korean-summarization

2020-07-01 ~ 2020-12-31까지 진행한 문서요약 프로젝트의 아카이브입니다.

Datasets

한글 데이터셋

- 신문기사
    - train: 260,697
    - dev: 10,000
    - test: 10,000
- 사설, 잡지
    - train: 53,265
    - dev: 5,000
    - test: 5,000
- 법률문서
    - train: 20,000
    - dev: 2,695
    - test: 2,694

Models & checkpoints

requirements는 최소한의 필요 패키지만 기재했습니다.
하이퍼 파라미터 및 실행 방법은 각 모델의 **학습&테스트 쉘 스크립트**를 참조해주세요.

전체적인 모델의 학습 환경은 Ubuntu 18.04 / CUDA 10.1 입니다.

Extractive Summarization

1. TextRank (paper link)
작성자의 코드 매뉴얼
프레임워크: pytorch

requirements

```
numpy==1.17.2
konlpy==0.5.2
tqdm==4.46.0
sklearn
```

소스코드

```
src/
└── extractive/
    └── textrank/
```

실행코드(학습 불필요)

```
scripts/
└── test_textrank_news.sh
└── test_textrank_magazine.sh
└── test_textrank_law.sh
```

2. SummaRunner(paper link, checkpoint)

작성자의 코드 매뉴얼

프레임워크: pytorch-lightning

requirements

```
torch==1.4.0
pytorch-lightning==1.0.2
numpy==1.17.2
konlpy==0.5.2
tqdm==4.46.0
dill==0.3.1.1
```

소스코드

```
src/
└── extractive/
    └── summarunner/
```

실행코드

```
scripts/
└── train_summarunner_news.sh
└── train_summarunner_magazine.sh
└── train_summarunner_law.sh
└── test_summarunner_news.sh
└── test_summarunner_magazine.sh
└── test_summarunner_law.sh
```

3. BertSumExt (paper link, checkpoint)

작성자의 코드 매뉴얼

프레임워크: pytorch

아래 내용은 BertSumAbs와 동일합니다.

requirement 설치 전, 아래 명령어를 통해 KoBERT 설치 필요

```
git clone https://github.com/SKTBrain/KoBERT.git
cd KoBERT
pip install -r requirements.txt
pip install .
```

requirements

```
multiprocess==0.70.9
mxnet == 1.7.0.post1
gluonnlp == 0.10.0
sentencepiece == 0.1.6
onnxruntime == 0.3.0
numpy==1.17.2
pyrouge==0.1.3
transformers==3.0.2
tensorboardX==1.9
torch==1.1.0
konlpy
```

소스코드

```
src/
└── bertsum/
```

실행코드

```
scripts/
└── train_bertsumext_news.sh
└── test_bertsumext_news.sh
└── test_bertsumext_magazine.sh
└── test_bertsumext_law.sh
```

Abstractive Summarization

1. PointerGenerator (paper link, checkpoint)
작성자의 코드 매뉴얼
프레임워크: pytorch-lightning

mecab tokenizer와 sentencepiece tokenizer를 사용하는 버전이 각각 존재하는데, mecab은 문장 생성 시 원형 단어를 복원할 수 없어 결과가 불완전합니다.
따라서 sentencepiece tokenizer 사용을 추천드립니다.

학습은 covloss를 사용하지 않고 학습한 뒤 covloss를 사용하여 조금 더 학습했습니다(논문 참고).

requirements

```
konlpy==0.5.2
gluonnlp==0.10.0
mxnet == 1.7.0.post1
onnxruntime == 0.3.0
pytorch-lightning==1.0.5
transformers==4.0.1
numpy==1.17.2
tqdm==4.46.0
tensorboard==2.4.0
sklearn
```

소스코드

```
src/
└── abstractive/
    └── pointergenerator/
```

실행코드

```
scripts/
└── train_pointer_generator_sp_news.sh
└── train_pointer_generator_mecab.sh
└── test_pointer_generator_sp_news.sh
└── test_pointer_generator_sp_magazine.sh
└── test_pointer_generator_sp_law.sh
```

2. BottomUpSummarization (paper link)

작성자의 코드 매뉴얼

프레임워크: pytorch-lightning

`bottomup` 폴더의 소스코드는 bottom-up summarization을 위해 필요한 input 파일을 생성하기 위한 BERT 모델입니다.
input 파일이 생성되면 이를 pointer-generator 모델의 input으로 추가해서 사용합니다.

