

문서명 README.pdf		작성일 2023.02.22.	전체 페이지수 15
소 속 복합지능연구실	작성자 김현우	제 목 챌린지 베이스라인 설명	문서관리자 송화전

챌린지 베이스라인 설명

서브태스크4 – 제로샷학습

목 차

1. 문서의 목적	1
2. DB	1
2.1. 패션 아이템의 메타데이터 DB	1
2.2. 학습 대화 DB	3
2.3. 제로샷학습 평가 대화 DB	4
2.4. 패션 아이템의 이미지와 특징 DB	5
3. 파일 구조	6
4. 실행	8
4.1. 실행 환경	8
4.2. 아나콘다에서 실행 환경 셋팅	8
4.3. 실행 방법	8
4.4. 옵션	9
5. 베이스라인 설명	10
5.1. 개요	10
5.2. 데이터 입출력	12
5.3. 모델	13
5.4. 학습 및 평가	13
6. 라이선스	13
7. 연락처	13

1. 문서의 목적

본 문서는 “2023 ETRI 자율성장 인공지능 경진대회”를 위한 챌린지 베이스라인(서브태스크4 - 제로샷학습)을 설명하는 문서이다.

2. DB

2.1. 패션 아이템의 메타데이터 DB

- 패션 아이템의 메타데이터(./data/mdata.wst.txt.2021.10.18)는 형태/색채/소재/감성 특징을 포함하고 그 약어는 (표 1)과 같음

특징 종류	약어
형태	F
소재	M
색채	C
감성	E

표 1. 패션 아이템의 메타데이터 특징 종류별 약어

- 패션 아이템의 종류별 개수와 약어는 (표 2)와 같음

종류	약어	개수
자켓	JK	480
코트	CT	390
점퍼	JP	450
니트	KN	408
스웨터	SW	272
셔츠	SH	193
블라우스	BL	244
가디건	CD	266
조끼	VT	132
원피스	OP	207
치마	SK	413
바지	PT	465
신발	SE	263
가방	BG	420
스카프	SC	300
모자	HC	356
합계		5,259

표 2. 패션 아이템의 종류별 개수와 약어

- 항목별 패션 아이템의 종류는 (표 3)과 같음

항목	약어	패션 아이템의 종류
겉옷(outer)	O	자켓(JK), 점퍼(JP), 코트(CT), 가디건(CD), 조끼(VT)
상의(top)	T	니트(KN), 스웨터(SW), 셔츠(SH), 블라우스(BL)
하의(bottom)	B	치마(SK), 바지(PT), 원피스(OP)
신발(shoe)	S	신발(SE)
액세서리(accessory)	A	가방(BG), 스카프(SC), 모자(HC)

표 3. 항목별 패션 아이템의 종류

- 파일 포맷은 “패션 아이템의 이름 - 항목 - 패션 아이템의 종류 - 특징 종류 - 특징 기술” 형태로 구성됨

- 예제는 (그림 1)과 같음
- 첫 번째 줄에서, "BL-001"은 패션 아이템의 이름을 말하고, "T"는 옷옷을 말하고, "BL"은 블라우스를 말하고 "F"는 형태 특징을 말하고, "단추 여밈 의 전체 오픈형"은 기술된 특징을 말함

