G-ADLFM

AI빅데이터프로젝트 : 캡스톤디자인 I

아무래도 그렇조

20172848 이지평
20172853 장성현
20192761 김정하

Introduction

" Adaptive Deep Modeling of Users and Items Using Side Information for Recommendation "

→ 기존의 Side Information을 활용하는 방법론은 한 유저와 모든 아이템에 대해서
고정된 Representation 값을 학습하게 되어 'Individual Diversity'를 고려하지 못함

→ 유연성 부족, 부정확한 추천을 야기

→ 따라서 User Input과 Item Input의 Attention Score를 활용한 'Individual Diversity'를 반영할 수 있는 방법론 제시

Introduction

Item Description을 포함한 데이터셋이 굉장히 드물기 때문에

Item Description이 아닌 Item Id를 Input으로 넣으면 일반화의 효과,

즉 적용가능한 데이터셋의 범위를 상당히 넓힐 수 있지 않을까 기대.

→ Input으로 Item Description을 넣은 모델과 Item Id를 넣은 모델의 비교 및

성능 향상을 위한 다양한 모델구조의 실험 진행

Dataset

Anime Recommendation Database 2020

	user_id	anime_id	rating	synopsis
0	0	430	9	In desperation, Edward Elric sacrificed his bo...
1	0	1004	5	It was on a rainy spring day that Chobi became...
2	0	3010	7	Diego Vega returns from his study trip to disc...
3	0	570	7	In an alternate history, following World War I...
4	0	2762	9	fter the death of Saizou, Kabamaru's horribly ...

Baseline 성능 : MSE 2.2380

Dataset

1. 평점 남긴 횟수 최소 Description Num 이상인 유저 추출

2. Description Num parameter만큼 랜덤 추출

```
[[[ 4723,  4399,  2818, ...,     0,     0,     0],
  [25303,  1439,   894, ...,     0,     0,     0],
  [25303,  1439,   894, ...,     0,     0,     0],
  [ 4399,  1651, 10108, ...,     0,     0,     0]],
```

X_des(유저가 기존에 평가했던 항목의 특징)

```
array([4723, 4399, 2818, ...,    0,    0,    0], dtype=int32)
```

X_train(유저가 평가할 아이템의 특징)

```
array([[ 5],
       [ 9],
       [ 6],
       ...,
       [ 5],
       [ 8],
       [10]])
```

평점

Experiment

1. Side Information → Item Id

(Item Description → Item Id)

