비정형데이터 분석 프로젝트 발표
(Unstructured Data Analysis)

고려대학교 산업경영공학과
Data Science & Business Analytics Lab.
11조
2017021201 송서하
2018020528 양우식
2018020516 정민성

INDEX

Advanced Word2Vec : Association Rules implemented(AWAR)

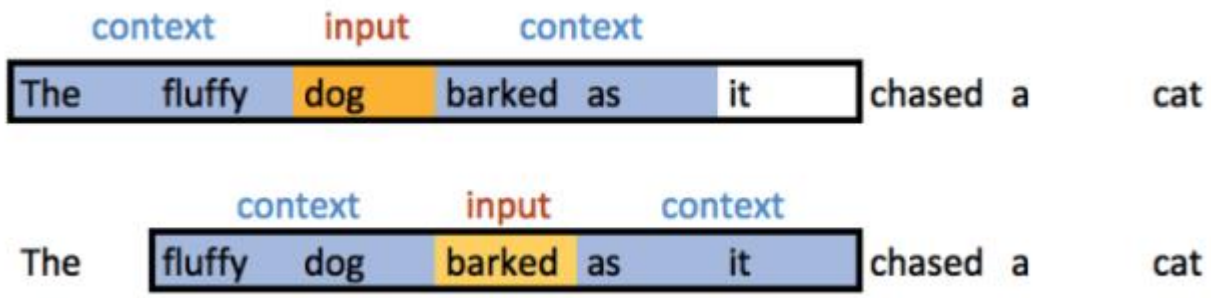
About Word2Vec

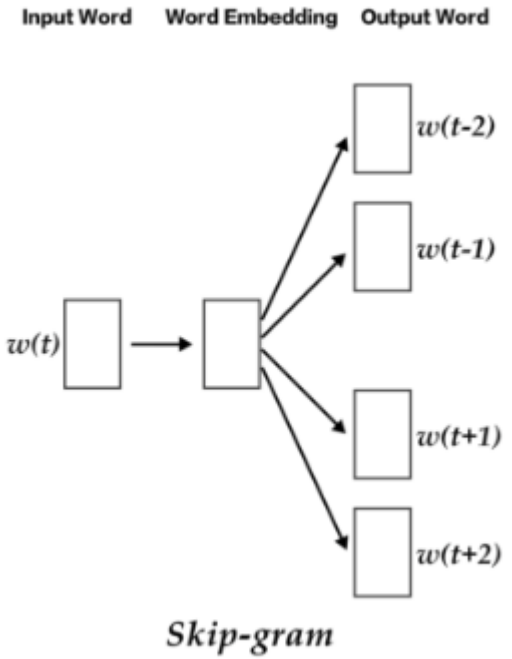
Word2Vec이란?

- 단어 임베딩(embedding) 방법론
- 단어를 벡터로 변환시키는 알고리즘이며 이를 통해 단어들 간의 의미 관계를 파악할 수 있다.

Skip-gram : Word2Vec의 한 방법론

- Center word가 주어졌을 때 주위의 Window size만큼의 output word를 예측하는 방식으로 학습





연구 배경

Approach to Skip-gram

knight → The
knight → mighty
knight → Lancelot
knight → fought
knight → bravely.

- Model probability of a context word given a word

feature for word w: x_w
classifier for word c: v_c

$$p(c|w) = \frac{e^{x_w^\top v_c}}{\sum_{k=1}^{K} e^{x_w^\top v_k}}$$

- Word vectors $x_w \in \mathbb{R}^d$

Center word와 Window size내 Context word에 대해서 모두 같은 weight

연구 배경 및 선행 연구

가정 : Window size 내에서도 center word와 output word의 관계가 반영되어야 한다.

Current Approaches taken:

- Redefining Context Windows for Word Embedding Models : An Experimental Study

  ➡ Continuous Skip-gram 기반 Window Size 내에서의 다양한 접근 시도
  1. Window size의 size
  2. center word 간의 거리
  3. center word 기준 position(왼쪽, 오른쪽, 양쪽)
  4. 언어적 처리(stop word와 같은)

- Not All Contexts Are Created Equal : Better Word Representations with Variable Attention

  ➡ CBOW with Attention :
  context window word에 attention weight를 적용시켜 학습 진행

➡ Window Size에 대해서 다른 접근 방법이 없을까?

Word2Vec using Association rules(AWAR)

Association rules?

