

시빅데이터프로젝트 : 캠프스톤디자인 I

# G-ADLFM

아무래도 그렇조 | 20172848 이지평  
20172853 장성현  
20192761 김정하



# CONTENTS

**01.**

**Introduction**

**02.**

**Related Work**

**03.**

**Proposal**

**04.**

**Experiment**

**05.**

**Conclusion**

# 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

# 01.

## Introduction

Item Description을 포함한 데이터셋이 굉장히 드물기 때문에  
Item Description이 아닌 Item Id를 Input으로 넣으면 일반화 효과,  
즉 적용가능한 데이터셋의 범위를 상당히 넓힐 수 있지 않을까 기대.

→ Input으로 Item Description을 넣은 모델과 Item Id를 넣은 모델의 비교

모델의 학습 속도 및 성능을 유지하는 것이 목표

→ 다양한 모델 구조를 여러 데이터셋에 적용하여 실험 진행

## 02.

### Related Work

#### Key Paper : ADLFM(Adaptive Deep Latent Factor Model)

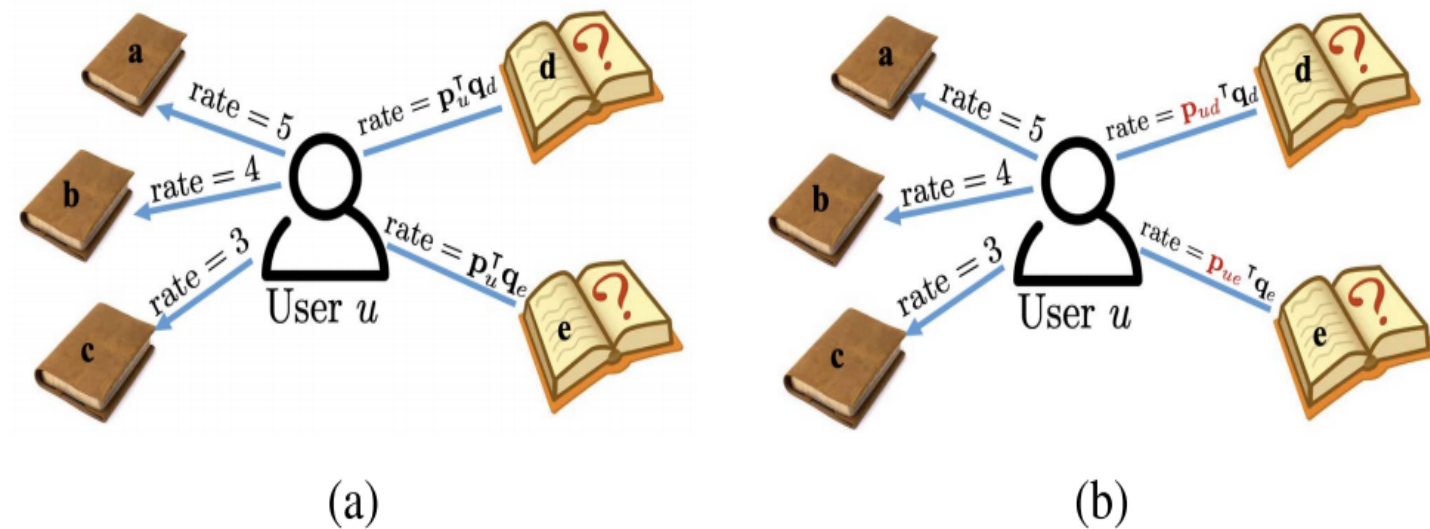


Fig. 2. Comparison between previous models and our model on how to construct users' preference representation. (a) Previous models: without individual diversity. (b) Our model: with individual diversity.