2. Self-Attention

3. Residual Connection

4. Transformer Encoder

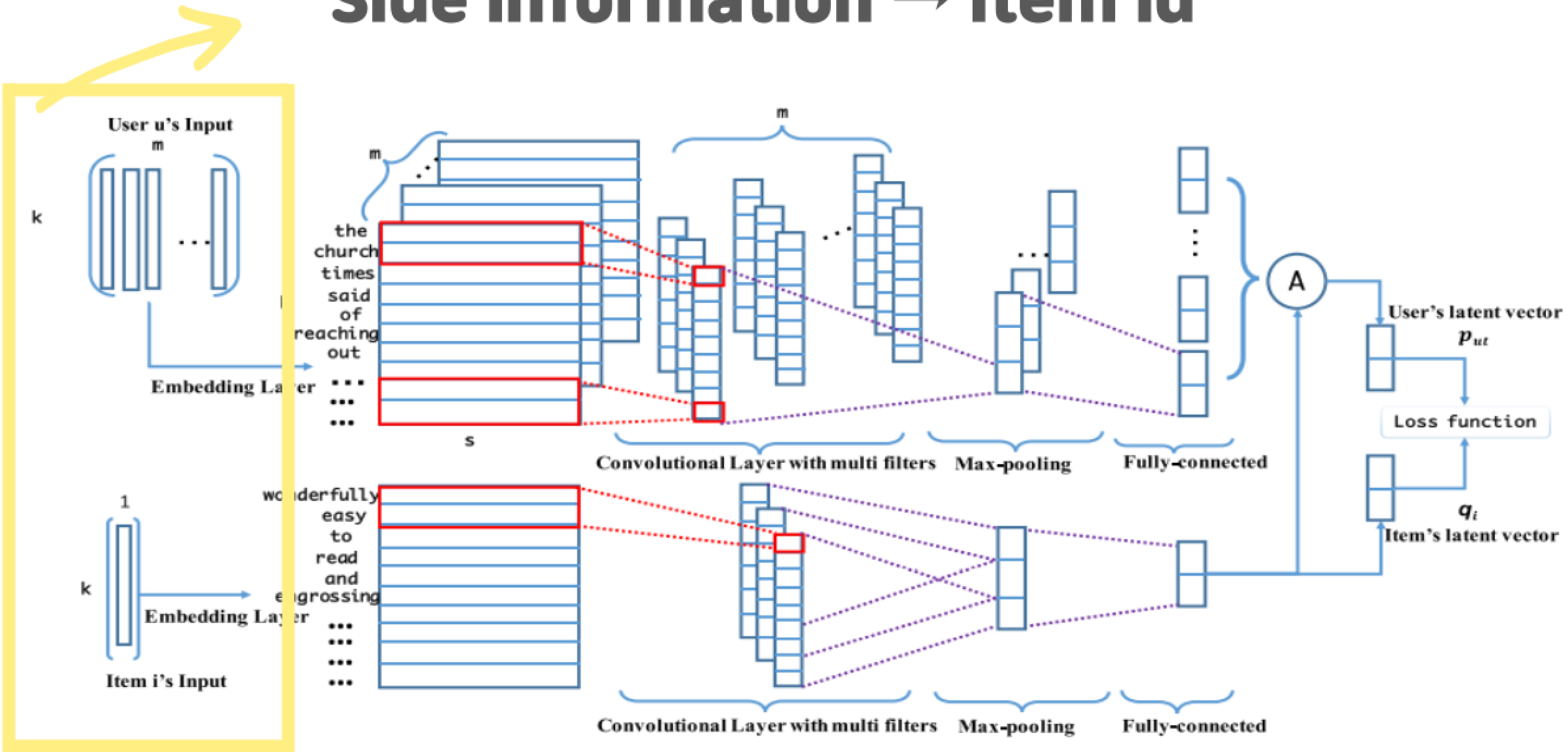
5. Multi-Head Conv1D

6. Multi Conv1D

Description (Id) Num 3, 4, 5

Experiment 1

" Side Information → Item Id "