- 장바구니 분석으로도 불리며, 품목(item)간의 집합이 빈번히 발생하는가를 알아보는데 사용하는 알고리즘
- 지지도, 신뢰도, 향상도 지표를 사용하여 규칙의 효용성을 평가

$$Rule: X \Rightarrow Y$$

$$Support = \frac{frq(X,Y)}{N}$$

$$Confidence = \frac{frq(X,Y)}{frq(X)}$$

$$Lift = \frac{Support}{Supp(X) \times Supp(Y)}$$

Rule	Support	Confidence	Lift
$A \Rightarrow D$	2/5	2/3	10/9
$C \Rightarrow A$	2/5	2/4	5/6
$A \Rightarrow C$	2/5	2/3	5/6
$B\&C \Rightarrow D$	1/5	1/3	5/9

출처:http://www.grroups.com/blog/what-the-heck-are-association-rules-in-analytics

Association rules on window size

➡ 연관성 분석을 통해 가질 수 있는 빈도 정보를 벡터공간에 반영하자

HOWEVER…

PMI (Pointwise Mutual Information)

- 상호정보량

$$\log \frac{\mathrm{P}(x,y)}{\mathrm{P}(x)\mathrm{P}(y)} = \log(\mathrm{LIFT})$$

Neural Word Embedding as Implicit Matrix Factorization(2014, NIPS)

➡ 이미 시도된 Approach ?

Neural Word Embedding as Implicit Matrix Factorization(2014, NIPS)

PMI를 활용, Word2Vec과 유사한 word representation을 얻는 방법론

1. Word-Context matrix에서 PMI를 적용

2. PMI matrix에 SVD 적용

→ Word2Vec과 유사한 word embedding공간 학습

→ Negativev Sampling을 이용하여 학습하는 Skip-gram (SGNS)가 PMI와 유사함을 수리적으로 증명

$$M_{ij}^{\text{SGNS}} = W_i \cdot C_j = \vec{w}_i \cdot \vec{c}_j = PMI(w_i, c_j) - \log k$$

❖단, PMI를 context matrix에만 한정하여 활용!

AWAR 방법론 소개

Architecture

Step 1. Lift Dictionary 생성

- 연관성 분석을 통한 transaction Dataset 설정
- Lift 값이 1 이상인 단어들 간의 집합 및 해당 lift 값 저장

Rule	Support	Confidence	Lift
Word1 -> Word3	…	…	1.5
Word4 -> Word6	…	…	2.3

<Association Rule dataset>

Word1, Word3 : 1.5

Word4, Word6 : 2.3

⋮

<Lift Dictionary>

AWAR 방법론 소개

Architecture

Step 2. Implement as Reward weight

- Window size 내 dictionary에 해당하는 단어 관계에 reward weight 도입 후 임베딩

Original skip-gram

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_I}\right)}$$

Association reward implemented

$$p(w_O|w_I) = \frac{\exp\left({v'_{w_O}}^{\top} v_{w_I}\right)}{\sum_{w=1}^{W} \exp\left({v'_w}^{\top} v_{w_I}\right)} \cdot \alpha_{W_{I,O}}$$

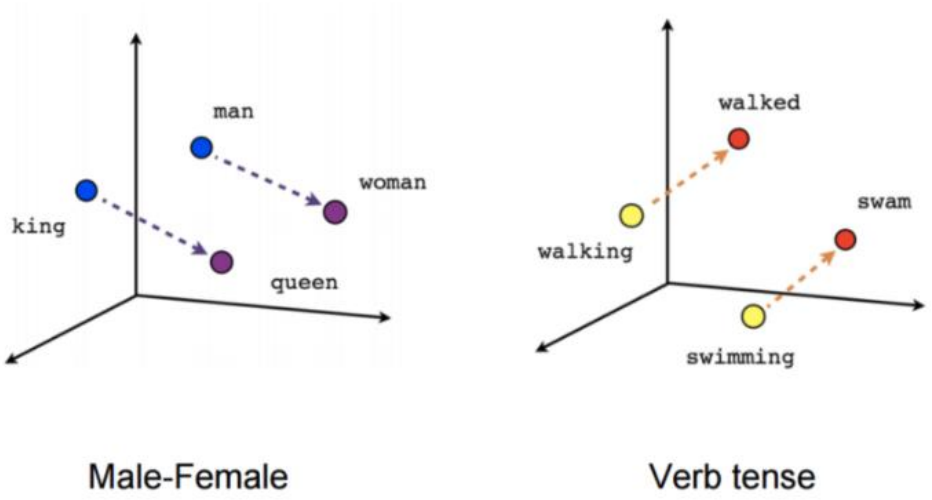
$\alpha_{W_{I,O}}$: input word와 output word가 dictionary에 있을 경우 적용되는 reward weight