1	BL-001	T	BL	F	단추 여밈 의 전체 오픈형
2	BL-001	T	BL	F	스탠드 칼라 와 브이넥 네크라인 의 결합 스타일
3	BL-001	T	BL	F	손목 까지 내려오 는 일자형 소매
4	BL-001	T	BL	F	여유로운 핏
5	BL-001	T	BL	F	어깨 에서 허리 까지 세로 절개 에 풍성 한 러플 장식
6	BL-001	T	BL	F	와이드 커프스
7	BL-001	T	BL	M	면 100%
8	BL-001	T	BL	M	구김 이 가 기 쉬운
9	BL-001	T	BL	M	드라이 클리닝 권장
10	BL-001	T	BL	C	시원해 보이 는 소라색 SKY BLUE
11	BL-001	T	BL	C	단색 의 깔끔한 느낌
12	BL-001	T	BL	E	여성스러운
13	BL-001	T	BL	E	페미닌한
14	BL-001	T	BL	E	세련된
15	BL-001	T	BL	E	사랑스러운
16	BL-001	T	BL	E	깔끔한
17	BL-001	T	BL	E	오피스 룩
18	BL-001	T	BL	E	로맨틱 한 데이트 룩
19	BL-001	T	BL	E	포멀한 이미지
20	BL-001	T	BL	E	단정 한 오피스 걸 룩 이미지
21	BL-002	T	BL	F	넓은 셔츠 칼라 네크라인
22	BL-002	T	BL	F	앞 중심 에 반 오픈 끈 여밈 있 는 스타일
23	BL-002	T	BL	F	드롭 숄더
24	BL-002	T	BL	F	퍼프형 소매 로 여성스러움 을 가미 함
25	BL-002	T	BL	F	세로 주름 조직
26	BL-002	T	BL	F	풍성 한 여유 가 있 는 몸통
27	BL-002	T	BL	M	면
28	BL-002	T	BL	M	세로 방향 크릴 를 조직
29	BL-002	T	BL	M	비치 는
30	BL-002	T	BL	M	얇 은
31	BL-002	T	BL	M	드라이 클리닝
32	BL-002	T	BL	C	단색 의
33	BL-002	T	BL	C	깔끔한 아이보리
34	BL-002	T	BL	E	여성스러운
35	BL-002	T	BL	E	편안한
36	BL-002	T	BL	E	복고풍 의
37	BL-002	T	BL	E	로맨틱한
38	BL-002	T	BL	E	섹시한

그림 1. 패션 아이템의 메타데이터 예제

2.2. 학습 대화 DB

- 학습 대화 DB(/data/ddata.wst.txt.2021.6.9)는 "발화번호 - <CO>|<US>|<AC> - 발화 - TAG"로 구성되어 있음
 - <AC>는 추천된 패션 코디를, <CO>코디 에이전트를, <US>는 사용자를 말함
 - TAG는 (표 4)와 같음

TAG	내용
INTRO	대화 도입부
EXP_RES_*	추천 의상 설명
USER_SUCCESS	사용자가 기술한 추천 의상 성공
USER_SUCCESS_PART	사용자가 기술한 일부 추천 의상 성공
USER_FAIL	사용자가 기술한 추천 의상 실패
FAIL	대화에서 의상 추천 실패
ASK_*	사용자가 원하는 의상유형이나 스타일, 색상 등에 대한 질문
CONFIRM_*	확인 질문
SUCCESS	대화에서 의상 추천 성공
CLOSING	대화 종료
WAIT	대기 요청
SUGGEST_*	제안 발화
NONE	의상 없음
HELP	사용자 지원

