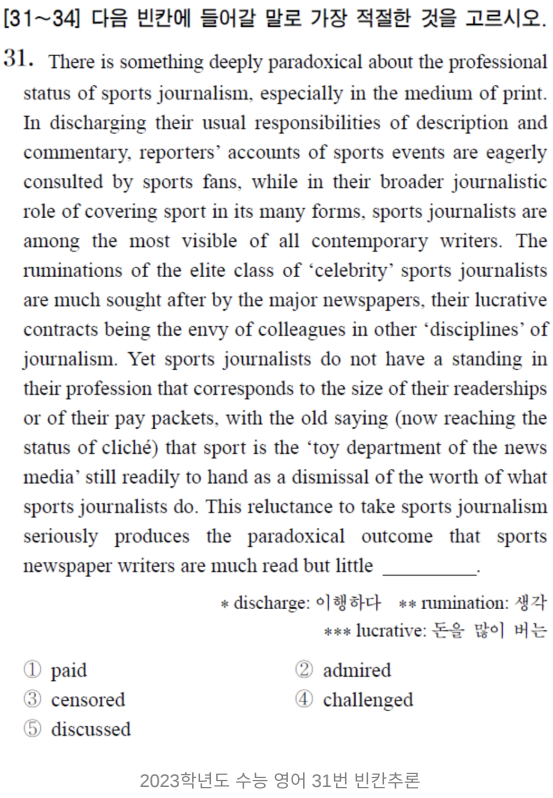
[3조] BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer (Sun et al., 2019)

BERT?

- Bidirecgtional Encoder Representations from Transformers의 약자
- Transformer 아키텍처를 기반으로 한 언어 모델
- 양방향(bidirectional)으로 문맥을 파악하여 단어나 문장의 의미를 이해



[31~34] 다음 빈칸에 들어갈 말로 가장 적절한 것을 고르시오.

31. There is something deeply paradoxical about the professional status of sports journalism, especially in the medium of print. In discharging their usual responsibilities of description and commentary, reporters' accounts of sports events are eagerly consulted by sports fans, while in their broader journalistic role of covering sport in its many forms, sports journalists are among the most visible of all contemporary writers. The ruminations of the elite class of 'celebrity' sports journalists are much sought after by the major newspapers, their lucrative contracts being the envy of colleagues in other 'disciplines' of journalism. Yet sports journalists do not have a standing in their profession that corresponds to the size of their readerships or of their pay packets, with the old saying (now reaching the status of cliché) that sport is the 'toy department of the news media' still readily to hand as a dismissal of the worth of what sports journalists do. This reluctance to take sports journalism seriously produces the paradoxical outcome that sports newspaper writers are much read but little _____.

* discharge: 이행하다 ** rumination: 생각
*** lucrative: 돈을 많이 버는

① paid ② admired
③ censored ④ challenged
⑤ discussed

2023학년도 수능 영어 31번 빈칸추론

💡 [참고자료]
BERT의 기본 개념인 Transformer 모델과 BERT에 대한 이해를 돕는 자료 by Jay Alammar

> Visualizing A Neural Machine Translation Model (Mechanics of Seq2seq Models With Attention)
> 최근 10년 동안의 자연어 처리 연구 중에 가장 영향력이 컸던 3가지를 꼽는 서베이에서 여러 연구자들이 꼽았던 연구가 바로 2014년에 발표됐던 sequence-to-sequence (Seq2seq) + Attention 모델입니다 (Sutskever et al., 2014, Cho et al., 2014).
> https://nlpinkorean.github.io/visualizing-neural-machine-translation-mechanics-of-seq2seq-models-with-attention/

Transformer model

> The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)
> 저번 글에 이어 이번엔 다른 contextualized Language Model 들인 BERT와 ELMo에 대한 글을 번역해보았습니다. 마찬가지로 블로그 by Jay Alammar에서 허락을 받고 가져온 글이며, 원문은 본 링크 에서 확인하실 수 있습니다.
> https://nlpinkorean.github.io/illustrated-bert/

