

문서명 README.pdf		작성일 2021.06.08.	전체 페이지수 11
소 속 복합지능연구실	작성자 김현우	제 목 챌린지 베이스라인 설명	
		문서관리자 송화전	

챌린지 베이스라인 설명

목 차

1. 문서의 목적	1
2. DB	1
2.1. 패션 아이템의 메타데이터 DB	1
2.2. 학습 대화 DB	3
2.3. 평가 대화 DB	4
2.4. 패션 아이템의 이미지 특징 DB	4
3. 파일 구조	5
4. 실행	6
4.1. 실행 환경	6
4.2. 아나콘다에서 실행 환경 셋팅	6
4.3. 실행 방법	7
4.4. 옵션	7
5. 베이스라인 설명	8
6. 라이선스	9
7. 연락처	9

1. 문서의 목적

본 문서는 “2021 ETRI 자율성장 인공지능 경진대회”를 위한 챌린지 베이스라인을 설명하는 문서이다.

2. DB

2.1. 패션 아이템의 메타데이터 DB

- 패션 아이템의 메타데이터(/data/mdata.wst.txt)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	310
코트	CT	310
점퍼	JP	280
니트	KN	235
스웨터	SW	160
셔츠	SH	129
블라우스	BL	148
가디건	CD	150
조끼	VT	111
원피스	OP	176
치마	SK	230
바지	PT	234
신발	SE	130
합계		2603

표 2. 패션 아이템의 종류별 개수와 약어

- 항목별 패션 아이템의 종류는 (표 3)과 같음

항목	약어	패션 아이템의 종류
겉옷	O	자켓, 점퍼, 코트, 가디건, 조끼
옷옷	T	니트, 스웨터, 셔츠, 블라우스
아래옷	B	치마, 바지, 원피스
신발	S	신발

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 커푸스
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	시원 해 보이 는 소라색 SKY BLUE
11	BL-001	T	BL	C	단색 의 깔끔 한 느낌
12	BL-001	T	BL	E	여성 스러운
13	BL-001	T	BL	E	페미닌 한
14	BL-001	T	BL	E	세련 된
15	BL-001	T	BL	E	사랑 스러운
16	BL-001	T	BL	E	깔끔 한
17	BL-001	T	BL	E	오피스 룩
18	BL-001	T	BL	E	로맨틱 한 데이트 룩
19	BL-001	T	BL	E	포멀 한 이미지
20	BL-001	T	BL	E	단정 한 오피스 걸 룩 이미지
21	BL-002	T	BL	F	넓 은 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	퍼포형 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	깔끔 한 아이보리
34	BL-002	T	BL	E	여성 스러운
35	BL-002	T	BL	E	편안 한
36	BL-002	T	BL	E	복고풍 의
37	BL-002	T	BL	E	로맨틱 한
38	BL-002	T	BL	E	섹시 한

