Tokopedia
2019

Department of Statistics - IPB
2015 - 2019

Contents

tokopedia

About Tokopedia

Vision

Build a Super Ecosystem where anyone can start and discover anything

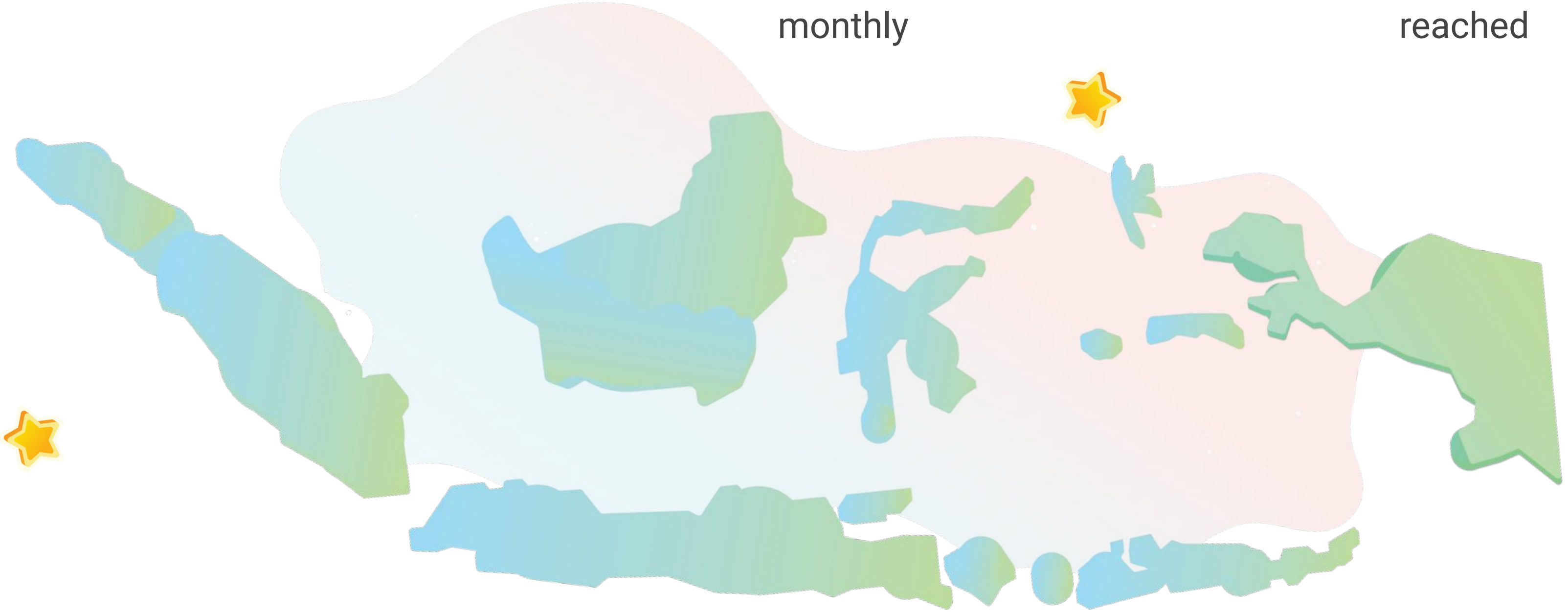
Mission

Democratize commerce through technology

4,500+
Nakama

100M+
Active users monthly

98%
Districts reached



350M+
Products

9.4M+
Merchants

86.5%
First-time entrepreneurs

Our Guiding Principles

Focus on
Consumer

Growth
Mindset

Make It Happen,
Make It Better

Data Office Tokopedia

Recommender System

What will you do when your friend ask these?

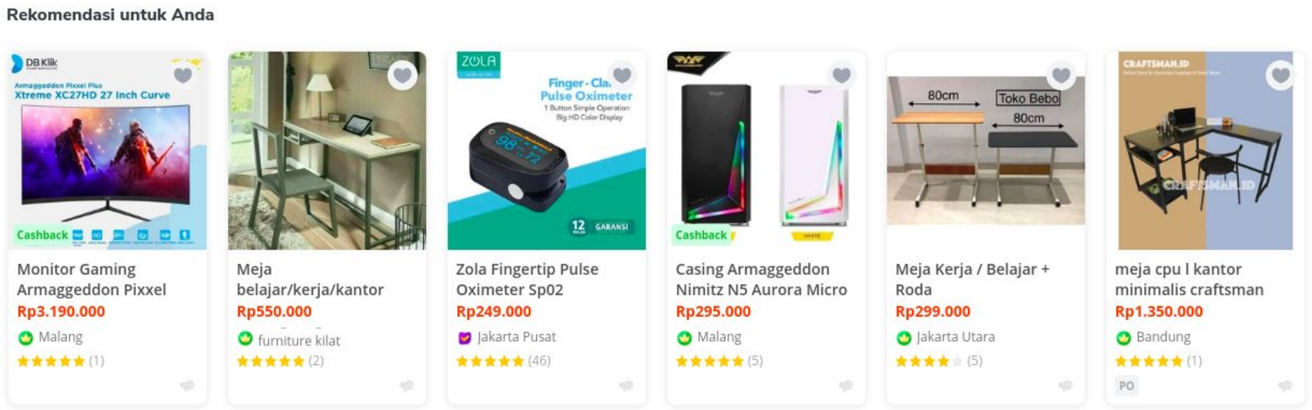
What **food** should I buy for lunch?

What **movie** should I watch tonight?