표 4. 학습 대화 DB의 TAG

- 학습 대화 예제는 (그림 1)와 같음

```

1 0 <CO> 어서 오 세 요 코디 북 입 니다 무엇 을 도와 드릴 까 요 INTRO
2 1 <US> 처음 대학교 들어가 는데 알 을 수 코디 해 주 세 요
3 2 <CO> 신의생 코디 에 어울리 게 화사 한 스웨터 를 추천_해 드릴_게 요 EXP_RES_SITUATION;EXP_RES_DESCRIPTION
4 3 <AC> SW-009
5 4 <US> 이 옷 에 어울리 는 치마 로 추천_해 주 세 요 USER_SUCCESS
6 5 <AC> SK-016
7 6 <CO> 고객 님 의 키 사이즈 에 맞추 면 이런 옷 도 잘 어울리 실 것 같_은데 어떠 신가 요 EXP_RES_BODY;CONFIRM_SATISFACTION
8 7 <US> 제 가 키 가 작_아서 짧 은 치마 로 추천_해 주 세 요 USER_FAIL
9 8 <AC> SK-052
10 9 <CO> 상의 색상 과 도 매칭 이 잘 어울리 는 짧 은 치마 입 니다 EXP_RES_COLOR;EXP_RES_LENGTH
11 10 <US> 어두운 계열 은 없 나 요 USER_FAIL
12 11 <AC> SK-053
13 12 <CO> 언밸런스 한 컷팅 으로 세련미 를 돋보이 게_하 는 치마 인데 마음 에 드 시_나 요 EXP_RES_LENGTH;EXP_RES_DESCRIPTION;CONFIRM_SATISFACTION
14 13 <US> 나쁘 지 않 네 요 외투 도 추천_해 주 시 겠_어 요 USER_SUCCESS
15 14 <CO> 요즘 계절 에는 가디건 이나 자켓 을 걸치기 에 좋_은데 특정 종류 로 원하 는 게 있_으신가 요 ASK_TYPE
16 15 <US> 트렌치 코트 종류 로 추천_해 주 세 요
17 16 <AC> CT-019
18 17 <CO> 이런 색상 과 무난 하 게 잘 어울릴 트렌치 코트 입 니다 EXP_RES_TYPE;EXP_RES_COLOR
19 18 <US> 신발 도 추천_해 주 세 요 USER_SUCCESS
20 19 <CO> 운동화 나 구두 중 어떤 걸 선호 하 시_나 요 ASK_TYPE
21 20 <US> 운동화 로 추천_해 주 세 요
22 21 <AC> SE-009
23 22 <CO> 어떤 스타일 과 도 무난 하 게 잘 어울리 는 기본 아이템 입 니다 EXP_RES_ETC
24 23 <US> 맘 에 드 네 요 전체 코디샷 볼_수 있 나 요 USER_SUCCESS
25 24 <AC> CT-019 SW-009 SK-053 SE-009
26 25 <CO> 네 지금 까지 제안 해 드린 아이템 으로 전체 코디샷 을 제안 해 드립 니다 CONFIRM_SHOW
27 26 <CO> 마음 에 드 시_나 요 CONFIRM_SATISFACTION
28 27 <US> 네 마음 에 드 네 요 감사 할 니다 USER_SUCCESS
29 28 <CO> 마음 에 드 신_다 니 다행 입 니다 SUCCESS
30 29 <CO> 이용 해 주 셔 서 감사 할 니다 CLOSING

```

그림 2. 대화 예제

2.3. 제로샷학습 평가 대화 DB

- 평가 대화 DB(/data/fs_eval_t1.wst.dev)는 “; 대화번호”, “US|CO 발화”, “R1|R2|R3 패션 코드”로 구성되어 있음
 - US는 사용자를, CO는 코디 에이전트를 말함

- R1는 1순위, R2는 2순위, R3는 3순위의 패션 코드를 말함
- 평가 대화 DB의 예제는 (그림 3)과 같음

```

1 ; 0
2 US 학원 으로 아르바이트 가 는데 편하 면서 따듯 한 코디 좀 추천_해 주 세 요
3 C0 어두운 색상 인 스트레이트 핏 바지 를 추천_해 드릴_까 요
4 US 좋 아 요
5 C0 여유 있 는 핏 의 따듯_한 겉옷 을 추천_해 드릴_까 요
6 US 좋 아 요
7 C0 편하 게 들 기 좋 은 무늬 가 있 는 가방 을 추천_해 드릴_까 요
8 US 좋 아 요
9 R1 JP-249 KN-214 PT-214 SE-040 BG-005
10 R2 JP-249 KN-214 PT-214 SE-040 BG-001
11 R3 JP-306 KN-214 PT-106 SE-040 BG-005

```

그림 3. 제로샷학습 평가 데이터의 예제

2.4. 패션 아이템의 이미지와 특징 DB

- 이미지 특징 추출 모델은 패션 아이템의 형태 특징과 감성 특징을 나누어 획득
 - 이미지 특징 추출 모델은 DeepFashion DB(230,000개)¹로 사전 훈련하고 ETRI 패션 DB(10,473개)로 미세 조정으로 획득²
 - ETRI 패션 DB는 16개 카테고리(일상성: 7개, 성: 6개, 장식성: 3개)로 구성됨
 - (그림 4)처럼 패션 아이템의 특징점 추출을 기반으로 형태 특징을 추출하는 <shape stream>과 감성 특징(일상성, 성, 장식성)들을 추출하는 <texture stream>으로 구성됨
- 학습된 이미지 특징 추출 모델을 패션 아이템의 이미지(./data/img/*.jpg)에 적용하고, 그 특징은 numpy array 파일 형태로 저장(./data/img_feats/*.npz)
 - 특징은 3개의 감성 특징과 형태 특징에 대한 2,048 사이즈의 벡터
 - Numpy array 포맷은 (형태, 일상성, 성, 장식) 순으로 (4,2048) 사이즈의 벡터들로 구성된 npy 파일

¹ Liu, Ziwei et al., "DeepFashion: Powering Robust Clothes Recognition and Retrieval with Rich Annotations", CVPR2016, 2016.