그림 1. 패션 아이템의 메타데이터 예제

2.2. 학습 대화 DB

- 학습 대화 DB(/data/ddata.wst.txt)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 기술했던 추천 의상 성공
USER_SUCCESS_PART	사용자가 기술했던 일부 추천 의상 성공
USER_FAIL	사용자가 기술했던 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 봇 입 니다 무엇 을 도와 드릴 까 요 INTRO
2 1 <US> 처음 대학교 들어가 는데 입 을 쏘 코디 해 주 세 요
3 2 <CO> 신입생 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> 연벌렌스 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> 나쁜 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 있으신가 요 ASK_TYPE
16 15 <US> 트렌치 코트 종류 로 추천_해 주 세 요
17 16 <AC> CT-019
18 17 <CO> 이너 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_나 니 다행 입 니다 SUCCESS
30 29 <CO> 이용_해 주 셔 셔 감사 할 니다 CLOSING
  
```

그림 2. 대화 예제

2.3. 평가 대화 DB

- 평가 대화 DB(/data/ac_eval_t1.wst.dev)는 “; 대화번호”, “US|CO 발화”, “R1|R2|R3 패션 코드”로 구성되어 있음
 - US는 사용자를, CO는 코드 에이전트를 말함
 - R1는 1순위, R2는 2순위, R3는 3순위의 패션 코드를 말함
- 평가 대화 DB의 예제는 (그림 3)과 같음

1	;	0
2	US	가을 축제 에 입고 갈 스타일 로 코드 해 주 세 요
3	CO	치마 바지 원피스 중 어떤 옷 이 포함_된 코드 를 추천_해 드릴_까 요
4	US	원피스 나 치마 로 추천_해 주 세 요
5	CO	가을 에 입 기 좋 은 적당_한 두께감 의 아우터 와 함께 추천_해 드릴_까 요
6	US	네
7	CO	상 의 는 무늬 없 는 티 로 추천_해 드리 는 것 은 어떠 신가 요
8	US	좋 습니다
9	R1	JP-137 KN-008 SK-047 SE-042
10	R2	JP-137 KN-045 SK-047 SE-004
11	R3	JP-137 KN-045 SK-047 SE-052

그림 3. 평가 데이터의 예시