기존 추천 시스템은 사용자 선호도 및 Item Features를 모델링하는 데 행렬 분해법(Matrix Factorization)이 널리 적용

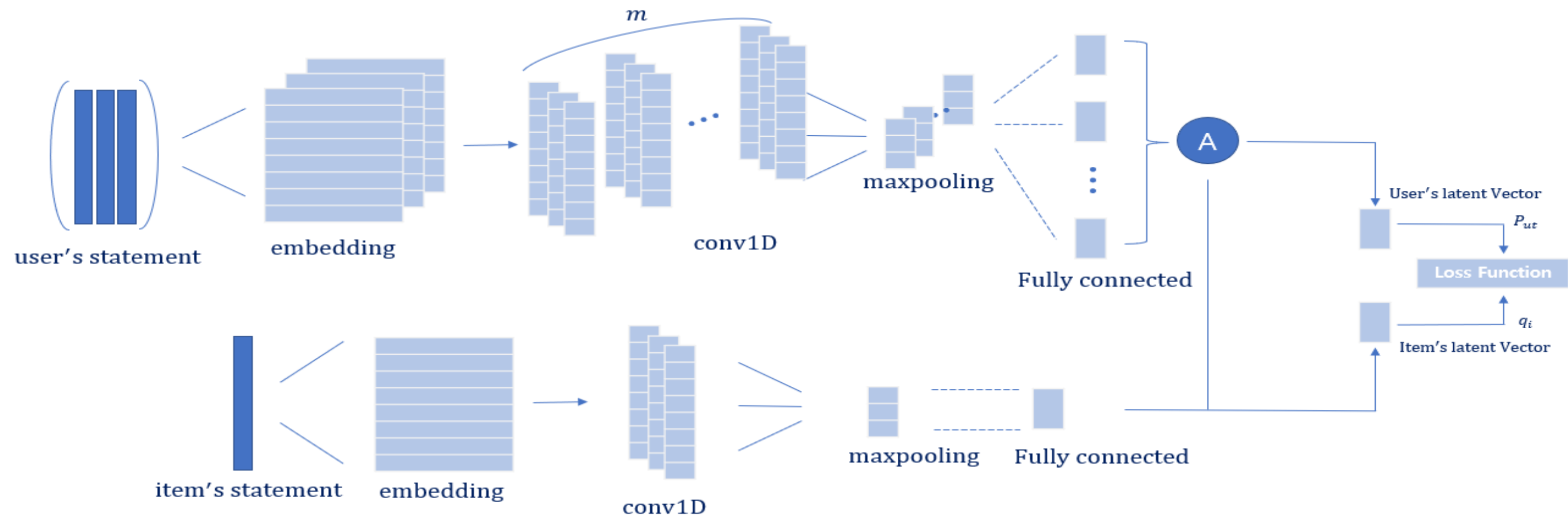
- 등급이 지정되지 않은 항목에 대한 사용자의 선호도가 다른 '개인의 다양성(Individual Diversity)'을 무시
- 개인의 다양성을 잘 모델링하지 못하여 반복적이고 부정확한 추천을 초래하게 된다.

이를 위해 해당 논문에서는 고려 중인 특정 항목에 따라 사용자의 선호도를 적응적으로 학습하는 ADLFM(Adaptive Deep Latent Factor Model)이라는 새로운 잠재 요인 모델을 제안했다.

## 02.

### Related Work

Key Paper : ADLFM(Adaptive Deep Latent Factor Model)



## 02.

### Related Work

성능 향상을 위해 사용된 모델

01.

Self-Attention

02.

Multi-Head  
Conv1D

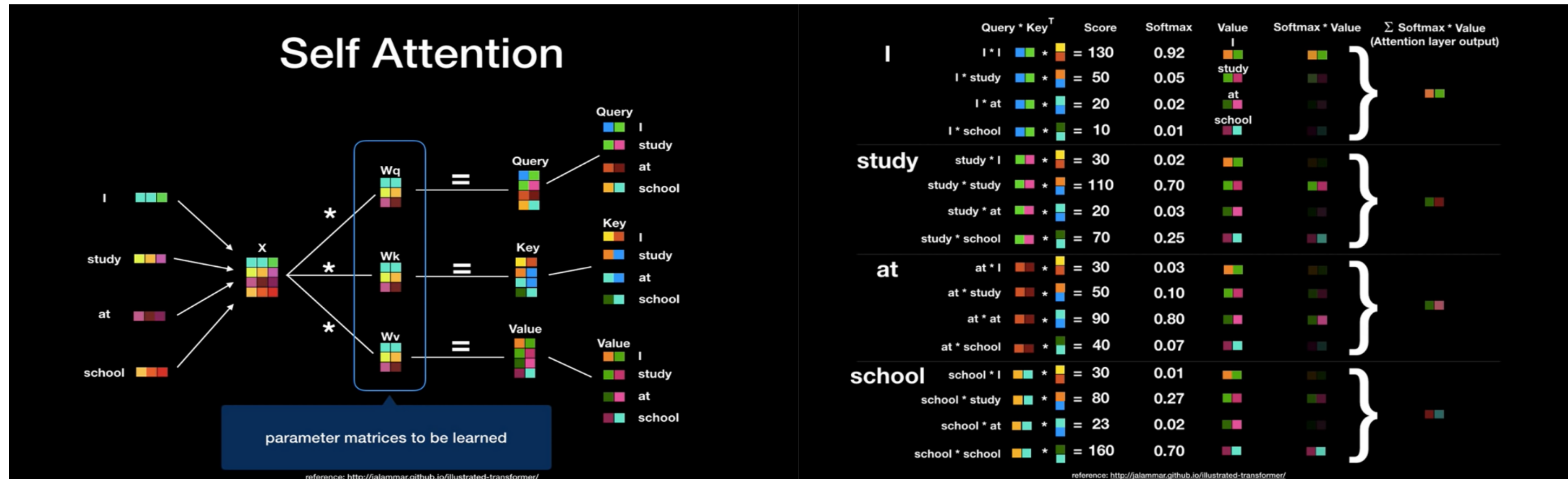
03.