AWAR 방법론 적용

Data : 10000개의 긍/부정 IMDB MOVIEW REVIEW DATASET

TASK : Association rules implemented word2vec의 representation 확인 및 긍/부정 분류 성능 확인

Visualization & Cosine similarity를
활용한 vector representation 확인



<Word2Vec visualization>

출처: https://www.tensorflow.org/images/linear-relationships.png

Sentiment Analysis 적용

- Baseline model을 기반으로 한 다양한 경우의 수의 성능 비교

AWAR 방법론 적용

연관성 분석

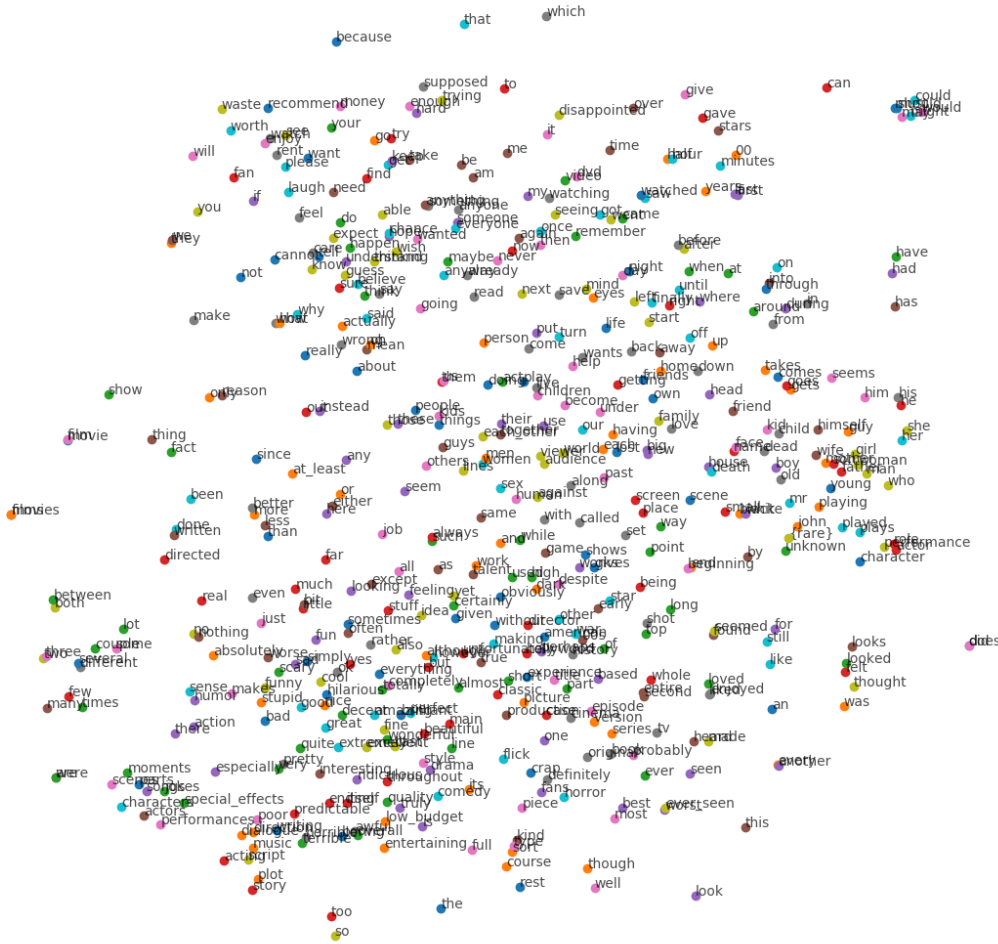
- Dataset : IMDB의 긍/부정 10000 문장
- 각 리뷰를 하나의 Transaction으로 설정 후 Transaction dataset 생성
- 연관성 분석 결과 최소 지지도(support) 0.01, 최소 향상도(Lift) 1이상인 딕셔너리 구축