² M. Park, H. J. Song and D. -O. Kang, "Imbalanced Classification via Feature Dictionary-Based Minority Oversampling," in IEEE Access, vol. 10, pp. 34236-34245, 2022, doi: 10.1109/ACCESS.2022.3161510.

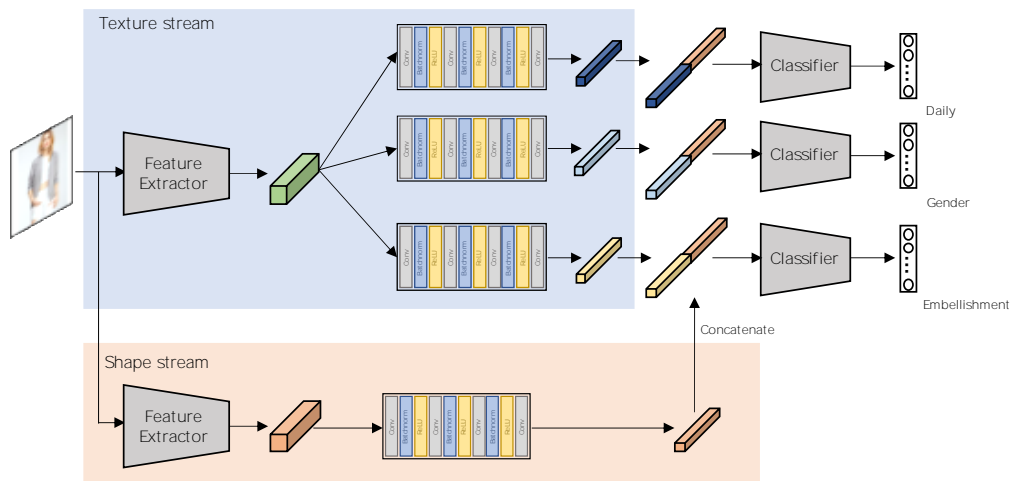


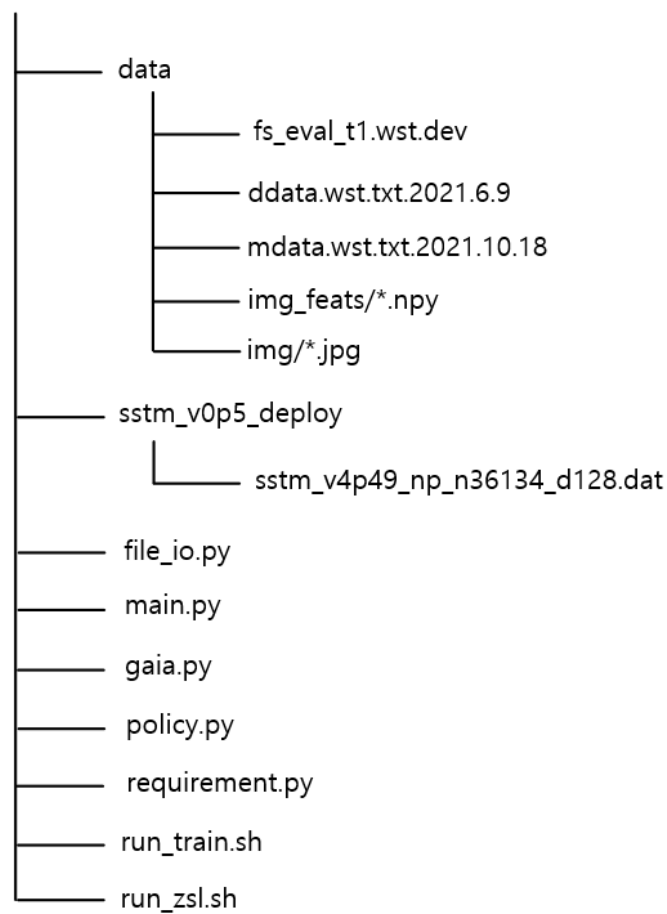
그림 4. 이미지 특징 추출 모델

3. 파일 구조

- ./data/mdata.wst.txt.2021.10.18
 - 패션 아이템의 메타데이터 DB
- ./data/ddata.wst.txt.2021.6.9
 - 학습 대화 DB
- ./data/fs_eval_t1.wst.dev
 - 제로샷학습 평가 대화 DB
- ./data/img_jpg/*jpg
 - 패션 아이템의 이미지 DB
- ./data/img_feats/*.npy
 - ETRI가 자체 개발한 모델로 추출한 패션 아이템의 이미지 특징 DB
- ./sstm_v0p5_deploy/sstm_v4p49_np_n36134_d128.dat
 - ETRI가 자체 개발한 서버워드 임베딩 DB
- ./main.py
 - 프로그램 실행 시 외부 인자를 받아서 설정하는 소스코드
- ./gaia.py
 - 설정된 외부 인자에 따라 훈련하거나 시험을 수행하는 소스코드
- ./file_io.py
 - 패션 아이템의 메타데이터 DB, 학습 및 평가 대화 DB, 이미지 특징 및 서버워드 DB를 읽어 오는 소스코드
- ./requirement.py

- 사용자의 요구사항을 추정하는 소스코드
- ./policy.py
 - 사용자의 요구사항에 적합한 패션 코디 순위를 산정하는 소스코드
- ./run_train.sh
 - 관찰되지 않은 데이터에 대한 작업을 수행할 수 있도록(제로샷 학습이 가능하도록) 학습하는 쉘스크립트
- ./run_zsl.sh
 - 제로샷학습을 평가하는 쉘스크립트

Fashion-How



4. 실행

4.1. 실행 환경

CentOS 또는 Ubuntu

Python 3.8

CUDA 10.1

CUDNN 7.6.4

PyTorch 1.8

4.2. 아나콘다에서 실행 환경 셋팅

```
conda create --name Fashion-How python=3.8
```

```
conda activate Fashion-How
```

```
conda install -c anaconda cudatoolkit=10.1
```

```
conda install -c anaconda cudnn=7.6.4
```

```
pip install torch==1.8
```

```
pip install scikit-learn tqdm
```

4.3. 실행 방법

- 관찰되지 않은 데이터에 대한 작업을 수행할 수 있도록 학습

./run_train.sh를 실행한다.

```

1 CUDA_VISIBLE_DEVICES="" python3 ./main.py --mode train ₩