Multi-Conv1D

# 02.

## Related Work

### 성능 향상을 위해 사용된 모델 1 : Self-Attention



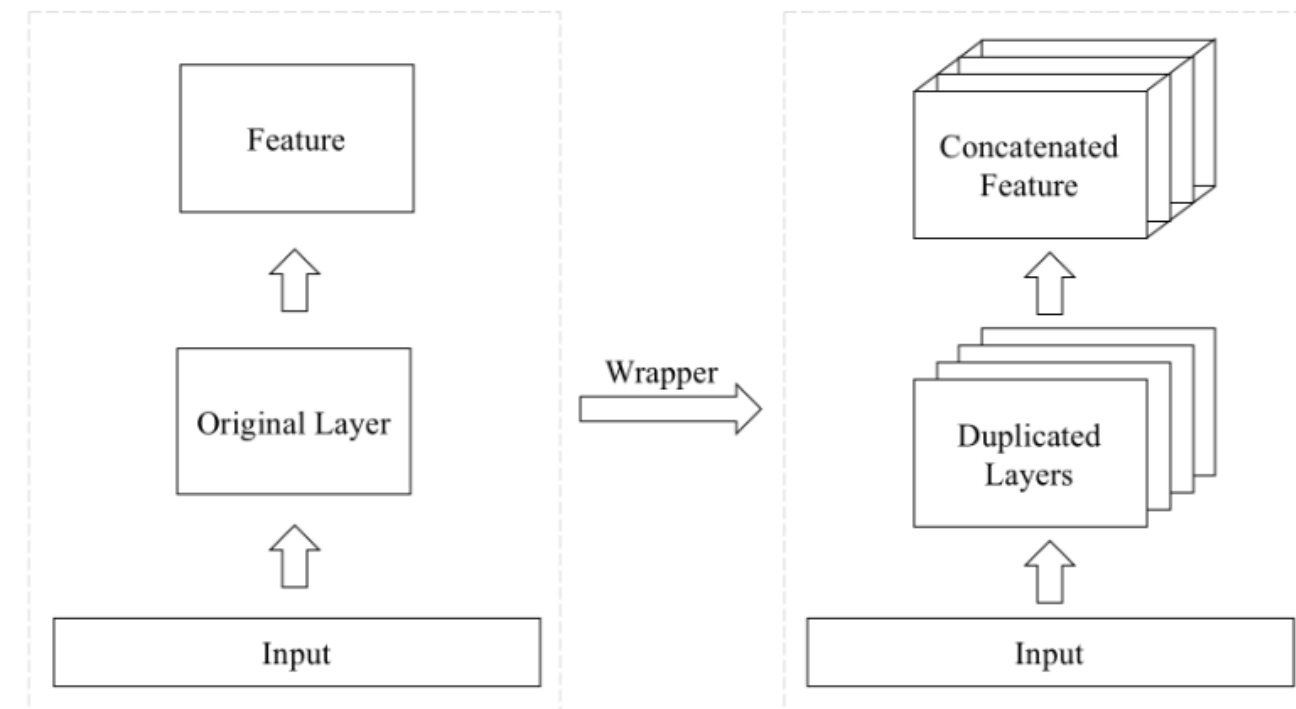
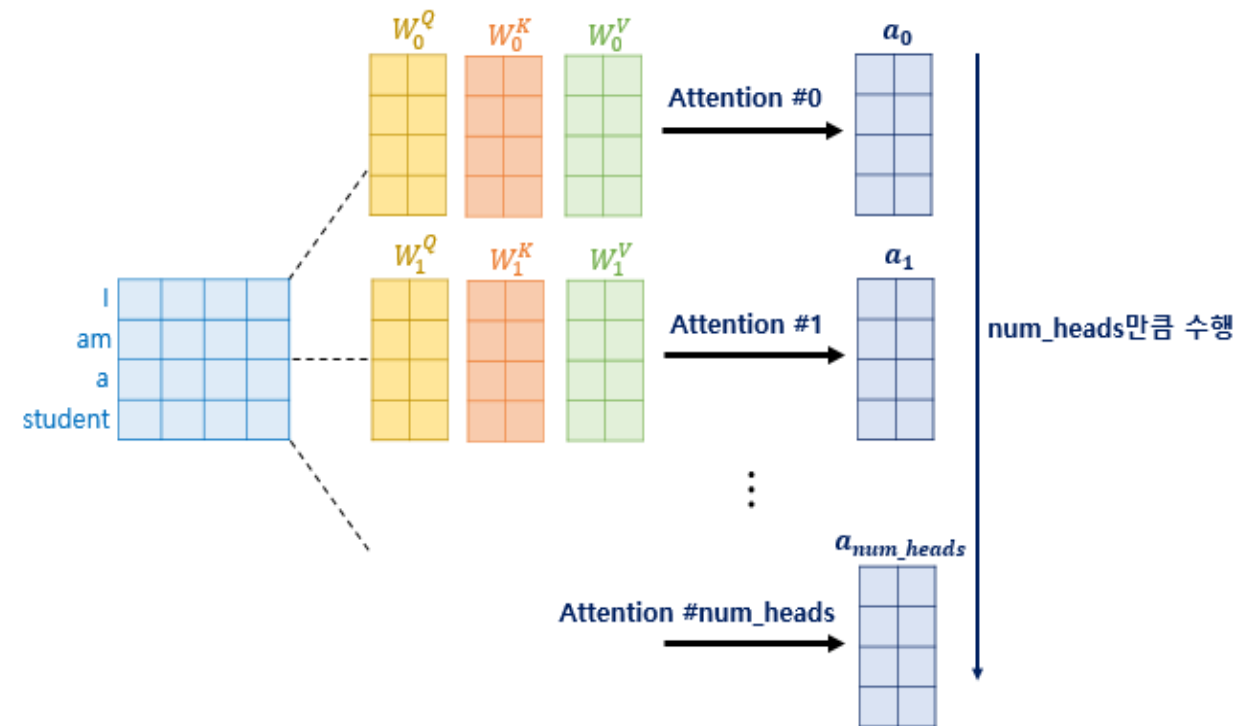
→ Embedding 강화시키는 목적