Word2Vec

- Dataset : IMDB의 긍/부정 10000 문장
  = 132만개의 token corpus
- 단어 빈도의 최소 기준을 3(min_count = 3)으로 설정하여 낮은 빈도의 단어 제거
  (unique word 35403 => 20027 개로 감소)
- Window size = 5, embedding dimension =128 로 설정
- learning rate = 0.01로 설정

AWAR 방법론 적용 결과 비교-Visualization

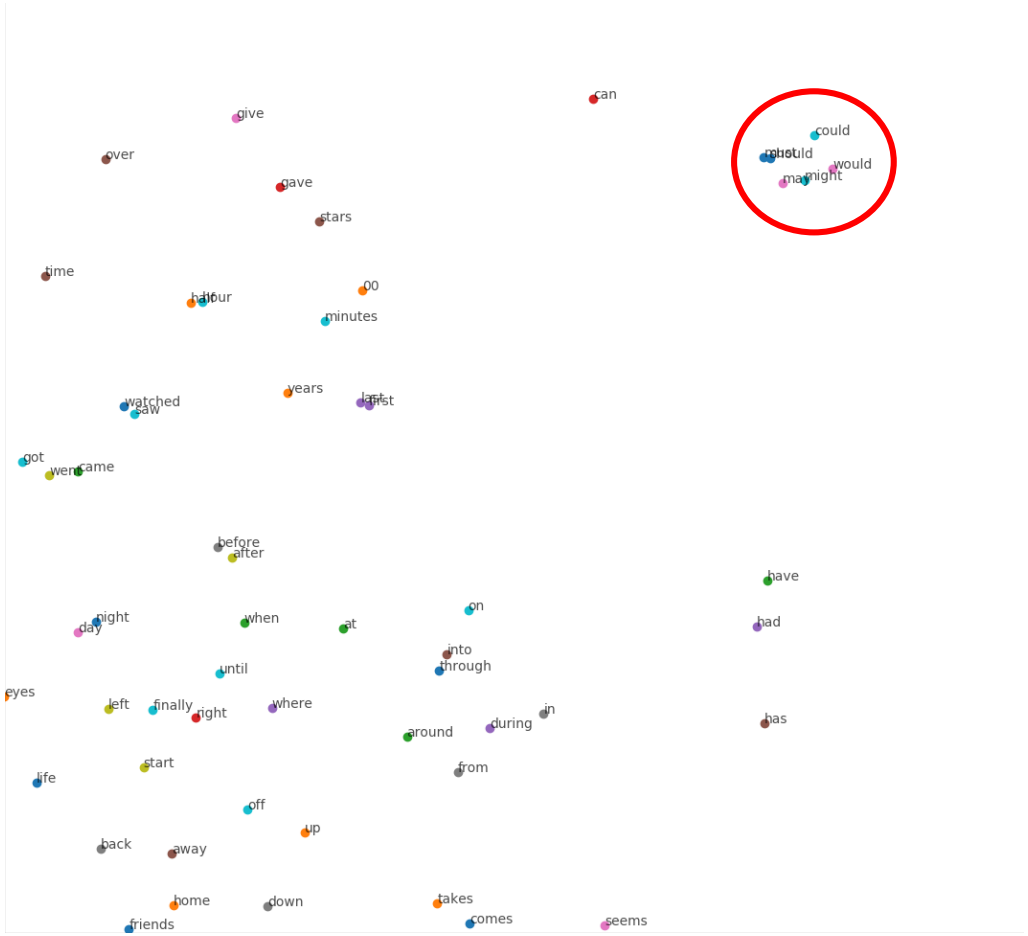
BASE Word2Vec



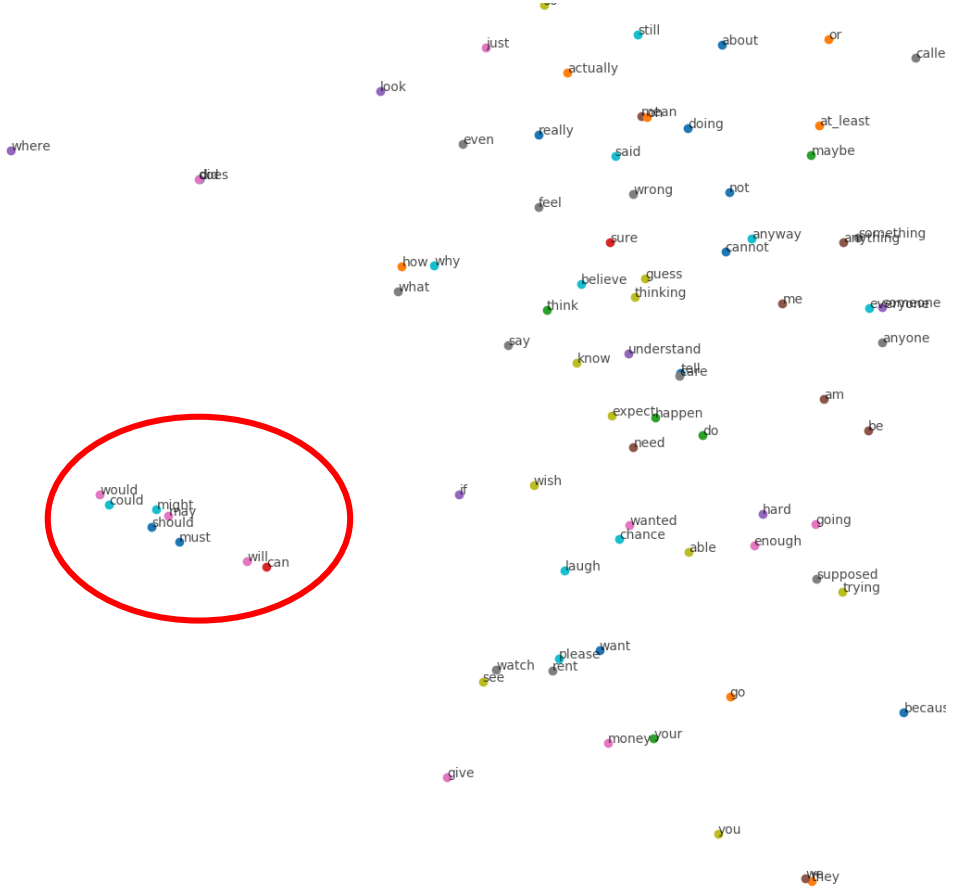
AWAR

AWAR 방법론 적용 결과 비교-Visualization

BASE Word2Vec



AWAR



AWAR 방법론 적용 결과 비교-Visualization

BASE Word2Vec

AWAR

AWAR 적용 및 결과 –Word similarity

Query Word = "child"

AWAR	BASE Word2Vec
Loves	Mr
Dead	Kid
Kid	Taking
Baby	Younger
Lover	Hit
Younger	Girlfriend
Married	Died
Parents	Himself
Girlfriend	Parents
Himself	Soon

AWAR 적용 및 결과 –Word similarity

Query Word = “nice”

AWAR	BASE Word2Vec
Fairly	Strange
Strange	Strong
Strong	Fairly
Cool	Perfectly
Solid	Hilarious
Hilarious	Moving
Cute	Cool
Great	Cute
Mostly	Talented
Extremely	Sweet