BERT

BERT4Rec 이전의 추천시스템

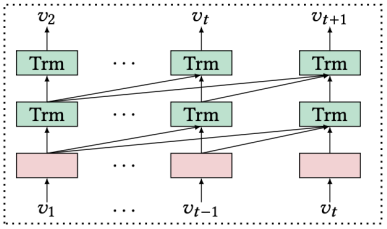
- General Recommendation:
  - Collaborative Filtering (CF) - Matrix Factorization (MF)
  - CF + RBM(Restricted Boltzmann Machines)
  - CF + auxiliary information (text, image, audio)
  - NCF

  💡 유저 과거 행동의 순서를 고려하지 않음

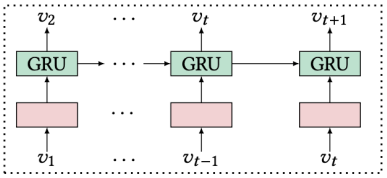
- Sequential Recommendation:
  - [MCs(Markov chains) based]
    MDPs(Markov Decision Processes), FPMC(Factorizing Personalized Markov Chains)
  - [RNN based]
    GRU4Rec, DREAM(Dynamic REcurrnet bAsket Model), NARM(attention-based GRU)
  - [other NN based]
    Caser(Convolutional Sequence Model), STAMP(Short-Term Attention/Memory Priority Model for Session-based Recommendation)
  - [Attention based]
    SASRec

  💡 유저 과거 행적과 선택의 맥락을 잘 반영하는가?

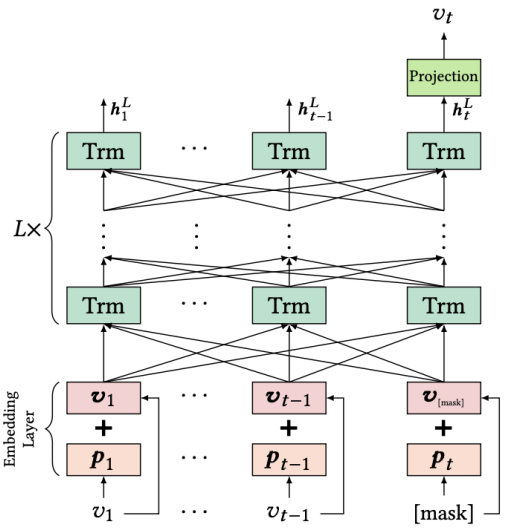
- 이전 추천시스템 vs. BERT4Rec
  - architecture: 단방향(unidirectional) → 양방향(bidirectional)
  - Cloze task (빈칸 맞추기)를 도입해 양방향 구조 채택으로 인한 문제 해결



(c) SASRec model architecture.



(d) RNN based sequential recommendation methods.



(b) BERT4Rec model architecture.

💡 BERT4Rec은 유저의 상품 구매 맥락을 보다 잘 반영하기 위해 양방향 transformer 구조의 인코더를 활용한 추천시스템 모델

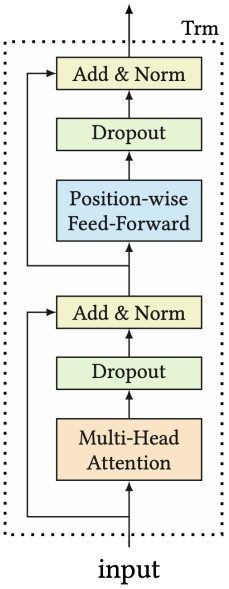
BERT4Rec의 구조

- **문제정의**
  - $\mathcal{U}$: a set of users
    v: a set of items
    S_u: the interaction sequence in chronological order for user u
    n_u: the length of interation sequence for user u
  - 목표는 아이템 상호작용 이력이 주어졌을 때, 사용자 u가 n_u+1 시점에서 상호작용할 항목을 예측하는 것

$$p(v_{n_u+1}^{(u)} = v | S_u)$$

아이템에 대한 구매행동 히스토리 S_u가 주어졌을 때, 유저가 n_u+1에 구매할 아이템을 예측 (아이템별 구매 확률을 구함)

- **Transformer Layer**



(a) Transformer Layer.