## 02.

### Related Work

#### 성능 향상을 위해 사용된 모델 2 : Multi-Head Conv1D



Ex) 그 동물은 길을 건너지 않았다. 왜냐하면 그것은 너무 피곤하였기 때문이다.

Conv1D를 병렬로 수행하여 다른 시각으로 정보들을 수집함으로써 성능 향상 도모

## 02.

### Related Work

#### 성능 향상을 위해 사용된 모델 3 : Multi-Conv1D



기존 : Conv1D layer에서 Kernel Size가 3인 모델구조를 사용 → 상대적으로 Local적인 정보만을 반영

→ Local + Global 정보를 반영하고자 함

Multi-Conv1D : Kernel Size 3, 5, 7인 Conv1D Layer를 각각 사용함으로써 Local + Global 정보 반영

# 03.

## Proposal

Input의 변경(Item Description → Item Id)으로 인한 정보의 손실!!!

우리는 성능 향상과 경량화 둘 다 잡는 것이 목표!

→ 단 하나의 레이어로 성능 향상 시킬 수 있는 모델 구조 선택

04.

Experiment

01.

Data Processing

02.

Modeling

03.

Experiments

실험환경 : Google Colab(Pro)

Optimizer : Adam

Filter\_num : 64

Epochs : 30

Batch\_size : 32

04.

Experiment

DATA PROCESSING

Anime Recommendation Database 2020



	user_id	anime_id	rating
0	0	402	9
1	0	907	5
2	0	2740	7
3	0	534	7
4	0	2539	9

anime

3 - 90408 /3 = 30,136

4 - 119700/4 = 29,925

5 - 151070/5 = 30,214

row : 57,633,278

Amazon - Ratings (Beauty Products)



	UserId	ProductId	Rating
0	A39HTATAQ9V7YF	0	5.0
1	A3JM6GV9MNOF9X	1	3.0
2	A1Z513UWSAAO0F	1	5.0
3	A1WMRR494NWEVV	2	4.0
4	A3IAAVS479H7M7	3	1.0

beauty

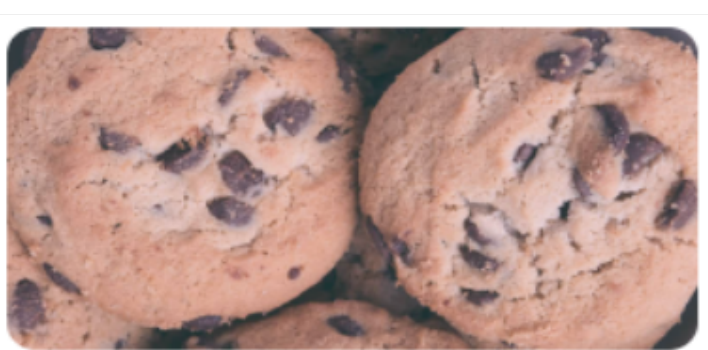
3 - 193008/3 = 64,336

4 - 225196/4 = 56,299

5 - 210270/5 = 42,054

row : 2,023,070

Amazon Fine Food Reviews



	UserId	ProductId	Score
0	A3SGXH7AUHU8GW	27619	5
1	A1D87F6ZCVE5NK	72383	1
2	ABXLMWJIXXAIN	15267	4
3	A395BORC6FGVXV	19718	2
4	A1UQRSCLF8GW1T	69007	5

food

3 - 130524/3 = 43,508

4 - 129796/4 = 32,449

5 - 114450/5 = 22,890

row : 568,454

# 04.

## Experiment

### DATA PROCESSING

#### Hold-Out Evaluation

- Train, Valid, Test 셋을 구분하여 평가하는 방식(seed 42 고정)
- 구매 이력 Reference\_num 미만인 유저 학습 데이터에서 제외.
- 구매 이력 Reference\_num 초과인 유저 Reference\_num만큼 랜덤 추출