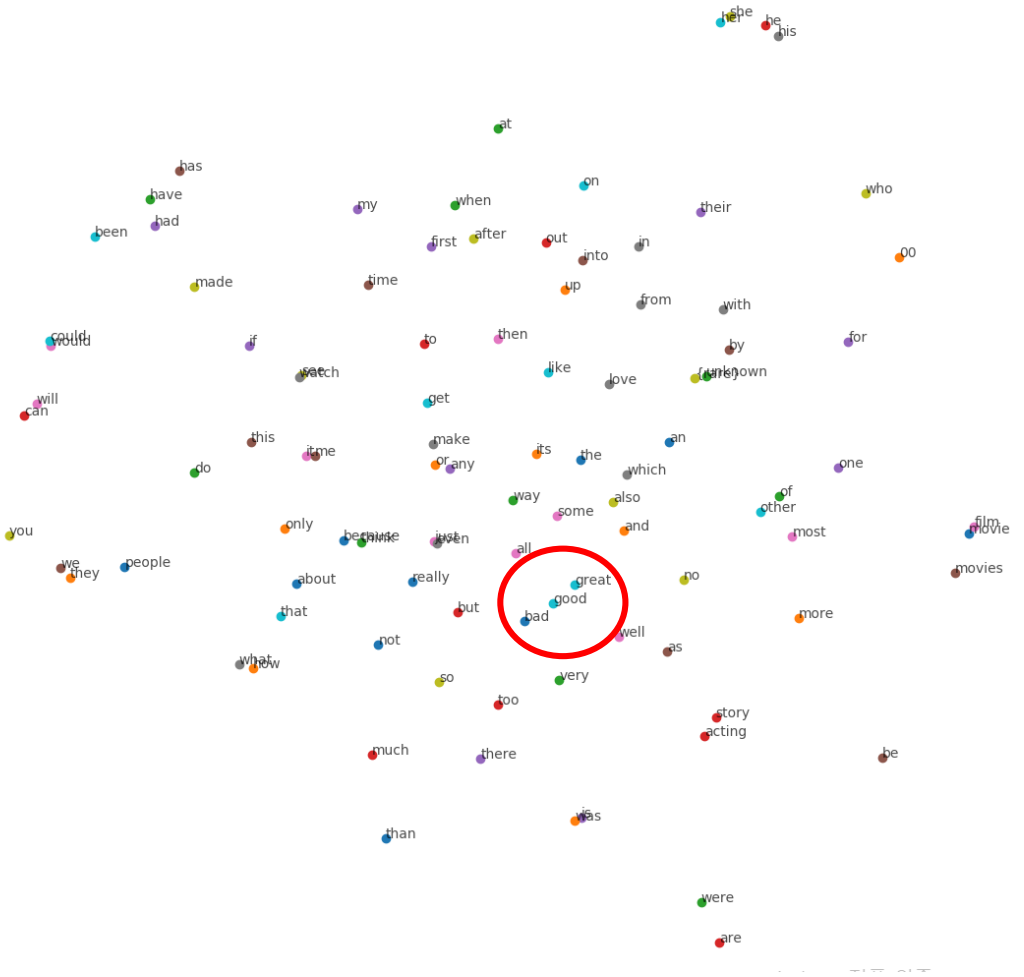
AWAR 적용 및 결과 –Word similarity

Query Word = “awesome”

AWAR	BASE Word2Vec
Surprise	Obvious
Odd	Surprise
Indeed	Odd
Obvious	Cute
Cute	Okay
Exciting	Exciting
Considering	Generally
Perfectly	Impressed
Fresh	Considering
Clever	Indeed

AWAR 방법론 적용 결과 비교-Visualization

BASE Word2Vec



AWAR

AWAR 적용 및 결과 –Word similarity

Query Word = "good"

AWAR	BASE Word2Vec
Great	Cute
Nice	Nice
Cool	Serious
Serious	Ok
Cute	Hilarious
Decent	Dumb
Scary	Bad
Hilarious	Okay
Fairly	Great
Indeed	Indeed

AWAR 적용 및 결과-Sentiment Analysis

실험개요

- AWAR 방법론의 실효성을 검증하기 위해 4가지의 실험진행.

1-1. AWAR를 통한 embedding 후, 해당 값을 update하며 Classifier 학습(end-to-end).

1-2. 기본적인 word2vec embedding 후, 해당 값을 update하며 Classifier 학습(end-to-end).

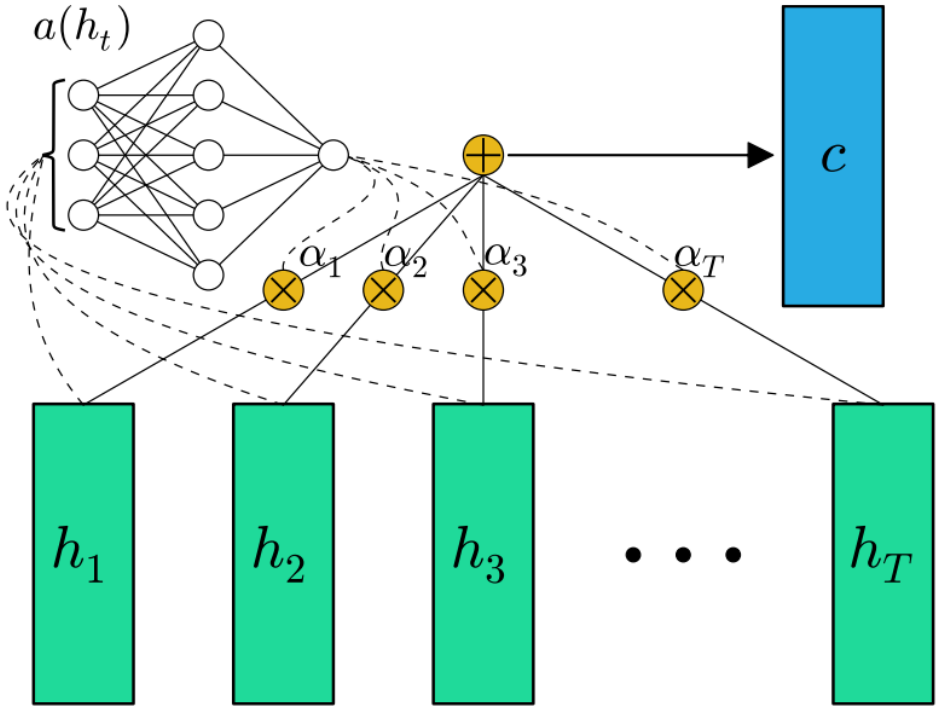
2-1. AWAR를 통한 embedding 후, 해당 값을 update하지 않고 Classifier 학습.