- Multi-Head Self-Attention
  - Self-Attention: 주어진 시퀀스의 각 위치에서 다른 모든 위치에 대한 "관련성 점수"를 계산
  - Multi-Head Self Attention은 여러 개의 독립적인 Self-Attention 헤드를 사용하는 것
  - 각 헤드는 서로 다른 관점에서 시퀀스의 특징을 추출 → 시퀀스의 맥락을 다양한 관점에서 포착 가능

$$\begin{aligned}\texttt{MH}(\boldsymbol{H}^l) &= [\text{head}_1; \text{head}_2; \ldots; \text{head}_h]\boldsymbol{W}^O \\ \text{head}_i &= \texttt{Attention}(\boldsymbol{H}^l\boldsymbol{W}_i^Q, \boldsymbol{H}^l\boldsymbol{W}_i^K, \boldsymbol{H}^l\boldsymbol{W}_i^V)\end{aligned}$$

$$\texttt{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \text{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{d/h}}\right)\boldsymbol{V}$$

- Position-wise Feed-Forward Network
  - 비선형 함수를 적용하여 모델이 더 복잡한 관계를 학습할 수 있도록 함
    (본 논문은 ReLu 대신 GELU를 사용)
  - 각 Self-Attention의 아웃풋(각 위치에서의 특성을 나타내는 벡터)에 GELU 활성화 함수를 적용

$$\begin{aligned}\texttt{PFFN}(\boldsymbol{H}^l) &= \left[\texttt{FFN}(\boldsymbol{h}_1^l)^\top; \ldots; \texttt{FFN}(\boldsymbol{h}_t^l)^\top\right]^\top \\ \texttt{FFN}(\boldsymbol{x}) &= \texttt{GELU}(\boldsymbol{x}\boldsymbol{W}^{(1)} + \boldsymbol{b}^{(1)})\boldsymbol{W}^{(2)} + \boldsymbol{b}^{(2)} \\ \texttt{GELU}(x) &= x\Phi(x)\end{aligned}$$

각 위치의 입력은 두 선형 변환 층을 통과한 후에 비선형 함수를 적용하여 출력을 생성

- Stacking Transformer Layer
  - residual connection, 각 서브레이어 결과에 dropout 적용
  - 네트워크 훈련의 안정화와 가속화를 위해 같은 레이어에 있는 모든 히든 유닛의 입력을 정규화

$$\begin{aligned}\boldsymbol{H}^l &= \texttt{Trm}(\boldsymbol{H}^{l-1}), \quad \forall i \in [1, \ldots, L] \\ \texttt{Trm}(\boldsymbol{H}^{l-1}) &= \texttt{LN}\left(\boldsymbol{A}^{l-1} + \texttt{Dropout}\left(\texttt{PFFN}(\boldsymbol{A}^{l-1})\right)\right) \\ \boldsymbol{A}^{l-1} &= \texttt{LN}\left(\boldsymbol{H}^{l-1} + \texttt{Dropout}\left(\texttt{MH}(\boldsymbol{H}^{l-1})\right)\right)\end{aligned}$$

- **Embedding Layer**
  - h: transformer layer의 input
    v: item vector
    p: positional encoding → 인풋 시퀀스를 maximum length N에 맞춰 최근 N개만 잘라서 사용
  - 사용자와 상품 정보를 임베딩, 즉 벡터로 표현

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

- **Output Layer**
  - ht^L: t 시점(위치)의 아이템이 마스킹되어 있음
    E^T: 아이템 임베딩 벡터 → 과적합(overfitting) 방지 및 모델 사이즈 축소
  - ht^L로 모든 아이템에 대한 확률을 예측
    → 최종적으로 사용자 u가 n_u+1 시점에서 상호작용할 아이템 도출

$$P(v) = \text{softmax}\left(\text{GELU}(\boldsymbol{h}_t^L \boldsymbol{W}^P + \boldsymbol{b}^P)\boldsymbol{E}^\top + \boldsymbol{b}^O\right)$$