2.4. 패션 아이템의 이미지 특징 DB

- 이미지 특징 추출 모델은 DeepFashion DB(230,000개)로 사전 훈련하고 ETRI 패션 DB(10,473개)로 미세 조정으로 획득
 - DeepFashion DB는 48개 카테고리과 1,000개 속성으로 구성됨
 - ETRI 패션 DB는 16개 카테고리(일상성: 7개, 성: 6개, 장식성: 3개)로 구성됨

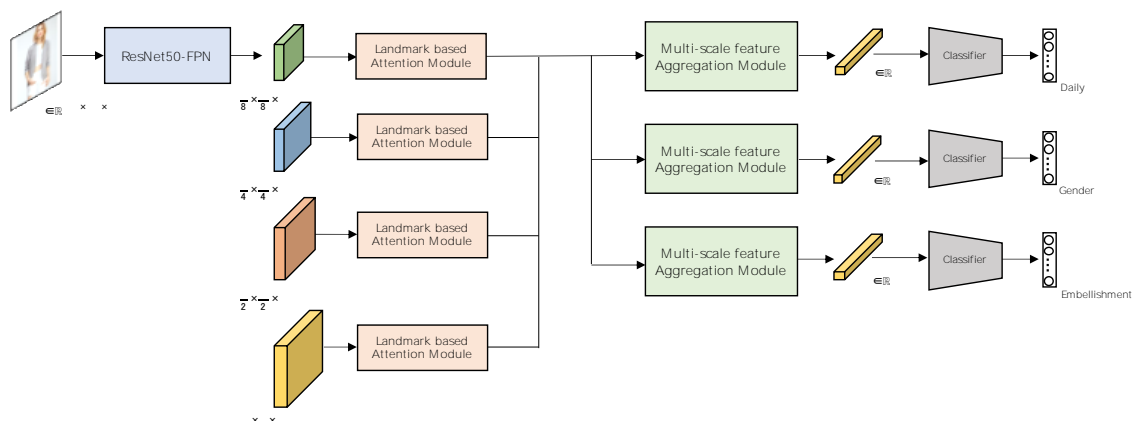
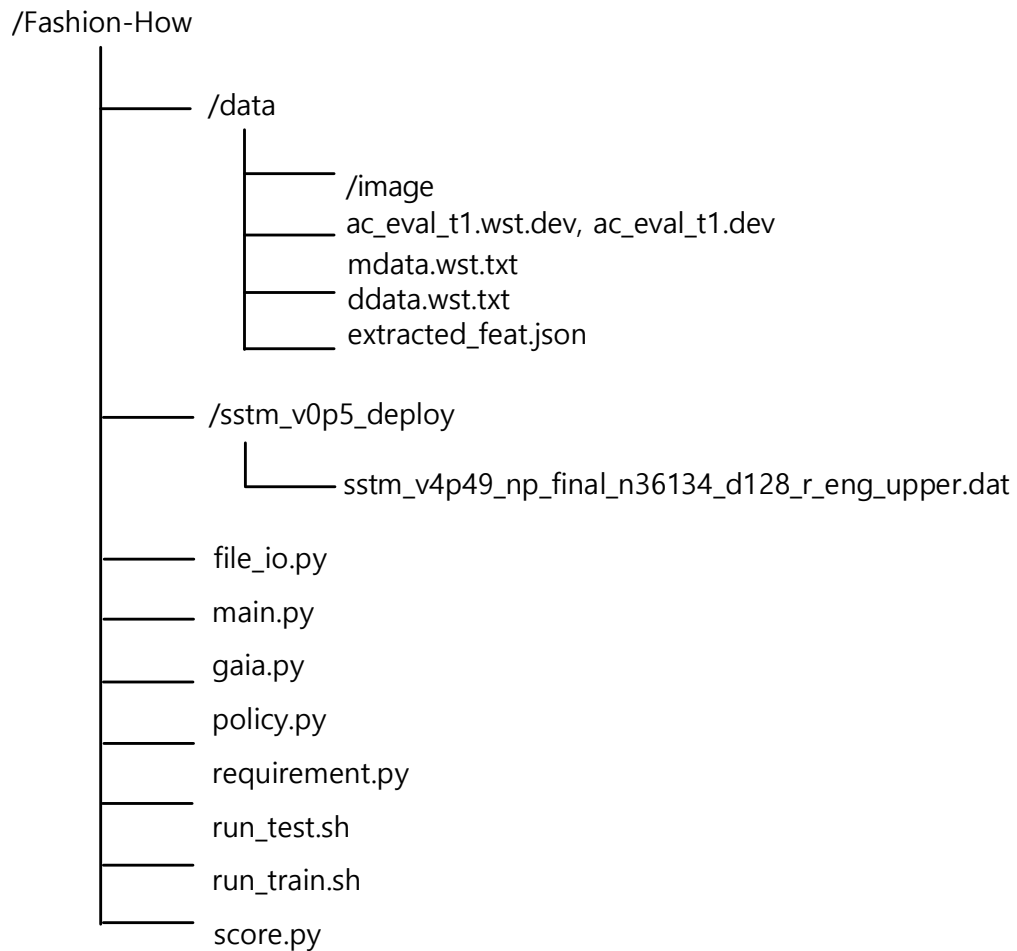


그림 4. 이미지 특징 추출 모델

- 학습된 이미지 특징 추출 모델을 패션 아이템의 이미지에 적용하고, 그 특징은 json 파일 형태로 저장
 - 특징은 3개의 감성 특징(일상성, 성, 장식)에 대한 4,096 사이즈의 벡터
 - json 포맷은 아래처럼 구성
 - '이미지 이름': 리스트 형태의 (3,4096) 차원의 벡터

3. 파일 구조



- ./data/image/ - 패션 아이템의 이미지 데이터 폴더
- ./data/mdata.wst.txt - 패션 아이템의 메타데이터 DB
- ./data/ddata.wst.txt - 학습 대화 DB

- ./data/ac_eval_t1.wst.dev(토큰 분리 버전) ac_eval_t1.dev(토큰 미분리 버전)
 - 평가 대화 DB
- ./data/extracted_feat.json
 - 사전 훈련된 네트워크로 추출된 이미지 특징 DB
- ./sstm_v0p5_deploy/sstm_v4p49_np_final_n36134_d128_r_eng_upper.dat
 - ETRI가 자체 개발한 서브워드 임베딩
- ./main.py
 - 프로그램 실행 시 외부 인자를 받아서 설정
- ./gaia.py
 - 설정된 외부 인자에 따라 훈련하거나 시험을 수행
- ./file_io.py
 - 패션 아이템의 메타데이터 DB와 학습 및 평가 대화 DB를 읽음
- ./requirement.py
 - 사용자의 요구사항을 추정
- ./policy.py
 - 사용자의 요구사항에 적합한 패션 코디 순위를 산정
- ./score.py
 - 평가에 사용되는 Weighted Kendall's tau 점수계산 코드 파일
(※ 참고용으로 제공)

4. 실행

4.1. 실행 환경

CentOS 또는 Ubuntu

Python 3.8

CUDA 10.1

CUDNN 7.6.4

PyTorch 1.8.1

4.2. 아나콘다에서 실행 환경 셋팅

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