2-2. 기본 word2vec embedding 후, 해당 값을 update하지 않고 Classifier 학습.

AWAR 적용 및 결과 – 통제변수

RNN with Attention model

- Bi-directional RNN 구조에 GRU cell을 사용.
- Hidden dimension = 256 으로 구성
- Epoch = 60회, learning rate = 0.00001
  (overfitting 막기 위해 반복관찰 후 설정)
- Batch size = 256, Adam Optimizer 사용
- 전체 data set 중 80% 는 train
  10% 는 Validation
  10% 는 test 위해 사용함.



AWAR 적용 및 결과

1. End-to-End 상황에서 : AWAR. VS Skip-gram.

1. Accuracy

	AWAR	Skip-gram
Acc	83.30	84.1

2. Confusion Matrix

<AWAR>

		Predict	
		긍정	부정
Actual	긍정	455	45
	부정	122	378

<Skip-gram>

		Predict	
		긍정	부정
Actual	긍정	422	78
	부정	81	419

AWAR 적용 및 결과

2. Embedding update 하지 않는 상황에서 : AWAR. VS Skip-gram.

1. Accuracy

	AWAR	Skip-gram
Acc	62.20	68.00

2. Confusion Matrix

<AWAR>

		Predict	
		긍정	부정
Actual	긍정	374	126
	부정	182	318

<Skip-gram>

		Predict	
		긍정	부정
Actual	긍정	329	171
	부정	149	351

AWAR 의의 및 보완점

의의

- 단어간의 연관성을 가중치로 주면서 Word2Vec Embedding보다 더 세심한 접근이 가능
- 특정한 Specific Task에서 더 나은 성능을 가져올 가능성 존재

차후 방향

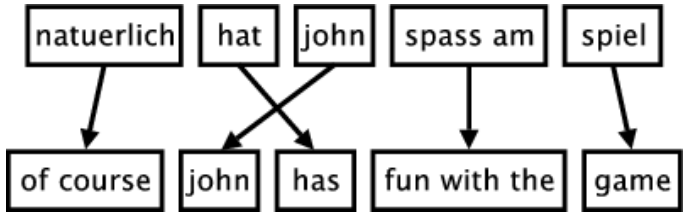
- 연관성 정보가 잘 활용될 Domain이 존재 가능성.
  -> 제안된 방법론에 적합한 Domain 과 Task연구 필요
- 연관성 분석의 정보를 활용하는 만큼 lift에 대한 세심한 parameter tuning에 대해 고민 필요