Recommend

Recommend = show something she/he **might be interested in**

Recommender System



The main purpose of a system is to **fill in the gaps**, deciding whether a product that **would interest him/her**

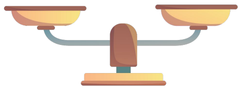
Focus on Consumer

Buyer

Improving customer satisfaction, personalized experience

Seller

Improving revenue and impression, improve CTRs and conversions



Tokopedia

Customer satisfaction which translates into increased customer loyalty, increased consumption, and more profit

Recommender System

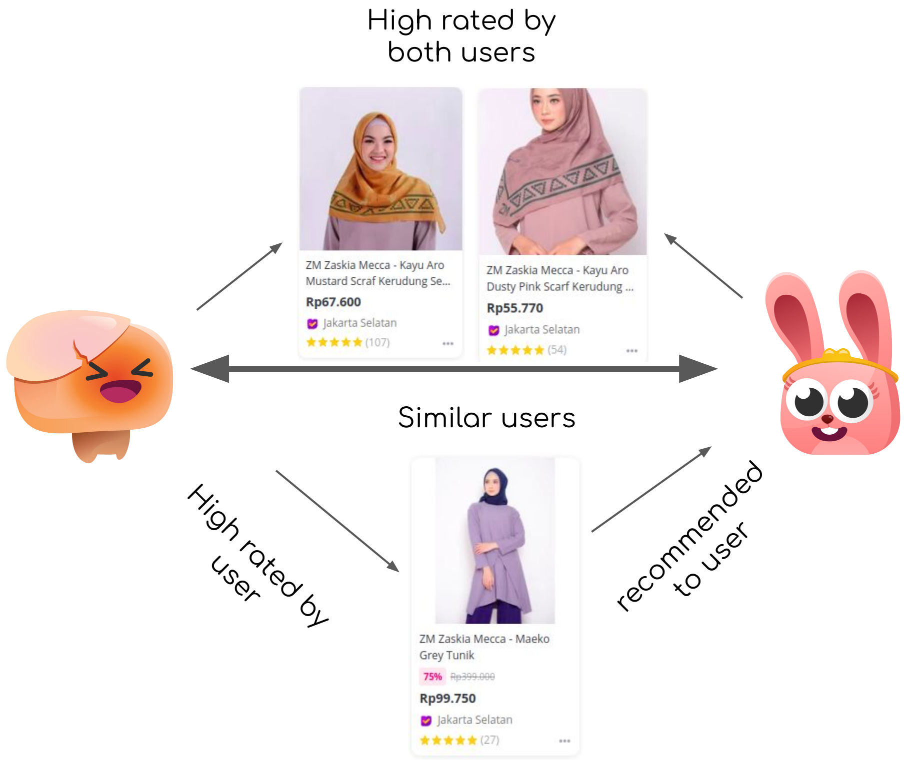
Basic Models of Recommender System

Approach	Conceptual Goal	Inputs
Collaborative Filtering	Give me recommendations based on a **collaborative approach** that leverages the **ratings and actions of my peers/myself**.	User ratings + community ratings
Content based	Give me recommendations based on the **content (attributes) I have favored** in my past ratings and actions.	User ratings + item attributes
Knowledge based	Give me recommendations based on **my explicit specification** of the kind of content (attributes) I want	User specification + item attributes + domain knowledge

Sources : Recommender Systems: The Textbook by Charu C. Aggarwal

Recommender System

Collaborative Filtering

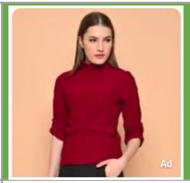


Collaborative filtering models **use the collaborative power of the ratings** provided by multiple users to make recommendations.

Recommender System

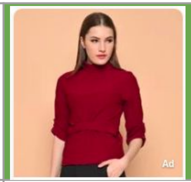
Collaborative Filtering

User Based

				
Novita		2 ★		2 ★
Ani		3 ★		3 ★
Diah	5 ★	2 ★	4 ★	
Tyas	5 ★	2 ★	?	

Find user with the **same taste** as Tyas

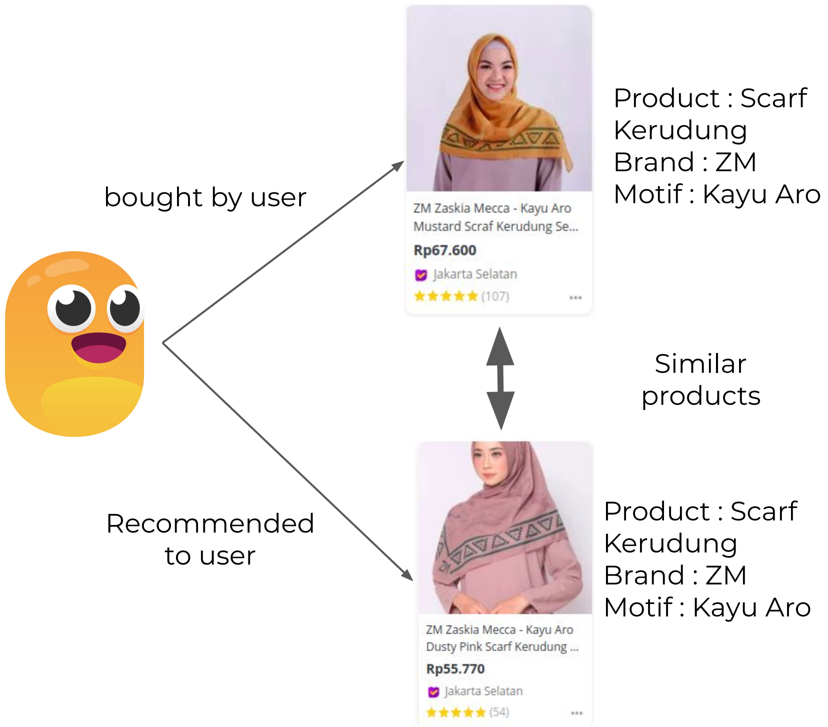
Item Based

				
Novita		2 ★		2 ★
Ani		3 ★		3 ★
Diah	5 ★	2 ★	4 ★	
Tyas	5 ★	2 ★		?