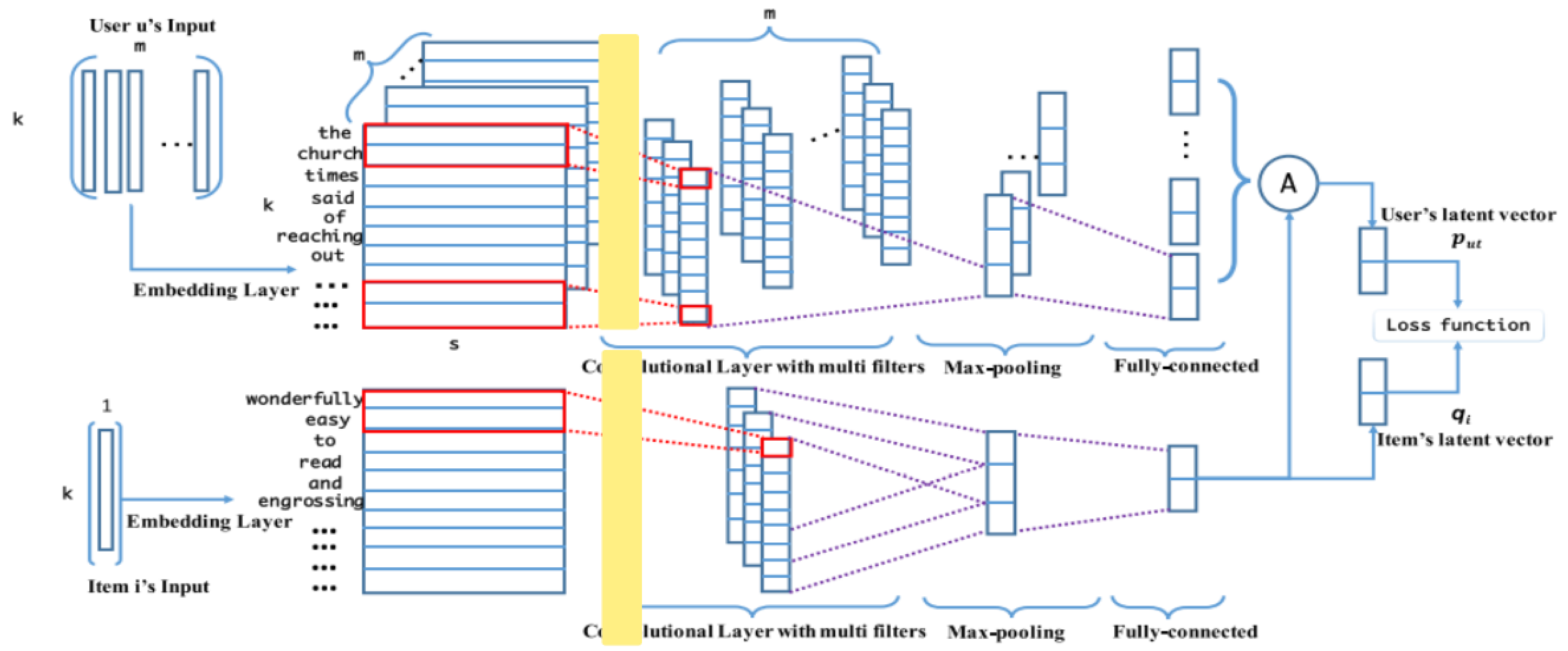


기존 ADLFM : MSE 2.2380

Item Id ADLFM : MSE 2.5767

Experiment 2

" Self-Attention "