- **주의 :** 실험 결과 한국어에 대해서는 bottom-up을 진행하지 않은 일반 pointer-generator가 성능이 더 잘 나옵니다. 사용 시 참고해주세요.

requirements

```
konlpy==0.5.2
gluonnlp==0.10.0
mxnet == 1.7.0.post1
onnxruntime == 0.3.0
pytorch-lightning==1.0.5
transformers==4.0.1
numpy==1.17.2
tqdm==4.46.0
tensorboard==2.4.0
sklearn
```

소스코드

```
src/
└── abstractive/
    └── bottomup/
    └── pointergenerator/
```

실행코드

```
scripts/
└── train_bottomup.sh
└── train_pointer_generator_mecab.sh
└── test_pointer_generator_sp_news.sh
└── test_pointer_generator_sp_magazine.sh
└── test_pointer_generator_sp_law.sh
```

3. BertSumAbs (paper link, checkpoint)

작성자의 코드 매뉴얼

프레임워크: pytorch

아래 내용은 BertSumExt와 동일합니다.

requirement 설치 전, 아래 명령어를 통해 KoBERT 설치 필요

```
git clone https://github.com/SKTBrain/KoBERT.git
cd KoBERT
pip install -r requirements.txt
pip install .
```

requirements

```
multiprocess==0.70.9
mxnet == 1.7.0.post1
gluonnlp == 0.10.0
sentencepiece == 0.1.6
onnxruntime == 0.3.0
numpy==1.17.2
pyrouge==0.1.3
transformers==3.0.2
tensorboardX==1.9
torch==1.1.0
konlpy
```

소스코드

```
src/
└── bertsum/
```

실행코드

```
scripts/
└── train_bertsumext_news.sh
└── test_bertsumext_news.sh
└── test_bertsumext_magazine.sh
└── test_bertsumext_law.sh
```

ROUGE Score on test set

생성 요약 및 추출 요약 모두 모델이 생성/선택한 정답 문장과 **생성요약문**과의 rouge score를 계산합니다.

Abstractive

신문기사

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
Pointer Generator	**생성요약문**	46.3	29.8	37.7
Bottom up	**생성요약문**	41.5	22.9	31.2
BertSumAbs	**생성요약문**	52.0	34.6	41.9

잡지, 기고문

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
Pointer Generator	**생성요약문**	49.4	31.1	37.8
Bottom up	**생성요약문**	40.3	18.6	27.0
BertSumAbs	**생성요약문**	54.8	36.0	42.4

법률문서

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
Pointer Generator	**생성요약문**	32.8	14.5	24.5
Bottom up	**생성요약문**	31.8	12.2	23.3

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
BertSumAbs	**생성요약문**	38.2	18.4	28.8

Extractive

주의: 추출요약은 대부분의 논문이 생성요약 정답문과의 비교를 통한 greedy selection 방식으로 정답 요약문을 설정합니다(e.g. oracle summary).
이는 존재하는 대부분의 벤치마크 데이터셋(ex. CNN/DailyMail)이 생성요약 정답문만을 제공하기 때문입니다.
그러나 본 과제에서는 추출요약 정답문도 annotator들에 의해 생성되었습니다.
TextRank와 Summarunner는 annotator가 생성한 추출 요약 정답문을 사용해 학습되었고 BertSumExt는 성능 문제로 greedy selection을 통해 선정된 추출 요약 정답문을 이용해 학습되었습니다.

신문기사

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
TextRank	**생성요약문**	36.0	19.1	25.5
SummaRunner	**생성요약문**	48.2	32.4	35.4
BertSumExt	**생성요약문**	50.8	35.6	38.0

잡지, 기고문

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
TextRank	**생성요약문**	31.6	11.2	21.0
SummaRunner	**생성요약문**	39.8	20.7	30.4
BertSumExt	**생성요약문**	38.3	19.3	28.0

법률문서

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
TextRank	**생성요약문**	55.1	37.9	44.0

Model	Reference	ROUGE-1(F1)	ROUGE-2(F1)	ROUGE-L(F1)
SummaRunner	**생성요약문**	56.2	39.0	46.4
BertSumExt	**생성요약문**	56.9	39.8	38.8