# 04.

## Experiment

### MODELING

1. Side Information → Item Id

( Item Description → Item Id )

2. New Architecture

2-1) Self-Attention

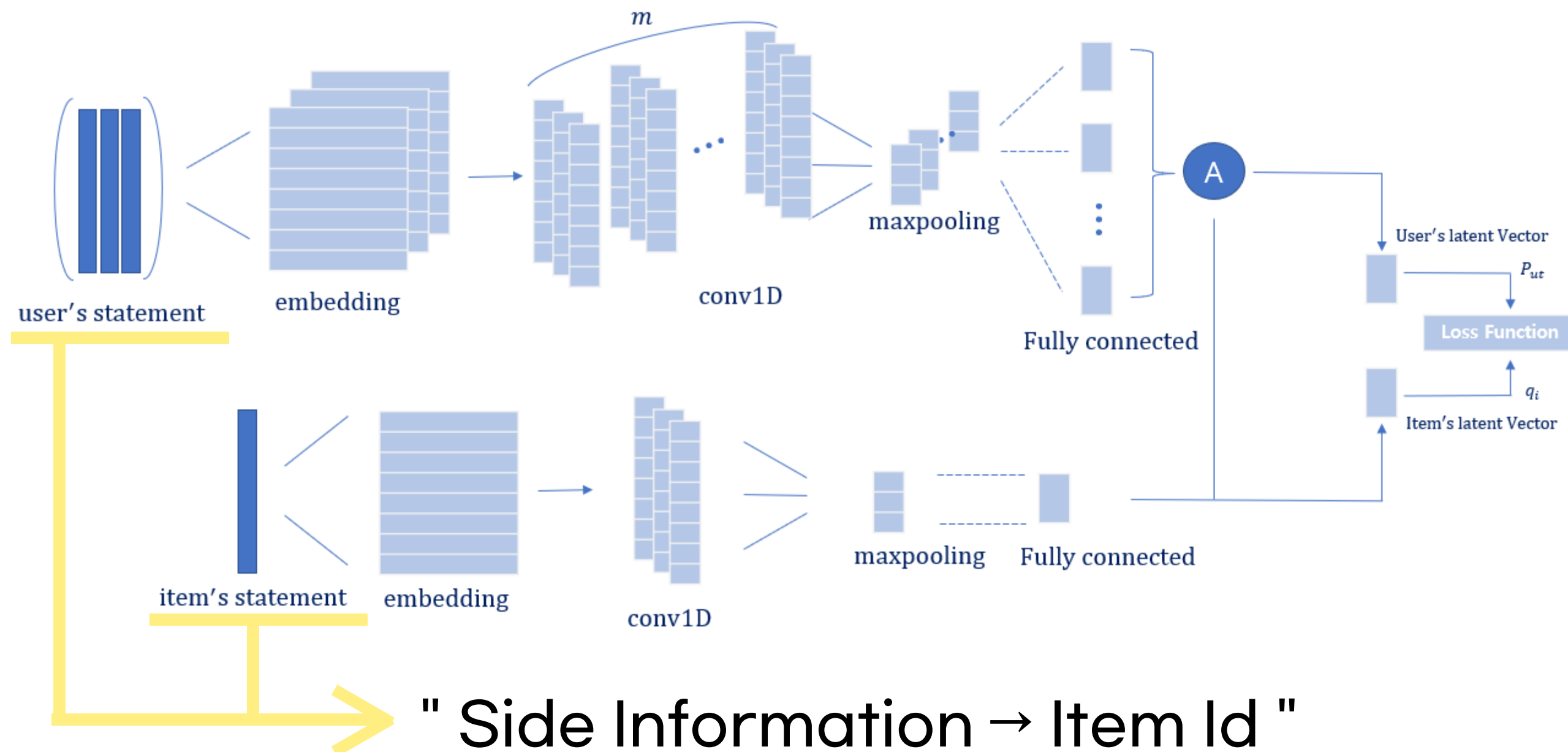