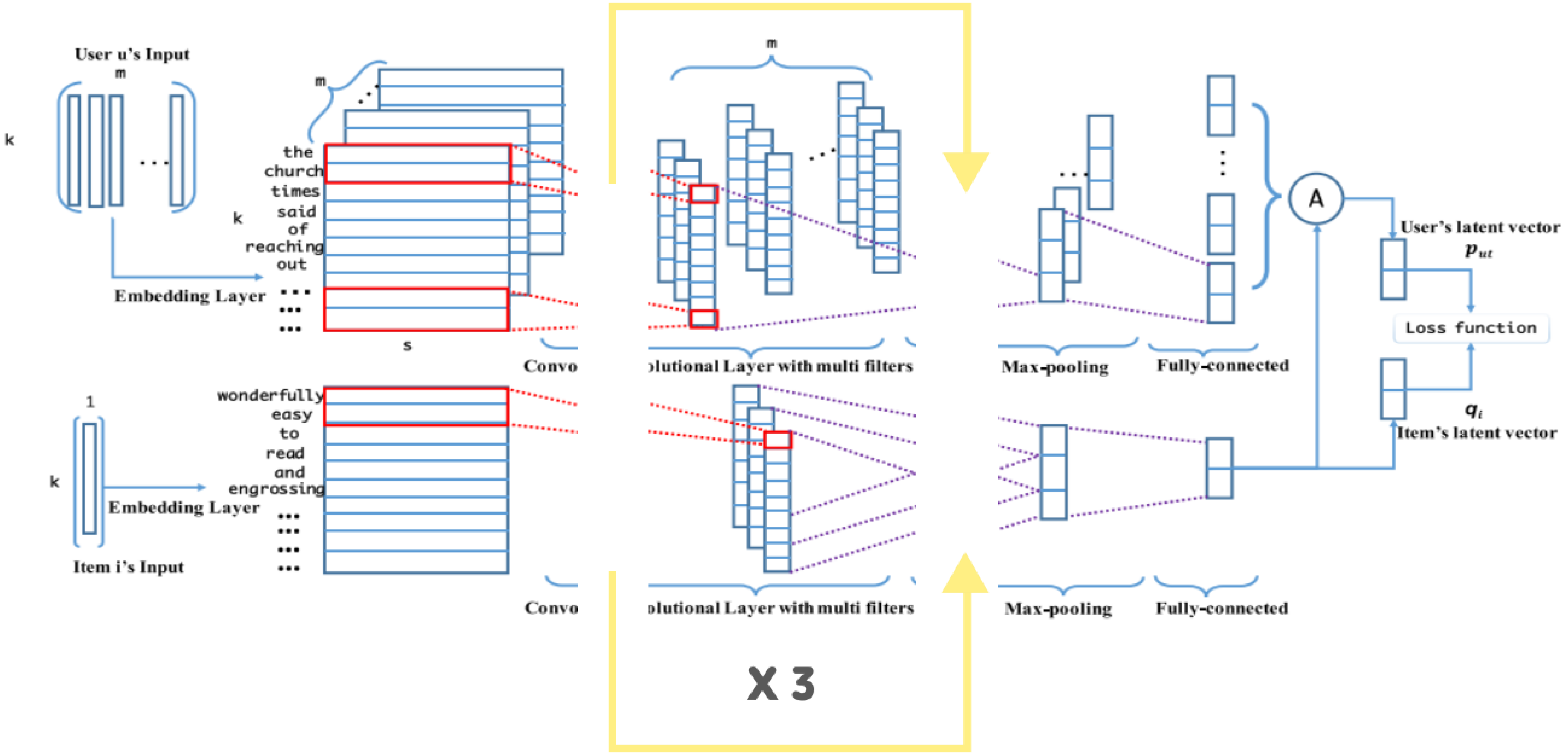


Self-Attention

Bidirectional한 연산을 통한 representation 성능 향상 도모

Experiment 3

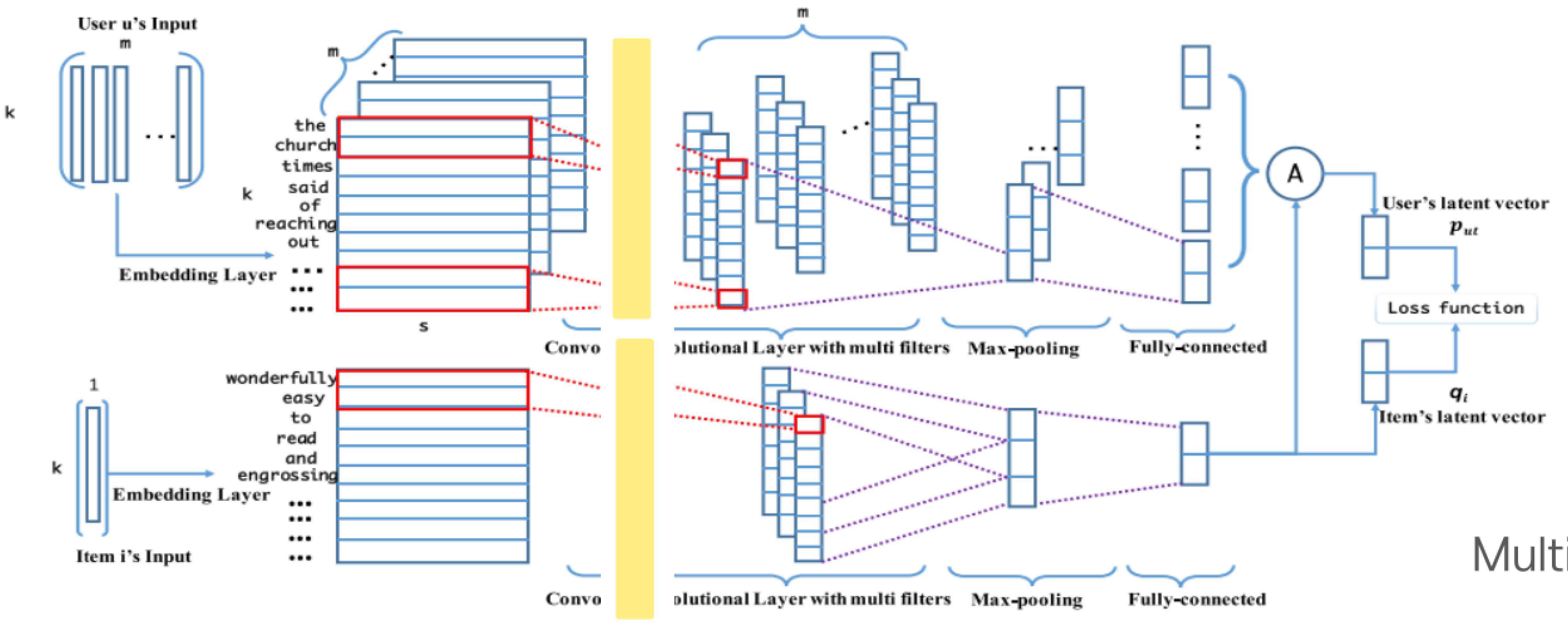
" Residual Connection "