Find item with the **same ratings** as item 4

Recommender System

Content Based



Content based use **content (description)** of the items to give recommendation

Content based model can also built by creating a **classification or regression model for each user**

Recommender System

Content Based

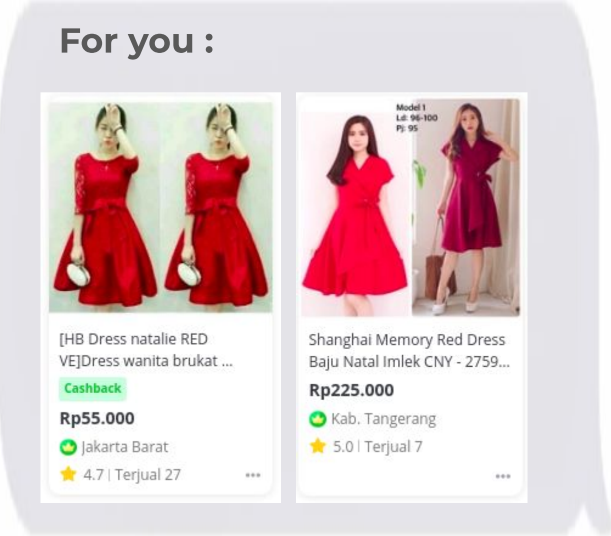
Disadvantages :

- Obvious recommendation : model will always give similar items to users, **no diversification**
- Can not handle new users : model **needs previous rating or purchase behaviour** to give the next recommendation

Recommender System

Knowledge Based

Give recommendation based on user's **explicit specification**



How to quantitatively assess the quality of recommender system?

Metrics

Hit Rate

Hit rate at k (HR@k) is the proportion of label appear in the top-k prediction/ recommendation. Often called recall@k

Example:
HR@5 of the data is ⅔ = 0.6667

Label	Recommended products	Hit@5
A	A, B, C, D, E	1
X	A, V, X, B, F	1
J	M, K, D, E, F	0

Metrics

nDCG (Normalized Discounted Cumulative Gain)

nDCG@k is a ranking metrics that can give higher weight to prediction result that put the label in higher rank

$$\mathrm{DCG_p} = \sum_{i=1}^{p} \frac{2^{rel_i} - 1}{\log_2(i+1)} \qquad \mathrm{IDCG_p} = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{\log_2(i+1)} \qquad \mathrm{nDCG_p} = \frac{DCG_p}{IDCG_p}$$

Example:

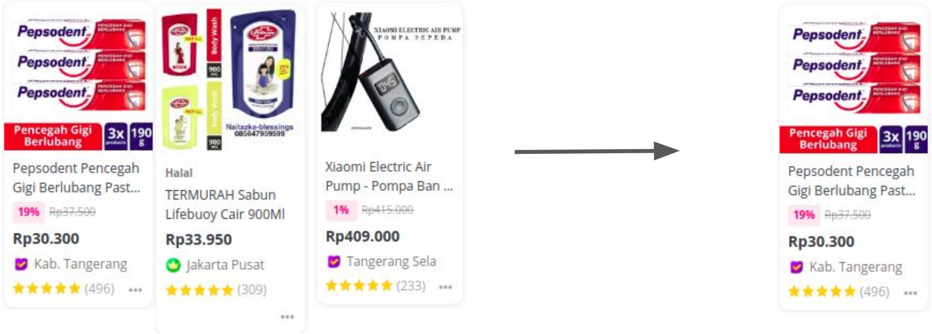
IDCG@5 = 1

nDCG@5 = 0.5

Label	Recommended products	DCG@5	nDCG@5
A	A, B, C, D, E	1	1
X	A, V, X, B, F	0.5	0.5
J	M, K, D, E, F	0	0