2-2) Multi-Head Conv1D

2-3) Multi Conv1D

# 04.

## Experiment

### MODELING 1



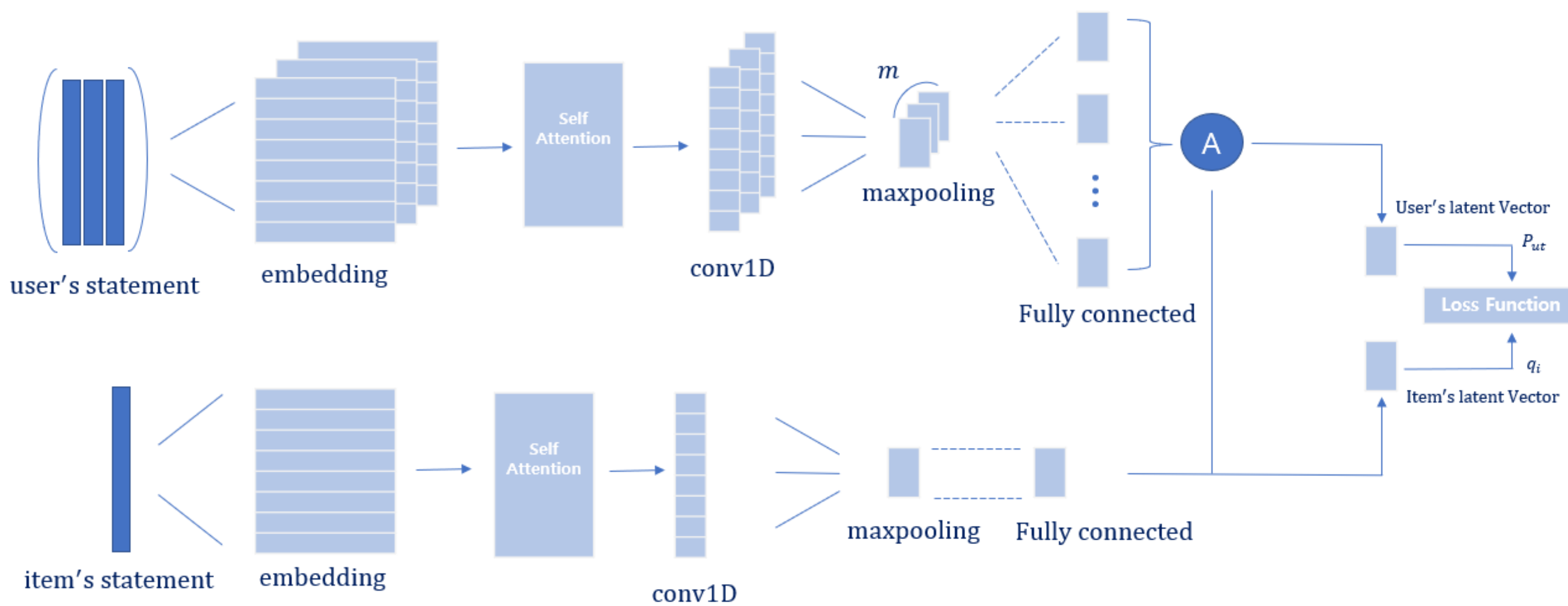


# 04.

## Experiment

### MODELING 2-1)

" Self-Attention "