2                                     --in_file_trn_dialog ./data/ddata.wst.txt.2021.6.9 ₩
3                                     --in_file_fashion ./data/mdata.wst.txt.2021.10.18 ₩
4                                     --in_dir_img_feats ./data/img_feats ₩
5                                     --subWordEmb_path ./sstm_v0p5_deploy/sstm_v4p49_np_n36134_d128.dat ₩
6                                     --model_path ./gAIA_model ₩
7                                     --req_net_type memn2n ₩
8                                     --eval_net_type tf ₩
9                                     --mem_size 32 ₩
10                                    --key_size 300 ₩
11                                    --hops 3 ₩
12                                    --eval_node [600,4000,4000] ₩
13                                    --tf_dropout 0.3 ₩
14                                    --tf_nhead 4 ₩
15                                    --tf_ff_dim 4096 ₩
16                                    --tf_num_layers 4 ₩
17                                    --epochs 2000 ₩
18                                    --save_freq 10 ₩
19                                    --batch_size 100 ₩
20                                    --learning_rate 0.0005 ₩
21                                    --max_grad_norm 20.0 ₩
22                                    --use_multimodal True ₩
23                                    --use_dropout True ₩
24                                    --use_input_mask True ₩
25                                    --eval_zero_prob 0.5 ₩
26                                    --corr_thres 0.9 ₩
27