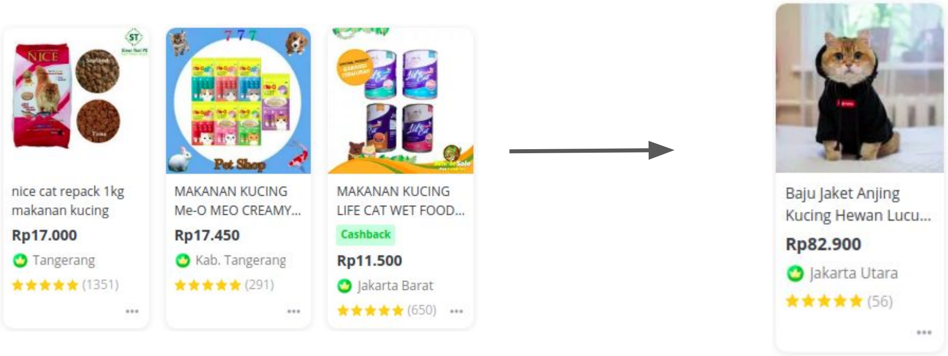
User behaviors in e-commerce

Repeat purchase

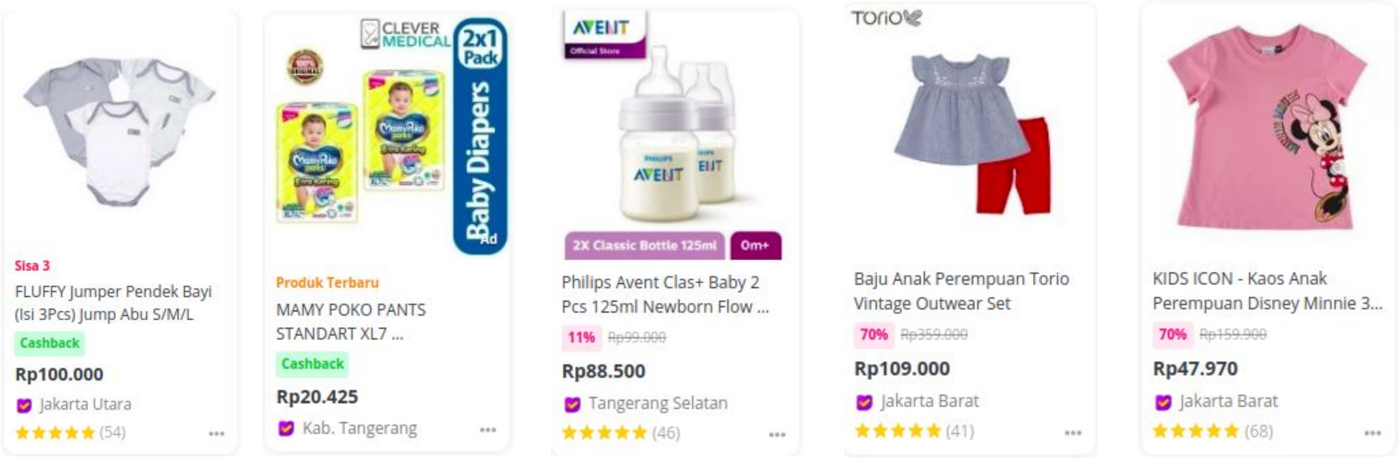


The order of product purchased is **important!**

Complementary products



The methods explained before **ignore the order** in users' behaviors
We should use **sequential recommendation model** to capture sequential patterns from user historical interactions

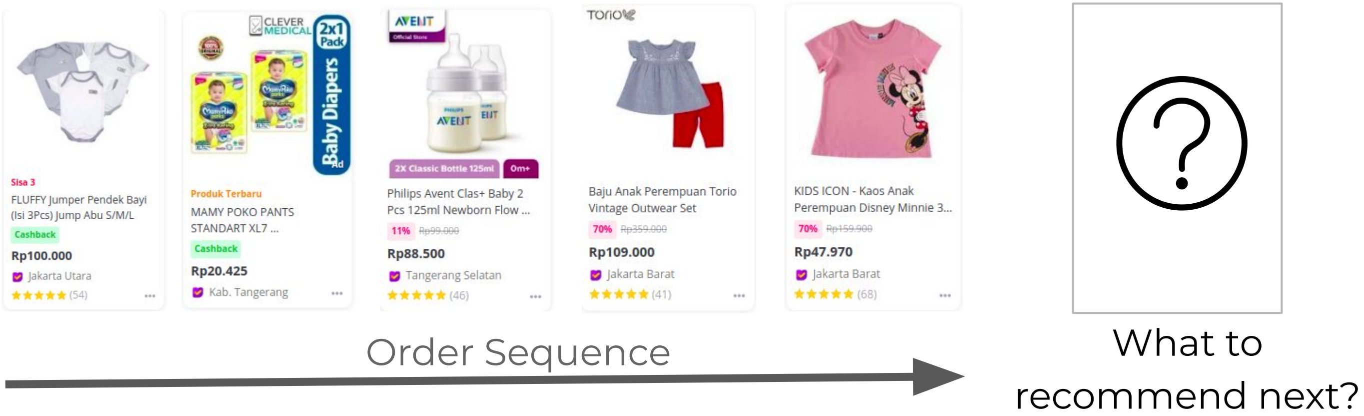


Order Sequence

Sequential Recommendation

Unidirectional

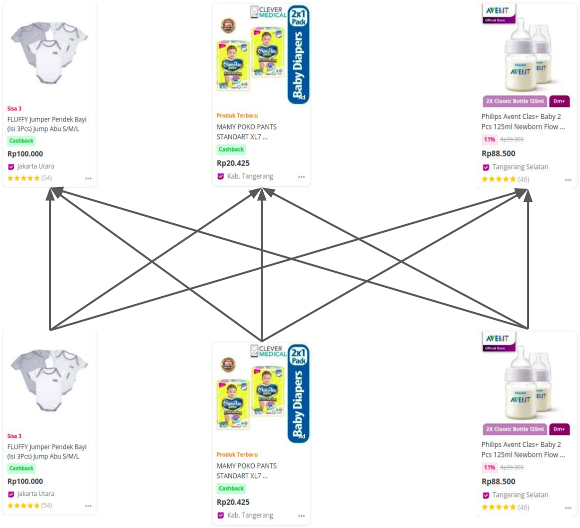
Summarized **user's previous records** into a vector
Example : Markov Decision Processes (MDPs), often use RNN and its variants (LSTM, GRU)



Sequential Recommendation

Bidirectional

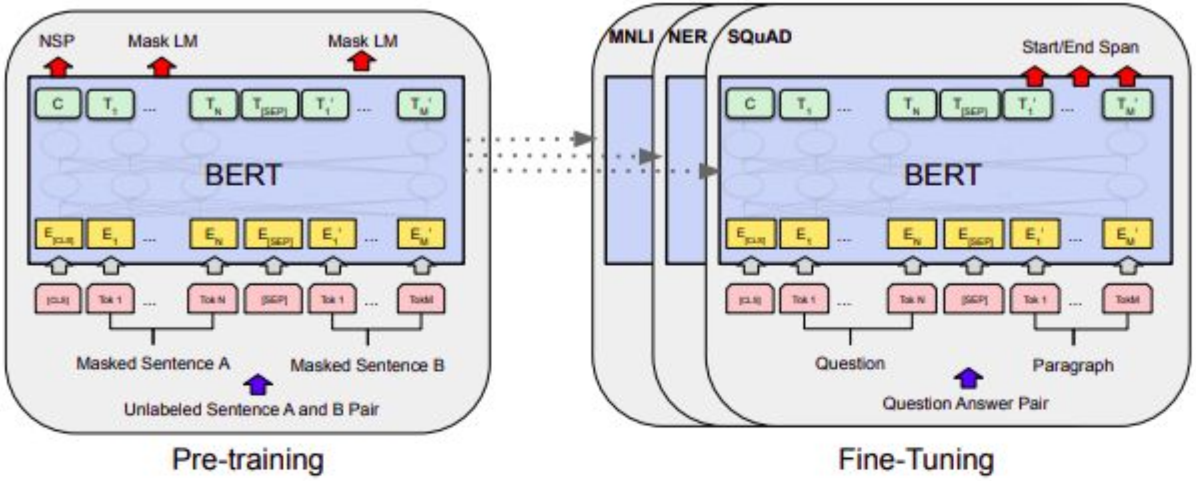
Learns hidden product representation **from left and right side of sequence**.
Because of that, we relaxes assumption of strict order of the sequence
Example : BERT4Rec