```
conda activate Fashion-How
```

```
conda install -c pytorch cudatoolkit=10.1 cudnn==7.6.4 pytorch==1.8.1
```

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


4.3. 실행 방법

- 학습 (--mode train)

.run_train.sh를 실행한다.

(※ 윈도우 환경의 경우 main.py 상단에 cmd용 전체 실행 명령이 주석으로 포함되어 있음)

```
CUDA_VISIBLE_DEVICES="" python3 ./main.py --mode train \
    --in_file_trn_dialog ./data/ddata.wst.txt \
    --in_file_fashion ./data/mdata.wst.txt \
    --in_file_img_feats ./data/extracted_feat.json \
    --subWordEmb_path ./sstm_v0p5_deploy/sstm_v4p49_np_final_n36134_d128_r_eng_upper.dat \
    --model_path ./gAIA_model \
    --mem_size 16 \
    --key_size 300 \
    --hops 3 \
    --eval_node [6000,6000,6000,200][2000,2000] \
    --epochs 20 \
    --save_freq 1 \
    --batch_size 100 \
    --learning_rate 0.005 \
    --max_grad_norm 20.0 \
    --use_multimodal True \
    --use_dropout True \
    --zero_prob 0.5 \
    --permutation_iteration 3 \
    --num_augmentation 5 \
    --corr_thres 0.7 \
```

- 평가(--mode test) 및 결과 파일 생성(--mode pred)

.run_test.sh(평가) / .run_pred.sh(결과 생성)를 실행한다.

(※ 윈도우 환경의 경우 main.py 상단에 cmd용 전체 실행 명령이 주석으로 포함되어 있음)