```

- 제로샷 학습을 평가

./run_zsl.sh를 실행한다.

```

1 CUDA_VISIBLE_DEVICES="" python3 ./main.py --mode zsl ₩
2                                     --in_file_tst_dialog ./data/fs_eval_t1.wst.txt ₩
3                                     --in_file_fashion ./data/mdata.wst.txt.2021.10.18 ₩
4                                     --in_dir_img_feats ./data/img_feats ₩
5                                     --subWordEmb_path ./sstm_v0p5_deploy/sstm_v4p49_np_n36134_d128.dat ₩
6                                     --model_path ./gAIA_model ₩
7                                     --model_file gAIA-500.pt ₩
8                                     --req_net_type memn2n ₩
9                                     --eval_net_type tf ₩
10                                    --mem_size 32 ₩
11                                    --key_size 300 ₩
12                                    --hops 3 ₩
13                                    --eval_node [600,4000,4000] ₩
14                                    --tf_nhead 4 ₩
15                                    --tf_ff_dim 4096 ₩
16                                    --tf_num_layers 4 ₩
17                                    --batch_size 100 ₩
18                                    --use_multimodal True ₩
19
20

```

4.4. 옵션

- mode: "train"이면 제로샷 학습을 위한 학습 모듈이 실행되고, "zsl"이면 제로샷 학습 평가 모듈이 실행
- in_file_trn_dialog: 학습 대화 DB 파일명
- in_file_tst_dialog: 평가 대화 DB 파일명
- in_file_fashion: 패션 아이템의 메타데이터 DB 파일명
- in_dir_img_feats: 사전 학습된 네트워크를 사용하여 추출된 이미지 특징 DB 경로명

- subWordEmb_path: 서브워드 임베딩 DB 파일명
- model_path: 저장하거나 저장된 학습 모델의 경로명
- model_file: 저장하거나 저장된 학습 모델의 파일명
- req_net_type: 요구사항 추정부에 사용한 신경망 종류
- mem_size: 요구사항 추정부의 기억망에서 사용하는 메모리 크기
- key_size: 요구사항 추정부의 기억망에서 출력 크기
- hops: 요구사항 추정부의 기억망에서 사용하는 hop의 개수
- eval_net_type: 패션 코디네이션 평가부에 사용한 신경망 종류
- tf_dropout: 패션 코디네이션 평가부의 트랜스포머에서 드롭아웃 노드의 비율
- tf_nhead: 패션 코디네이션 평가부의 트랜스포머에서 멀티헤드 개수
- tf_ff_dim: 패션 코디네이션 평가부의 트랜스포머에서 순방향 신경망 노드 수
- tf_num_layers: 패션 코디네이션 평가부의 트랜스포머에서 계층 개수
- eval_node: 패션 코디 결정부의 평가망에서 노드 개수들
- zero_prob: 패션 코디 결정부의 평가망에서 드롭아웃 노드의 비율
- use_batch_norm: 패션 코디 결정부의 평가망에서 배치 정규화 기법 사용 유무
- use_dropout: 드롭아웃 기법 사용 유무
- use_input_mask: 모달리티 마스킹 사용 유무
- epochs: 학습 횟수
- save_freq: 모델의 저장 주기
- batch_size: 배치 크기
- learning_rate: 학습률
- max_grad_norm: 최대 그래디언트 값
- corr_thres: 학습 데이터를 증강할 때 패션 아이템의 대체하는 임계값
- evaluation_iteration: 입력의 순서를 바꾸어 수행할 평가 횟수
- use_multimodal: 복합 모달 입력 적용 유무

5. 베이스라인 설명

5.1. 개요

평가 데이터는 (그림 3)처럼, 적색 부분의 대화와 청색 부분의 패션 코디들로 구성된다. 본 알고리즘의 목적은 대화를 토대로 가장 적절한 패션 코디의 순위를 매기는 것이다.

기본 베이스라인의 구성도는 (그림 5)와 같다. 우선 요구사항 추정부는 기억망을 사용하여 과거 및 현재의 질의응답(대화)으로부터 시간/장소/상황에 적합한 사용자의 요구사항을 추정

한다. 여기서 기억망으로 페이스북의 종단간 기억망³을 사용한다. 다음으로 패션 코디네이션 평가부는 후보 패션 코디네이션이 추정된 요구사항에 적절하고 얼마나 어울리는지를 평가한다. 패션 아이템들간의 관계 추론에 적합한 트랜스포머⁴를 사용한다. 마지막으로 패션 코디네이션 결정부는 FC(Fully-Connected)망으로 후보 패션 코디네이션의 순위를 산정한다.