BERT4Rec

BERT Overview

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained model generated by Google AI team that can be used for wide range of tasks in NLP



Source : BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

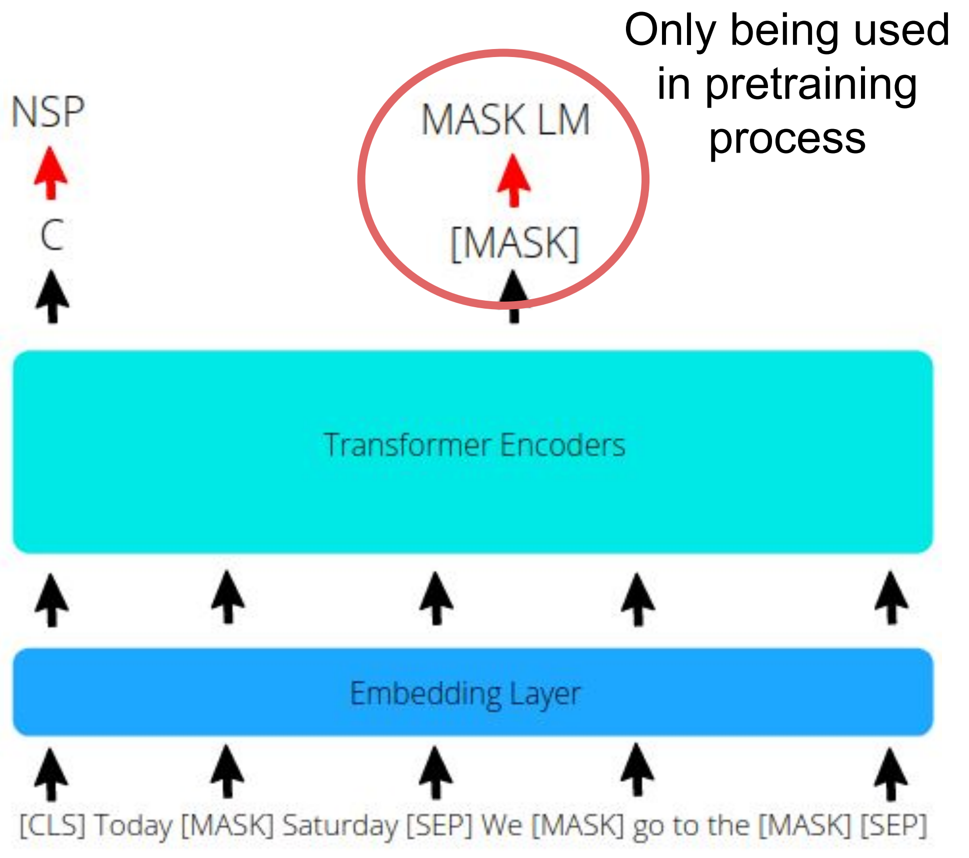
BERT Architecture

BERT trained with inputs :

- Two sentences with random masked words

BERT trained using two tasks :

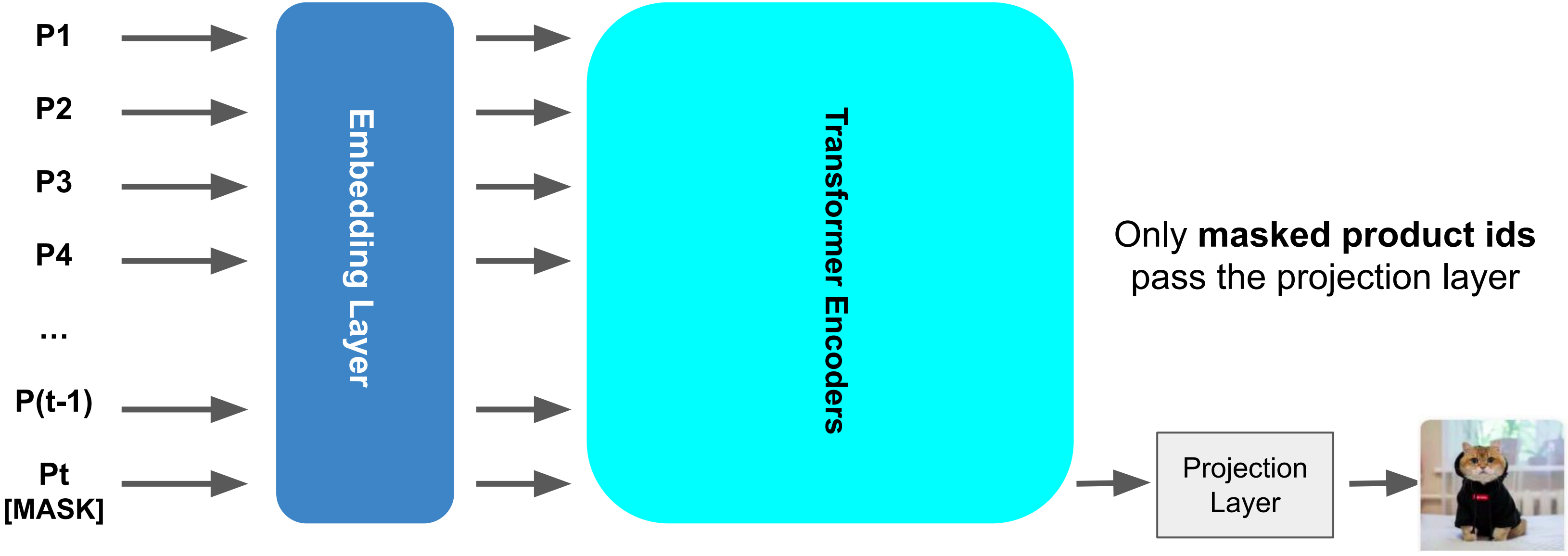
- **MLM** (Mask Language Model) : predict the masked words
- **NSP** (Next Sentence Prediction) : Predict *IsNext or NotNext*



