Dacon : AI야, 진짜 뉴스를 찾아줘!

Rainism

01

비즈니스이해

02

데이터 이해

03

데이터 전처리

04

모델링

05

결론

01

비즈니스이해

분석 목표

- **주가 변동**에 있어서 뉴스는 중대한 영향을 끼침
- 가짜 뉴스, 재 생성되는 뉴스들이 많음

→ **정확성 시간성 모두** 달성 필요

제공 데이터

- **Train Data :**시간, 기사 제목, 기사 내용, 가짜 뉴스 여부
- **Test Data :** 기사 내용

→ **기사 내용**으로 가짜 뉴스 **판별**

평가 기준

- **Accuracy**
- **Speed**

→ **속도**에서 **경쟁력**을 갖고자 함

02

데이터 이해

1. n_id : 뉴스 아이디
2. date : 뉴스 날짜
3. title : 뉴스 제목
4. content : 뉴스 내용
5. ord : 뉴스 내용 순서
6. info : 가짜 뉴스 여부

테스트셋에서 제공하는 데이터

뉴스 내용(content)으로 가짜 뉴스 여부를 판별하는 모델을 만듦

03

데이터 전처리

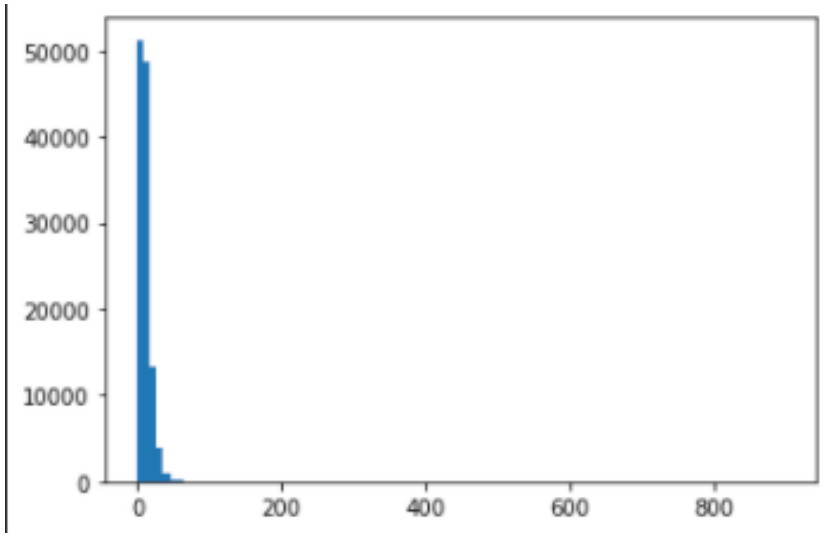
1. 결측치 없음

```
[108…  train_data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 118745 entries, 0 to 118744
Data columns (total 6 columns):
 #   Column   Non-Null Count   Dtype
---  ------   --------------   -----
 0   n_id     118745 non-null  object
 1   date     118745 non-null  int64
 2   title    118745 non-null  object
 3   content  118745 non-null  object
 4   ord      118745 non-null  int64
 5   info     118745 non-null  int64
dtypes: int64(3), object(3)
memory usage: 5.4+ MB
```