Gradient Vanishing 문제를 해결하기 위한 모델

Experiment 4

" Transformer Encoder "

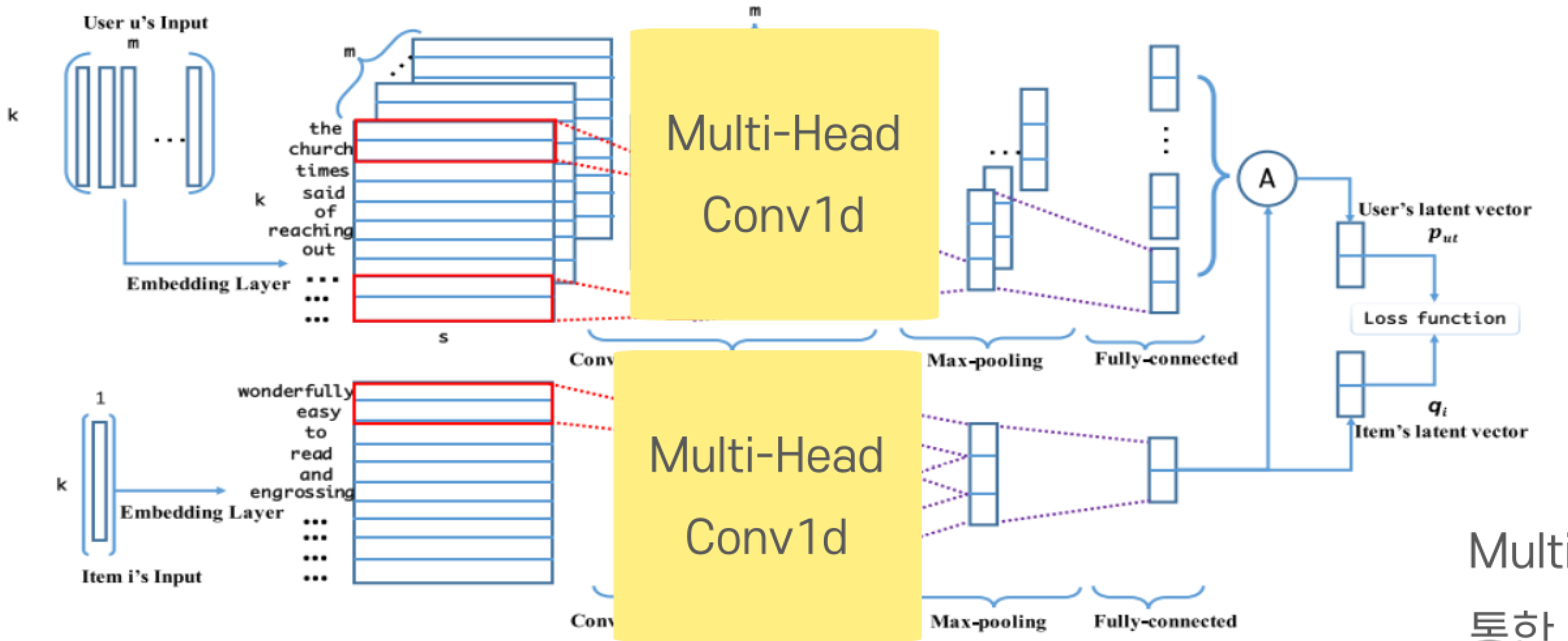


Transformer Encoder