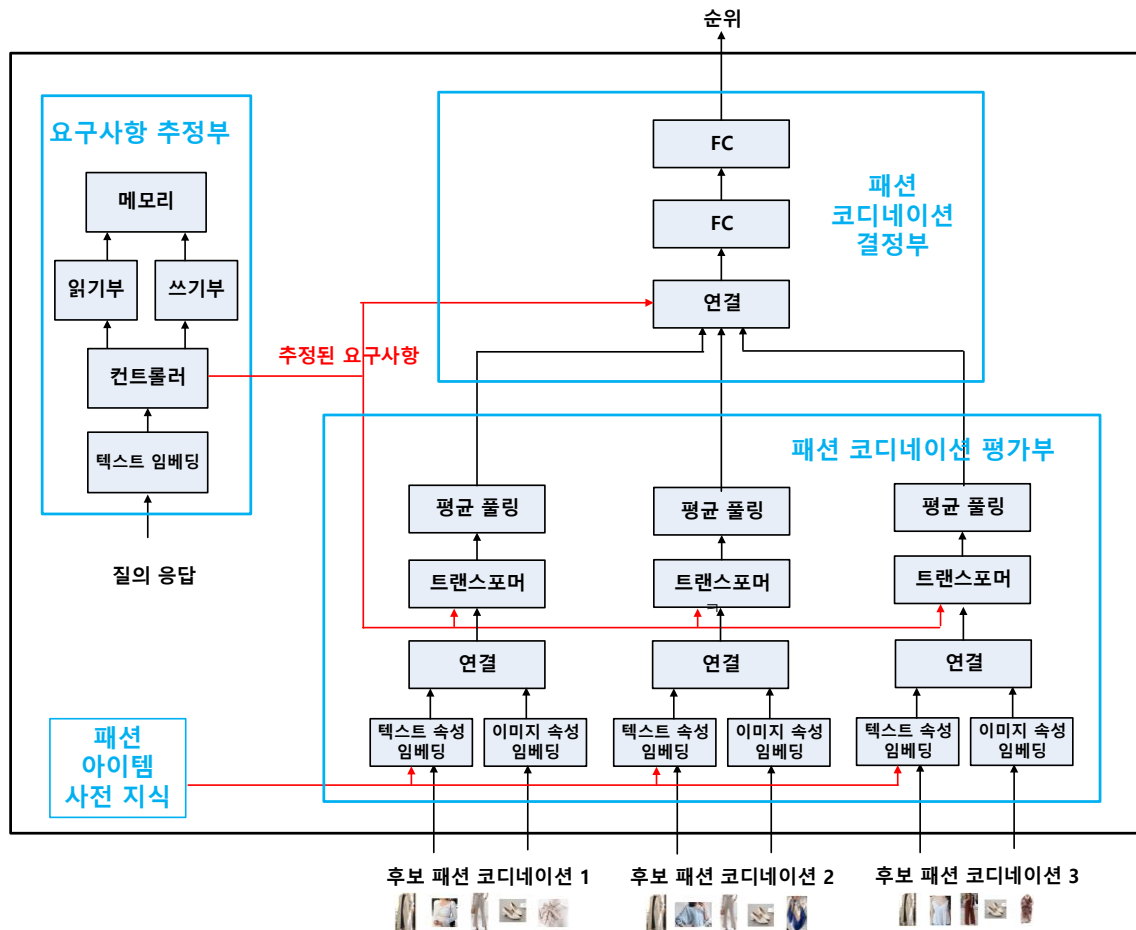


그림 5. 베이스라인 구성도

³ Sainbayar Sukhbaatar et al., "End-To-End Memory Networks", NIPS2015

⁴ Ashish Vaswani et al., "Attention Is All You Need", NIPS2017

5.2. 데이터 입출력

아래의 순서로 학습 및 평가를 위한 DB를 읽는다.