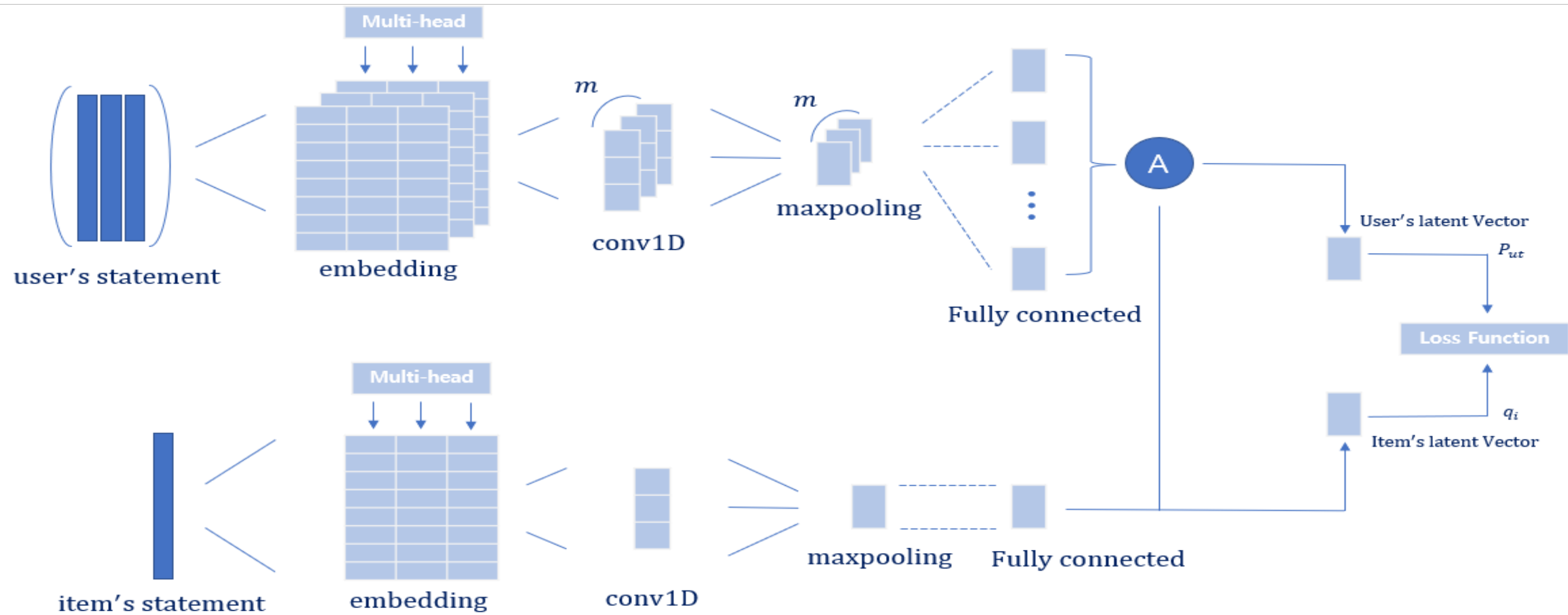


# 04.

## Experiment

### MODELING 2-2)

#### "Multi-Head Conv1D"

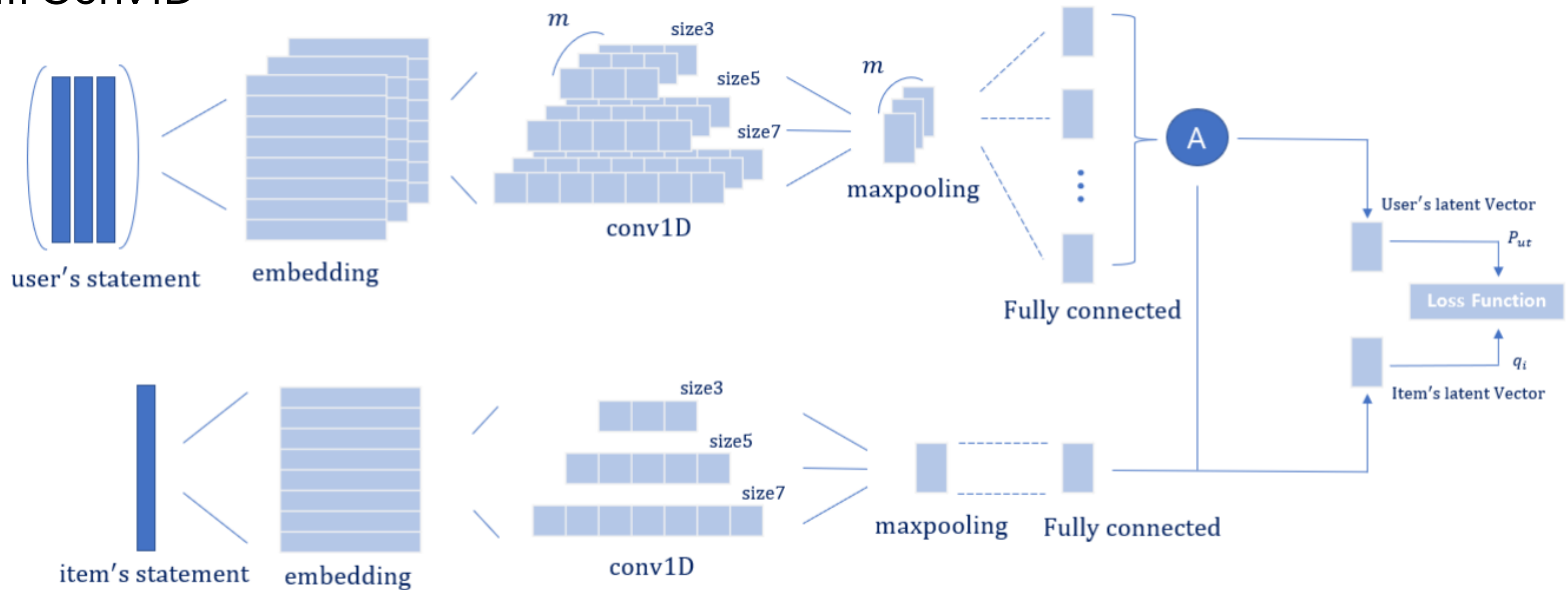


# 04.

## Experiment

### MODELING 2 - 3 )

" Multi Conv1D "