Multi-head attention 구조를 활용한 representation 성능 향상 도모

Experiment 5

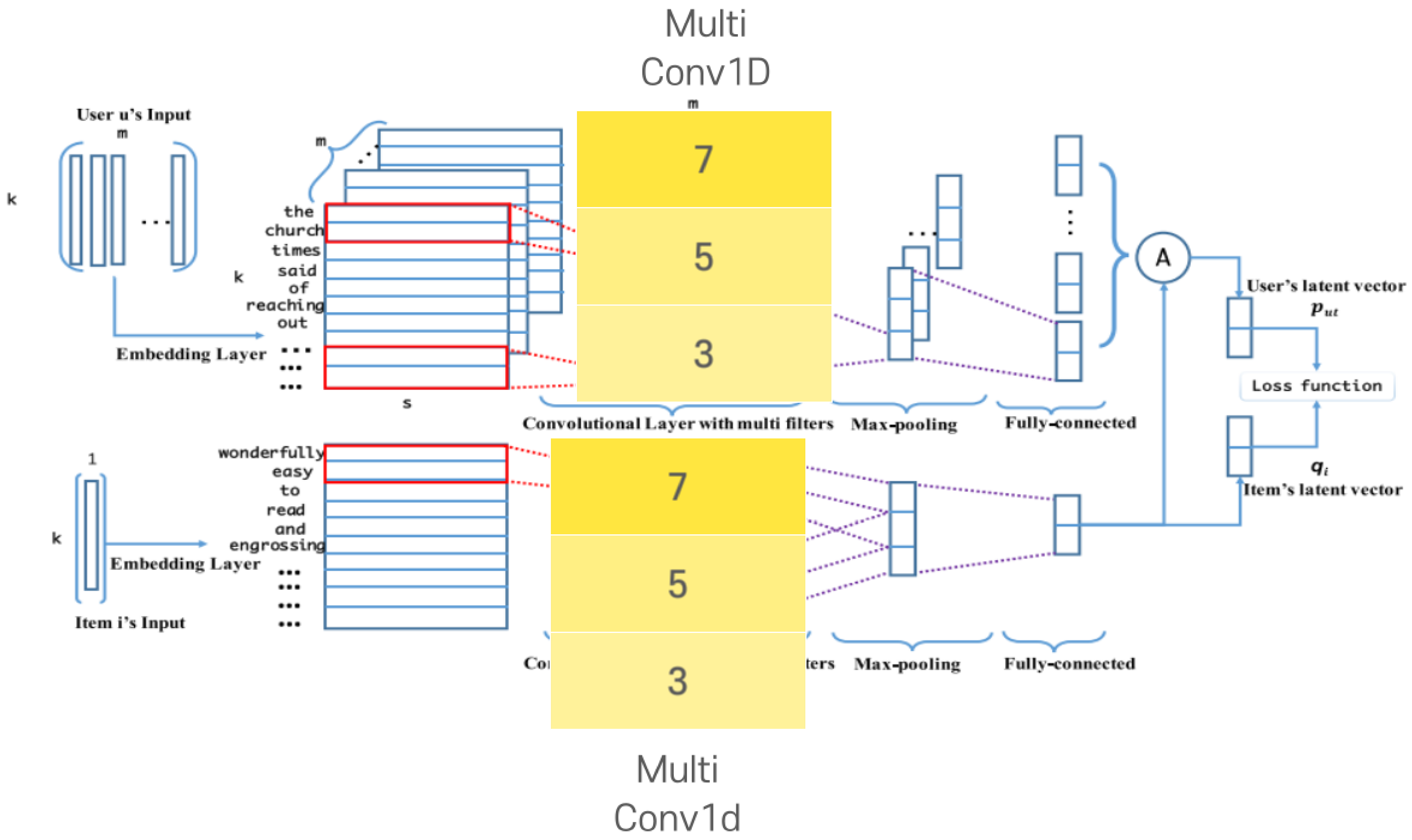
" Multi-Head Conv1D "