BERT vs BERT4Rec

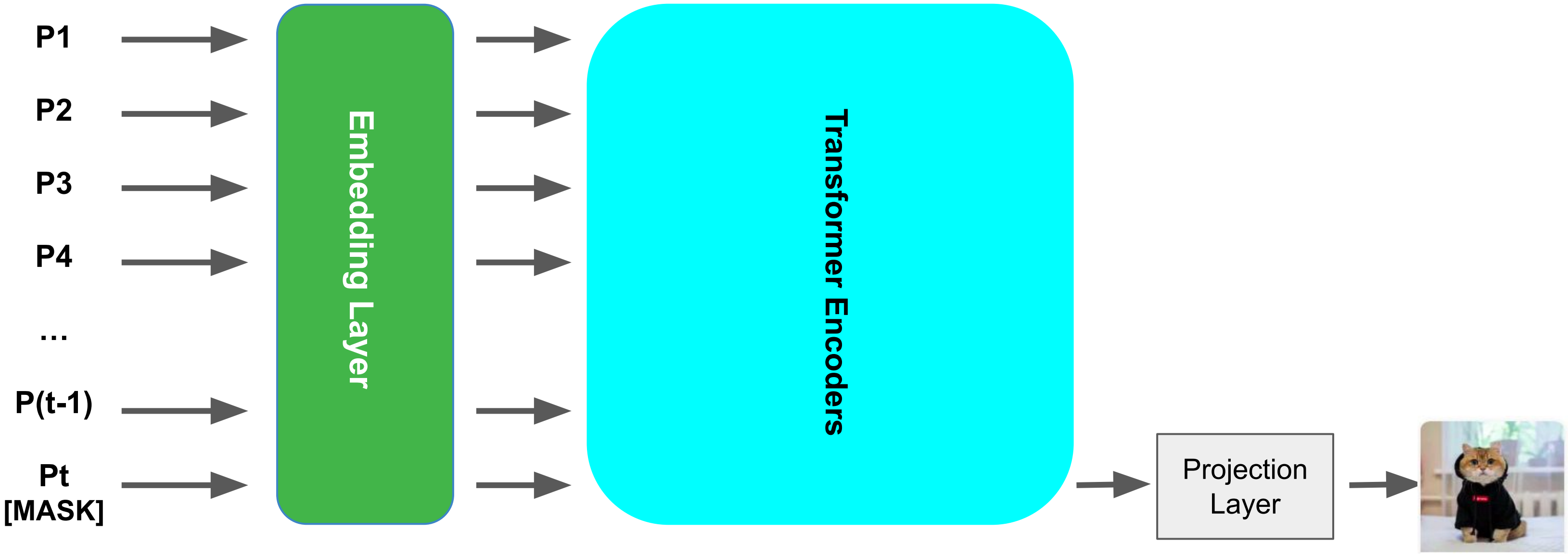
BERT	BERT4Rec
Used in NLP task	Used in recommendation task
Pretrained model that can be used in many NLP cases	Recommendation model specific for a set of products in a platform
Multitask learning (NSP and MLM)	One task learning (MLM/Cloze task)
Token represent word/subword	Token represent product id