BERT4Rec 학습 및 테스트 실험 결과

- **학습**
  - 양방향 모델의 문제점 해결을 위해 Cloze task 도입: 임의의 비율만큼의 아이템들을 [mask] 토큰으로 대체 → 좌, 우 문맥에 의존해서 원래 id를 예측하도록 함
    Input: $[v_1, v_2, v_3, v_4, v_5]$ → randomly mask → $[v_1, [\text{mask}]_1, v_3, [\text{mask}]_2, v_5]$
    Labels: $[\text{mask}]_1 = v_2$, $[\text{mask}]_2 = v_4$
  - 마스킹된 아이템의 은닉상태(h)도 소프트맥스로 들어가고, negative log-likelihood로 손실 계산

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

S_u: user behavior history
S_u': S_u의 masked version
S_u^m: random masked items
v_m: masked item
v_m^*: true item for the masked item v_m

- **테스트**
  - user behavior의 마지막에 [mask] 토큰을 추가하여, 해당 아이템을 예측하도록 함

- **실험 결과**
  - 실험 데이터셋

데이터셋			데이터 특성
Beauty	아마존 화장품 리뷰	유저, 리뷰, 아이템	sparse data (대부분의 값이 0)
Steam	게임 리뷰	유저, 리뷰, 아이템	sparse data
MovieLens	영화 평점	유저, 평점, 아이템	dense data

  - 성능 지표(Evaluation Metric): 값이 클수록 좋은 성능을 의미
    - Hit Rate (HR@K, K=1,5,10): 추천 시스템에서 사용자가 실제로 구매 또는 상호 작용한 아이템이 추천 목록에 포함된 경우를 측정하는 지표
    - NDCG(Normalized Discounted Cumulative Gain): 추천 목록에서 상위에 위치한 아이템에 대한 가중치를 높게 평가하는 지표

- MRR(Mean Reciprocal Rank) ≈ MAP(Mean Average Precision): 추천된 목록에서 가장 높은 위치에 있는 첫 번째 정답 아이템의 역수를 측정하는 지표

[Recommendation] BERT4Rec : Sequential Recommendation with Bidirectional Encoder Representations from Transformer

추천 알고리즘에서 bert를 적용한 논문 bert4rec 을 살펴보겠다. https://arxiv.org/abs/1904.06690 BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer Modeling users' dynamic and evolving preferences from their historical behaviors is challenging and crucial for recommendation systems. Previous methods employ sequential neural networks (e.g., Recurrent Neural

https://huidea.tistory.com/290

- 모델별 성능 비교 결과

Table 2: Performance comparison of different methods on next-item prediction. Bold scores are the best in each row, while underlined scores are the second best. Improvements over baselines are statistically significant with $p < 0.01$.

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%

  - 단방향인 SASRec보다 성능이 개선되었다는 점에서, 양방향 self-attention, Cloze 학습 방법이 효과가 있다고 볼 수 있음
  - 미래의 정보까지 함께 학습하기에 성능이 개선된 것으로 볼 수 있음
- 논문 4.5 ~ 4.8 부분은 hidden vector 차원, masking 비율(p) sequence 길이(N)에 따른 성능 변화에 대한 연구 결과
  - hidden vector 차원: 16~256까지 조정하면서 실험한 결과 sparse dataset에서는 고차원 불필요
  - mask proportion p: sparse dataset에서는 0.4~0.6 정도, dense dataset에서는 0.2가 적절
  - maximum sequence length N: sparse dataset에서는 짧을수록 우수했으며, dense dataset에서는 길수록 우수한 성능
  - 아키텍처 구조에서 positional embedding이 전체 성능에 중요한 역할을 하고 있음

BERT4Rec 실습

Google Colaboratory

https://colab.research.google.com/drive/1QbmtR-FK7QkdeOttcGSn9zVQNKMemA67?usp=sharing