# 04.

## Experiment

### EXPERIMENTS

1. Side Information → Item Id  
( Item Description → Item Id )



1. Input을 Item Description에서 Item Id로 바꿔서  
적용했을 때의 경량화 효과 및 일반화 확인

2. New Architecture

2-1) Self-Attention

2-2) Multi-Head Conv1D

2-3) Multi Conv1D



2. 다양한 모델 구조의 성능 향상 효과 검증

3. 다양한 데이터셋에 대한 일반화 효과 검증

# 04.

## Experiment

### EXPERIMENTS 1

#### 기존 ADLFM

(MAE)

Reference_num	Item Description(A)	Item Id(B)	diff(B-A)
5	1.1330 ( 157s )	1.2093 ( 2s )	0.0763
4	1.1166 ( 92s )	1.1918 ( 2s )	0.0752
3	1.1067 ( 49s )	1.1972 ( 2s )	0.0905

→ 학습속도 감소(경량화) 확인

→ Input 변화로 인한 성능 감소폭 미미(MAE 기준 평균 0.0806) → 일반화 확인

# 04.

## Experiment

### EXPERIMENTS 2

(num) : Reference\_num / (평가지표 : MAE )

Anime	Item Id ( 3 )	Item Id ( 4 )	Item Id ( 5 )
기존 ADLFM	1.1972 ( 3.1s )	1.1918 ( 2.8s )	1.2093 ( 2.7s )
Self-Attention	1.1941 ( 3.1s )	1.1905 ( 3.1s )	1.2219 ( 2.8s )
Multi-Head Conv1D	1.1822 ( 3.1s)	1.1847 ( 2.9s )	1.2147 ( 2.9s )
Multi Conv1D	1.1102 ( 3.1s )	1.1045 ( 3.1s )	1.1390 ( 3.1s )

- 다양한 모델구조에 대한 성능 향상 확인
- Multi Conv1D 구조에서 제일 좋은 성능 보임 확인

# 04.

## Experiment

### EXPERIMENTS 3

→ 다양한 데이터셋에 대한 성능 차이 미미  
→ 일반화 효과 검증

[ 데이터셋 - 평균 성능 ]

Anime : 1.17834

Beauty : 1.0274

Food : 0.91873

(num) : Reference\_num / (평가지표 : MAE )

Beauty	Item Id ( 3 )	Item Id ( 4 )	Item Id ( 5 )
기존 ADLFM	1.0522 ( 4.5s )	1.0514 ( 3.1s )	1.0546 ( 3.1s )
Self-Attention	1.0595 ( 4.5s )	1.0535 ( 3.5s )	1.0407 ( 3.1s )
Multi-Head Conv1D	1.0137 ( 4.5s )	1.0083 ( 3.1s )	1.0069 ( 3.1s )
Multi Conv1D	1.0063 ( 5.2s )	0.9745 ( 3.5s )	1.0072 ( 3.1s )

Food	Item Id ( 3 )	Item Id ( 4 )	Item Id ( 5 )
기존 ADLFM	0.9133 ( 3.1s )	0.8946 ( 2.7s )	0.9393 ( 2.6s )
Self-Attention	0.9202 ( 3.5s )	0.8951 ( 2.8s )	0.9401 ( 2.7s )
Multi-Head Conv1D	0.9168 ( 3.1s )	0.8911 ( 2.9s )	0.9391 ( 2.7s )
Multi Conv1D	0.9195 ( 3.5s )	0.8994 ( 2.9s )	0.9563 ( 3.1s )

# 05.

## Conclusion

### 1. 일반화 & 경량화

#### 1.1) Input의 일반화

- Item Description 대신 Id값을 Input으로 사용해 Input에 제한을 받지 않는 더 광범위한 데이터셋에 적용 가능

#### 1.2) Domain의 일반화

- 다양한 도메인의 데이터셋(Anime, beauty, food)에서 비슷한 성능이 도출되어 데이터셋 일반화가 가능함을 확인

#### 1.3) Model의 경량화

- Input의 크기(용량)가 1로 줄어듦과 동시에 최소한의 모델 구조 수정을 통해 경량화 효과 도모

### 2. 일반화로 저하된 성능을 완화시켜주는 모델 구조 제안

Self-Attention

Multi-Head Conv1D

Multi-Conv1D 



# REFERENCE

[ ADLFM ]

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8736041&tag=1>

[ Self-Attention ]

<https://arxiv.org/abs/1706.03762>

[ Multi-head ]

<https://pypi.org/project/keras-multi-head/>

[ Data ]

<https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database>

<https://www.kaggle.com/datasets/skillsmuggler/amazon-ratings>

<https://www.kaggle.com/datasets/snap/amazon-fine-food-reviews>

시빅데이터프로젝트 : 캡스톤디자인 I

# 감사합니다.

아무래도 그렇조  
20172848 이지평  
20172853 장성현  
20192761 김정하