BERT4Rec Architecture



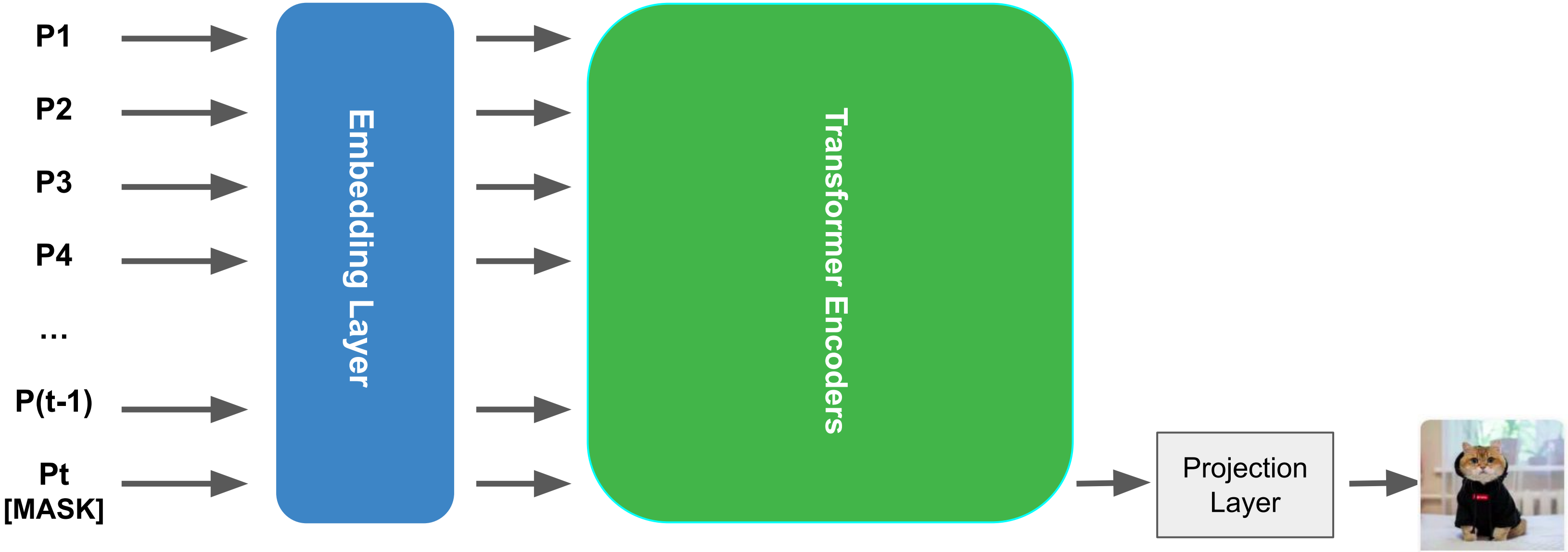
Embedding Layer

Result of this layer is **embedding vectors of the related products** injected with learnable sinusoid positional embeddings



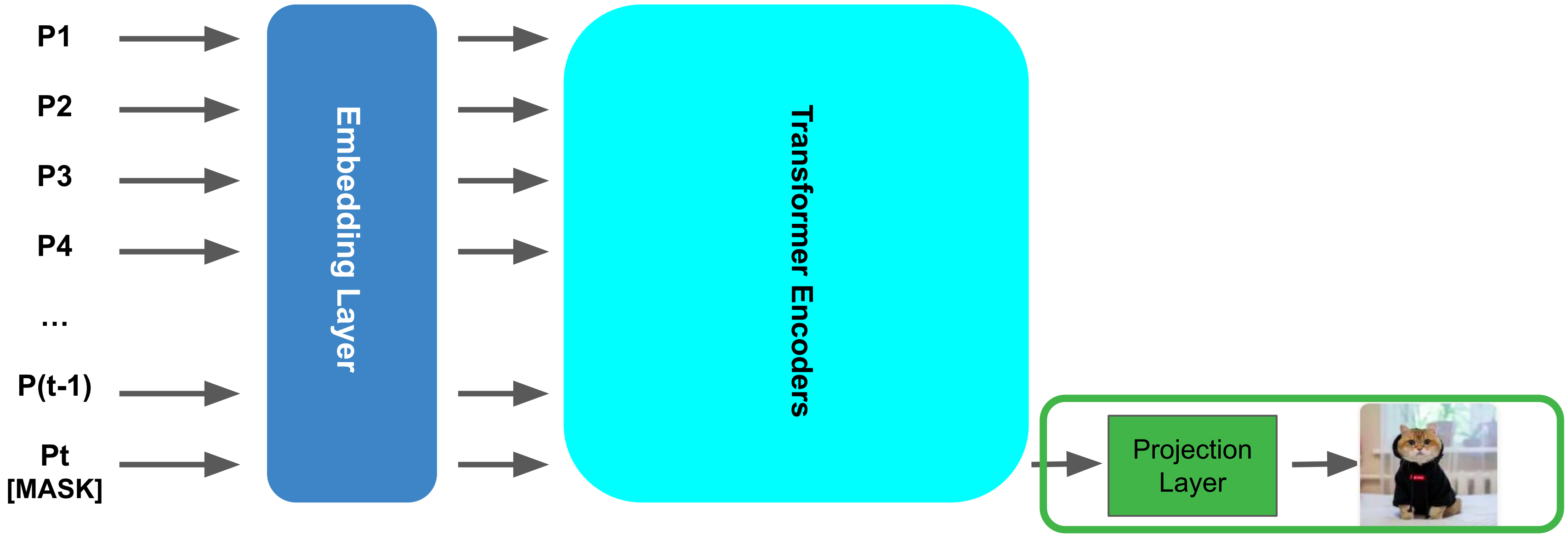
Transformer Encoders

This encoder layer in BERT4rec **stack multiple transformer layers** to gather information from different position in the sequence



BERT4Rec Architecture

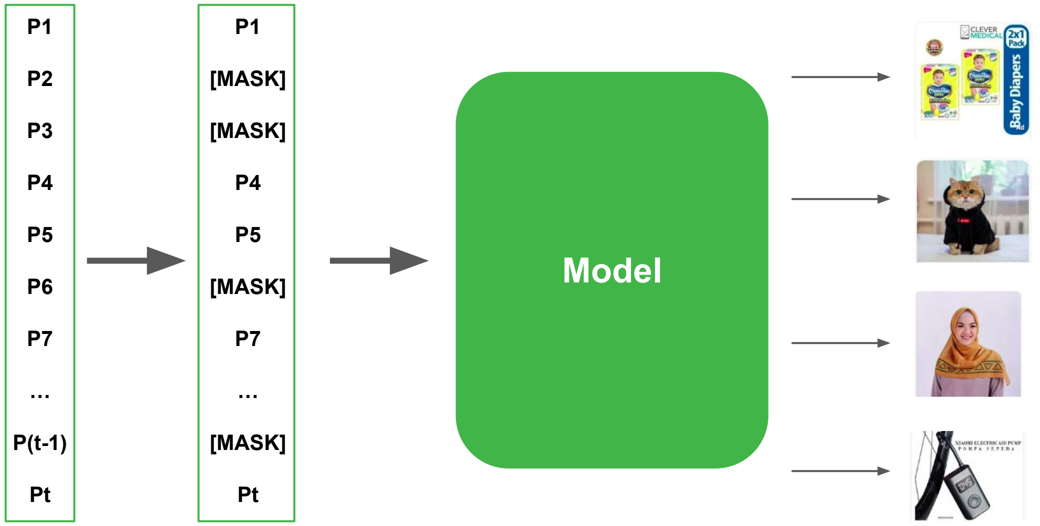
This projection layer is **final part of BERT4Rec** to decide which product will be recommended