Multi Head Module(Keras) 활용을
통한 다양한 관점에서의 Conv1D 적용

Experiment 6

" Multi Conv1D "



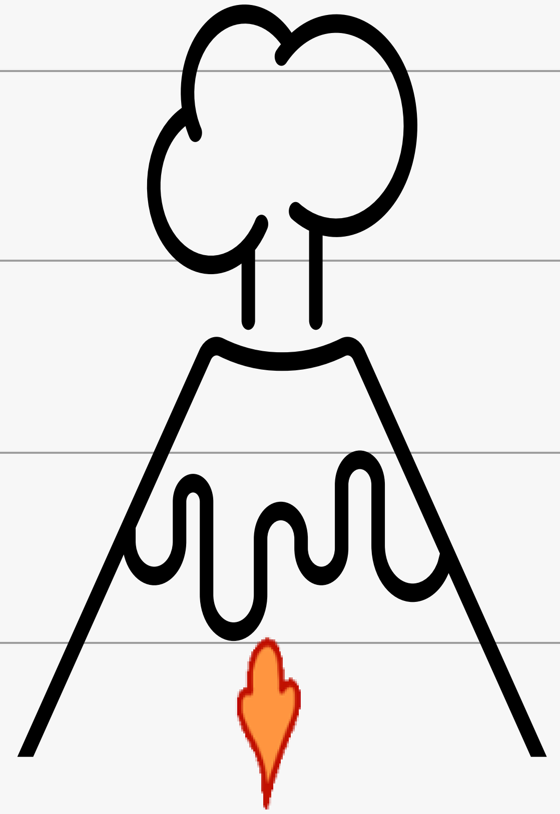
Kernel Size 조절을 통한

Global 정보와 Local 정보 결합

Experiment (Description Num 5)

(MSE)

	Item Description	Item Name	Item Id
기존 ADLFM	2.2380	2.2444 (7s)	2.5767 (2.7s)
Self-Attention		2.2509 (48s)	2.6150 (3s)
Residual Connection		2.3436 (17s)	4.1272 (3.1s)
Transformer Encoder		2.2390 (3s)	2.9123 (3.1s)
Multi-Head Conv1D		2.1936 (9s)	2.6511 (2.7s)
Multi Conv1D		2.28	2.2814 (2.9s)



Experiment (Description Num 4)

(MSE)

	Item Description	Item-Id
기존 ADLFM	2.2610 (92s)	2.5605 (3s)
Self-Attention		2.5743 (3s)
Residual Connection	3.0919 (150s)	4.2360 (3s)
Transformer Encoder		2.9773 (3s)
Multi-Head Conv1D	2.2778 (92s)	2.6431 (2.9s)
Multi Conv1D	2.2765 (153s)	2.2761 (3.12s)

Experiment (Description Num 3)

(MSE)

	Item Description	Item-Id
기존 ADLFM	2.2327 (49s)	2.6855 (2s)
Self-Attention	💣	2.6776 (2s)
Residual Connection	2.3866 (60s)	3.8914(3s)
Transformer Encoder	💣	3.1238 (4s)
Multi-Head Conv1D	2.2262 (49s)	2.7648 (6s)
Multi Conv1D 👑	2.2275 (108s)	2.3098 (3s)

결론

1. Description Num을 줄여도 성능 Same~

2. Multi-Conv1D 적용 성능 Good!

앞으로의 방향성

1. 여러 종류의 데이터셋에 적용 및 비교

2. 종강까지 화이땡!

Appendix

" IMDB Data Experiment "

	Item Description
기존 ADLFM	6.7655 (27s)
Self-Attention	
Residual Connection	8.0920 (33s)
Transformer Encoder	
Multi-Head Conv1D	6.7192 (27s)
Multi Conv1D	6.7016 (62s)

Thank You.

AI빅데이터프로젝트 : 캡스톤디자인 I

아무래도 그렇조
20172848 이지평
20172853 장성현
20192761 김정하