- <file_io.py>에 있는 "make_metadata" 함수를 사용하여 패션 아이템의 메타데이터 DB를 읽고 임베딩을 수행
 - <file_io.py>에 있는 "_load_fashion_item" 함수를 사용하여 패션 아이템의 메타데이터 DB를 읽음
 - <file_io.py>에 있는 "SubWordEmbReaderUtil" 클래스의 인스턴스를 생성하고, <file_io.py>에 있는 "vectorize_dlg" 함수에서 읽어 온 패션 아이템의 메타데이터에 대해 임베딩을 수행
 - "SubWordEmbReaderUtil" 클래스에서 사용하는 <./sstm_v0p5_deploy/sstm_v4p49_np_n36134_d128.dat>은 ETRI 자체 개발한 임베딩 DB임
 - <file_io.py>에 있는 "_categorize" 함수를 사용하여 패션 아이템을 항목별 분류
 - <file_io.py>에 있는 "_load_fashion_feature" 함수를 사용하여 사전 학습된 네트워크를 사용하여 추출된 패션 아이템의 이미지 특징을 읽음
 - "sklearn" 파이썬 패키지에 있는 "cosine_similarity" 함수를 사용하여 패션 아이템들간의 코사인 유사도를 계산
- <file_io.py>에 있는 "make_io_trn_data"와 "make_io_eval_data" 함수를 사용하여 학습 및 평가 대화 DB를 읽고, 학습 데이터는 평가에 적합한 형태로 변환하며 데이터를 증강
 - <file_io.py>에 있는 "_load_trn_dialog" 함수를 사용하여 학습 대화 DB를 읽음
 - <file_io.py>에 있는 "_load_eval_dialog" 함수를 사용하여 평가 대화 DB를 읽음
 - <file_io.py>에 있는 "_make_ranking_examples" 함수를 사용하여 최종 제안한 패션 코디로부터 대화의 역방향으로 중복 없이 3개의 패션 코디를 후보 패션 코디들로 결정. 또한 USER_SUCCESS가 발생한 대화의 최종 제안한 패션 코디에서 임의로 패션 아이템을 선택하고 그 아이템과 코사인 유사도가 적은 아이템으로 교체함으로써 후보 패션 코디를 생성
 - 후보 패션 코디의 순서는 임의로 배정
 - <file_io.py>에 있는 "_vectorize" 함수와 "_memorize" 함수를 사용하여 대화에 대해 임베딩을 수행하고 기억망을 위한 메모리에 저장
 - <file_io.py>에 있는 "_indexing_coordi" 함수와 "_convert_coordi_to_metadata" 함수를 사용하여 패션 코디의 아이템들을 해당하는 메타데이터로 변환
- "FashionHowDataset"("torch.utils.data"에 내장되어 있는 "Dataset"과 "DataLoader"에 기반) 클래스 생성한 후 학습 데이터를 무작위로 섞어 미니배치 단위로 처리

5.3. 모델

대화를 토대로 패션 코디 순위를 산정하기 위한 모델을 생성한다.

- 사용자의 요구사항을 추정하기 위해 <requirement.py>에 있는 "RequirementNet" 클래스의 인스턴스(모델 네트워크)를 생성
 - "RequirementNet" 클래스 내부에서 기억망을 담당하는 "MemN2N" 클래스의 인스턴스를 생성
- 대화에 가장 적합한 패션 코디의 순위를 산정하기 위해 <policy.py>에 있는 "PolicyNet" 클래스의 인스턴스(모델 네트워크)를 생성
 - 요구사항을 참조하여 후보 패션 코디를 평가 요약하기 위한 트랜스포머 생성
 - 최종 순위를 산정하기 위한 복수 계층의 FC망 생성

5.4. 학습 및 평가

- <gaia.py>에 있는 "train" 함수를 사용하여 학습 데이터를 배치 단위로 읽어 교차 엔트로피로 손실 함수를 사용하여 확률적 경사 하강법으로 학습하고 주기적으로 모델을 저장한다.
- <gaia.py>에 있는 "test" 함수를 사용하여 저장된 모델을 불러오고 성능을 측정한다.
 - "gaia.py"에 있는 "_evaluate" 함수를 사용하여 평가 데이터를 입력으로 그래프에 따라 인퍼런스를 수행하고, 그 수행된 결과에 대해 "scipy" 파이썬 패키지의 stats.weightedtau로 평균 Weighted Kendal Tau Correlation을 계산

6. 라이선스

- 본 소프트웨어는 MIT 라이선스(<https://opensource.org/licenses/MIT>)를 따라야 함
- 타 오픈소스 SW 활용 시 해당 오픈소스 SW 라이선스에서 요구하는 라이선스 준수 의무를 이행해야 함

7. 연락처

담당자: 김현우

E-mail: kimhw@etri.re.kr

끝.