Model learning

Before masking the products using Cloze task, there is one important step. It is to **duplicate the sequence k times** (k : parameter duplication factor)

This is such an example of **augmenting the data** so we the model can learn from more masked products in each sequence



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

Calculate the loss

Evaluation

Public dataset

Amazon Beauty : This is a series of product review datasets crawled from Amazon.com by McAuley et al. They split the data into separate datasets according to the top level product categories on Amazon. In this work, we adopt the “Beauty” category

Steam : This is a dataset collected from Steam, a large online video game distribution platform, by Kang and McAuley

MovieLens : This is a popular benchmark dataset for evaluating recommendation algorithms. In this work, we adopt two well-established versions, MovieLens 1m (ML1m) and MovieLens 20m (ML-20m)

Source : BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

BERT4Rec Performance on Public Dataset

Datasets	Metric	POP	BPR-MF	NCF	FPMC	GRU4Rec	GRU4Rec$^+$	Caser	SASRec	BERT4Rec	Improv.
Beauty	HR@1	0.0077	0.0415	0.0407	0.0435	0.0402	0.0551	0.0475	0.0906	**0.0953**	5.19%
	HR@5	0.0392	0.1209	0.1305	0.1387	0.1315	0.1781	0.1625	0.1934	**0.2207**	14.12%
	HR@10	0.0762	0.1992	0.2142	0.2401	0.2343	0.2654	0.2590	0.2653	**0.3025**	14.02%
	NDCG@5	0.0230	0.0814	0.0855	0.0902	0.0812	0.1172	0.1050	0.1436	**0.1599**	11.35%
	NDCG@10	0.0349	0.1064	0.1124	0.1211	0.1074	0.1453	0.1360	0.1633	**0.1862**	14.02%
	MRR	0.0437	0.1006	0.1043	0.1056	0.1023	0.1299	0.1205	0.1536	**0.1701**	10.74%
Steam	HR@1	0.0159	0.0314	0.0246	0.0358	0.0574	0.0812	0.0495	0.0885	**0.0957**	8.14%
	HR@5	0.0805	0.1177	0.1203	0.1517	0.2171	0.2391	0.1766	0.2559	**0.2710**	5.90%
	HR@10	0.1389	0.1993	0.2169	0.2551	0.3313	0.3594	0.2870	0.3783	**0.4013**	6.08%
	NDCG@5	0.0477	0.0744	0.0717	0.0945	0.1370	0.1613	0.1131	0.1727	**0.1842**	6.66%
	NDCG@10	0.0665	0.1005	0.1026	0.1283	0.1802	0.2053	0.1484	0.2147	**0.2261**	5.31%
	MRR	0.0669	0.0942	0.0932	0.1139	0.1420	0.1757	0.1305	0.1874	**0.1949**	4.00%
ML-1m	HR@1	0.0141	0.0914	0.0397	0.1386	0.1583	0.2092	0.2194	0.2351	**0.2863**	21.78%
	HR@5	0.0715	0.2866	0.1932	0.4297	0.4673	0.5103	0.5353	0.5434	**0.5876**	8.13%
	HR@10	0.1358	0.4301	0.3477	0.5946	0.6207	0.6351	0.6692	0.6629	**0.6970**	4.15%
	NDCG@5	0.0416	0.1903	0.1146	0.2885	0.3196	0.3705	0.3832	0.3980	**0.4454**	11.91%
	NDCG@10	0.0621	0.2365	0.1640	0.3439	0.3627	0.4064	0.4268	0.4368	**0.4818**	10.32%
	MRR	0.0627	0.2009	0.1358	0.2891	0.3041	0.3462	0.3648	0.3790	**0.4254**	12.24%
ML-20m	HR@1	0.0221	0.0553	0.0231	0.1079	0.1459	0.2021	0.1232	0.2544	**0.3440**	35.22%
	HR@5	0.0805	0.2128	0.1358	0.3601	0.4657	0.5118	0.3804	0.5727	**0.6323**	10.41%
	HR@10	0.1378	0.3538	0.2922	0.5201	0.5844	0.6524	0.5427	0.7136	**0.7473**	4.72%
	NDCG@5	0.0511	0.1332	0.0771	0.2239	0.3090	0.3630	0.2538	0.4208	**0.4967**	18.04%
	NDCG@10	0.0695	0.1786	0.1271	0.2895	0.3637	0.4087	0.3062	0.4665	**0.5340**	14.47%
	MRR	0.0709	0.1503	0.1072	0.2273	0.2967	0.3476	0.2529	0.4026	**0.4785**	18.85%

Sources : BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer

Conclusion

Conclusion

- Basic Models of Recommender System are **collaborative filtering, content based, and knowledge based**
- **BERT4Rec** is one of a bidirectional sequential recommendation approach
- BERT4Rec use ***Cloze task*** that aims to predict the masked item in each sequence. This *Cloze task* also helps us to get product representations utilizing **both left and right context**
- **Main components of BERT4Rec** are embedding layer, transformer encoder, and the projection layer
- Evaluation on public datasets shows that **BERT4Rec performs better** that unidirectional sequential recommendation approaches

Sources

- **BERT4Rec** : https://arxiv.org/pdf/1904.06690.pdf
- **BERT** : https://arxiv.org/pdf/1810.04805.pdf
- **Attention is all you need** : https://arxiv.org/pdf/1706.03762.pdf
- **Recommender Systems: The Textbook** by Charu C. Aggarwal


Thank you!