```
CUDA_VISIBLE_DEVICES="" python3 ./main.py --mode test \
    --in_file_tst_dialog ./data/ac_eval_t1.wst.dev \
    --in_file_fashion ./data/mdata.wst.txt \
    --in_file_img_feats ./data/extracted_feat.json \
    --subWordEmb_path ./sstm_v0p5_deploy/sstm_v4p49_np_final_n36134_d128_r_eng_upper.dat \
    --model_path ./gAIA_model \
    --model_file gAIA-12.pt \
    --mem_size 16 \
    --key_size 300 \
    --hops 3 \
    --eval_node [6000,6000,6000,200][2000,2000] \
    --batch_size 100 \
    --use_multimodal True \
```

4.4. 옵션

- mode: "train"이면 학습 모듈이 실행되고, "test"이면 평가 모듈이 실행
"pred"로 설정하면 제출용 결과파일(prediction.csv) 생성
- 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_file_img_feats: 사전 학습된 네트워크를 사용하여 추출된 이미지 특징들 파일명
- model_path: 저장하거나 저장된 학습 모델의 경로명
- model_file: 저장하거나 저장된 학습 모델의 파일명
- eval_node: 평가망의 노드 개수들

- subWordEmb_path: 서브워드 임베딩을 위한 파일명
- learning_rate: 학습률
- max_grad_norm: 최대 그래디언트 값
- use_dropout: 드롭아웃 기법 사용 유무
- zero_prob: 드롭아웃 노드의 비율
- permutation_iteration: 동일한 학습 데이터 출력에 대해 입력의 순서를 바꾸는 개수
- num_augmentation: 학습 데이터를 증강하는 횟수
- corr_thres: 학습 데이터를 증강할 때 패션 아이템의 대체하는 임계값
- batch_size: 배치 크기
- epochs: 학습 횟수
- save_freq: 모델의 저장 주기
- hops: 기억망에서 사용하는 hop의 개수
- mem_size: 기억망에서 사용하는 메모리 크기
- key_size: 기억망의 출력 크기
- evaluation_iteration: 입력의 순서를 바꾸어 수행할 평가 횟수
- use_batch_norm: 배치 정규화 기법 사용 유무
- use_multimodal: 복합 모달 입력 적용 유무

5. 베이스라인 설명

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 기억망을 사용하여 대화로부터 사용자의 요구사항을 추정한다. 여기서 기억망으로 페이스북의 종단간 기억망(S. Sukhbaatar et al., "End-to-end memory networks," NIPS2015)을 사용한다. 다음으로 다수 계층의 FCN(Fully-Connected Network)로 구성된 평가망을 사용하여 패션 코디들의 순위를 산정한다. 각 패션 코디는 패션 아이템의 메타데이터로 임베딩을 수행하고, 사용자의 요구사항을 참조하여 3개 계층의 FCN으로 각 후보 패션 코디를 요약하며, 그 요약된 결과를 연결하여 2개의 FCN을 통해 최종 순위를 산정한다.

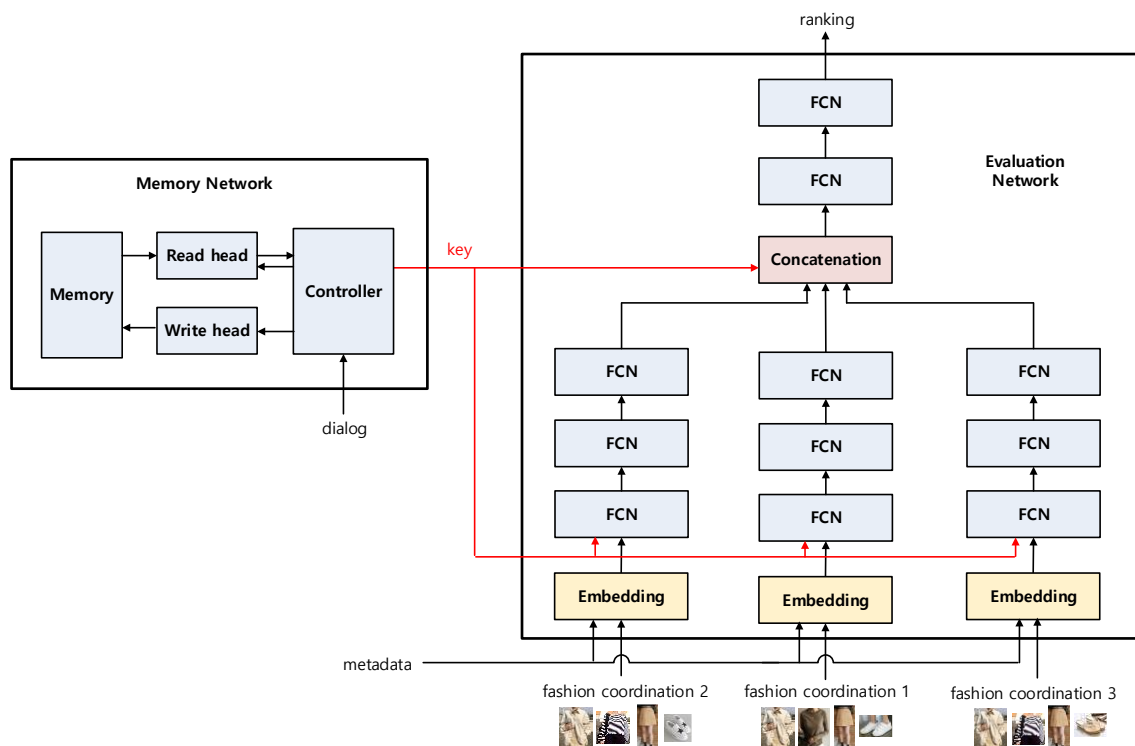


그림 5. 베이스라인 구성도

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.