2. 하나의 내용물(content)이 갖는 단어의 분포도를 확인

→ 50개 이상의 토큰을 갖는 문장이 많지 않아 50개로 토큰 한정



03

데이터 전처리

3. Stopwords
   -> 특수문자, 한 글자(예 : 아, 어, 잘, 등), 하나의 알파벳을 제거
   -> Teanaps 패키지에 저장된 불용어 사전을 추가

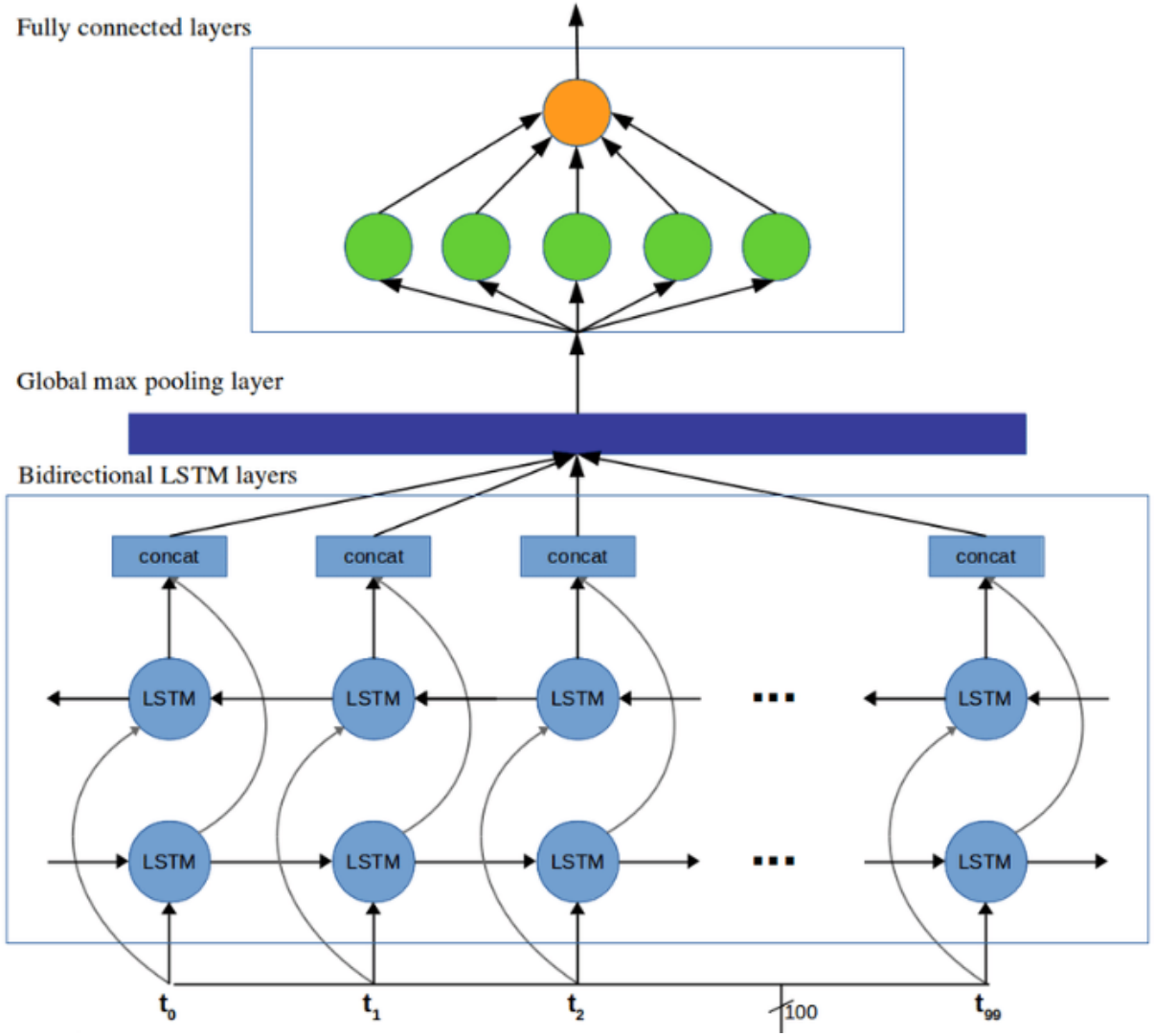
4. Tokenization
   -> 정확하고 빠른 Mecab을 사용.
   -> 명사만 사용
   -> 27,874개의 유니크한 형태소중 20,000개 사용

5. Pre-Trained-Embedding
   -> 속도 측면에서 강점을 갖기 위해 용량이 비교적 작은 Word To Vector로 학습된 Embedding을 선택
   -> Embedding의 dimension은 200으로 사용

출처 : https://github.com/Kyubyong/wordvectors

04

모델링



- 대회 리더보드 기준으로 정확도에서 경쟁력을 갖기 힘들다고 판단
- 정확도와 속도가 빠른 **Bidirectional-LSTM**모델을 사용

05

결론

분석 결과

485580	submission.csv	2020-12-17 21:26:44	0.9592652998	

경과 시간

```
print(time.time() - start)
```

```
193.83389401435852
```

Thank you