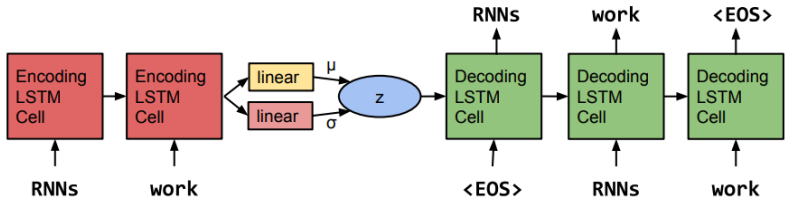
SENTENCE STYLE SELECTION: 3S

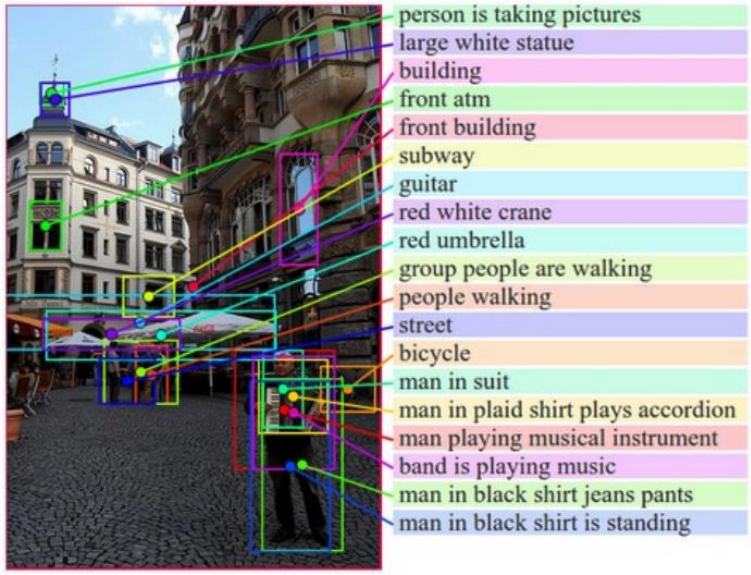
연구 배경

Generating Texts

- 기계 번역, VQA, 채팅 봇 등 다양한 영역에서 활용







- 신경망 알고리즘의 등장과 함께 활발하게 연구중인 분야 중 하나

연구 배경

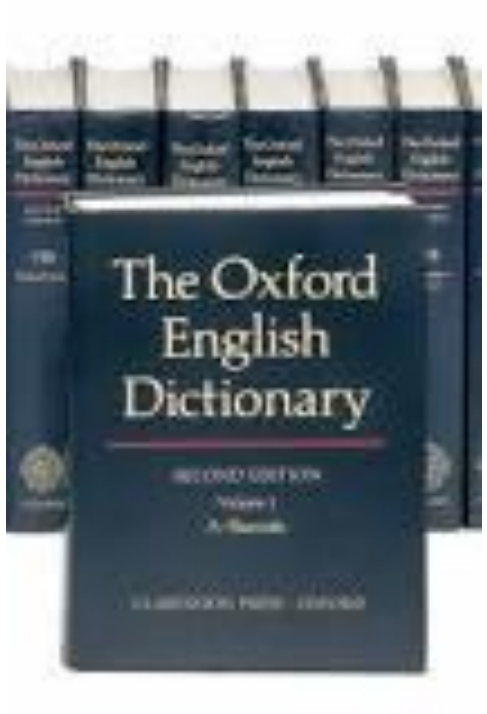
Generating Texts

- 실용적인 면에서 아직 나아갈 길이 멀다.

연구 배경

Generating Texts: Why it is difficult ?

- 한 언어권에 존재하는 단어 수가 너무 많아 계산 비용이 너무 높다.



20-volume Oxford English Dictionary

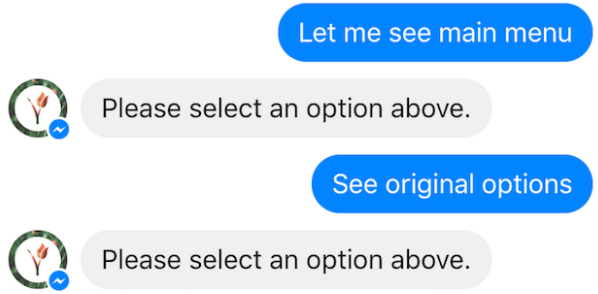
171,476 words in current use
+ (Slang) + (Derivative) + (emoticon) + …

Computational resource (memory, time) for
[*batch_size* * *sentence_length* * *vocab_size*] * epoch_size * layer_n * cell_n …

연구 배경

Generating Texts: Why it is difficult ?

- 언어라는 것이 이미지에 비해 모호함과 함축성이 크다.
- 잠재변수로 Mapping하기 이전부터 충분히 압축된 정보일 수 있다.
- 반어법, 은유/비유, 동음이의어 등...



연구 배경

Project Purpose

- 모호함 속에서 원하는 변수를 정확히 찾아내자.
- 그 변수를 통해 문장 스타일을 바꿔보자. *(Text Style Selection)*
- {긍정/부정 문장}을 {부정/긍정 문장}으로 바꿔보자.

CMD. “make negative opinion”
OUTPUT. **“The film is strictly routine !”**

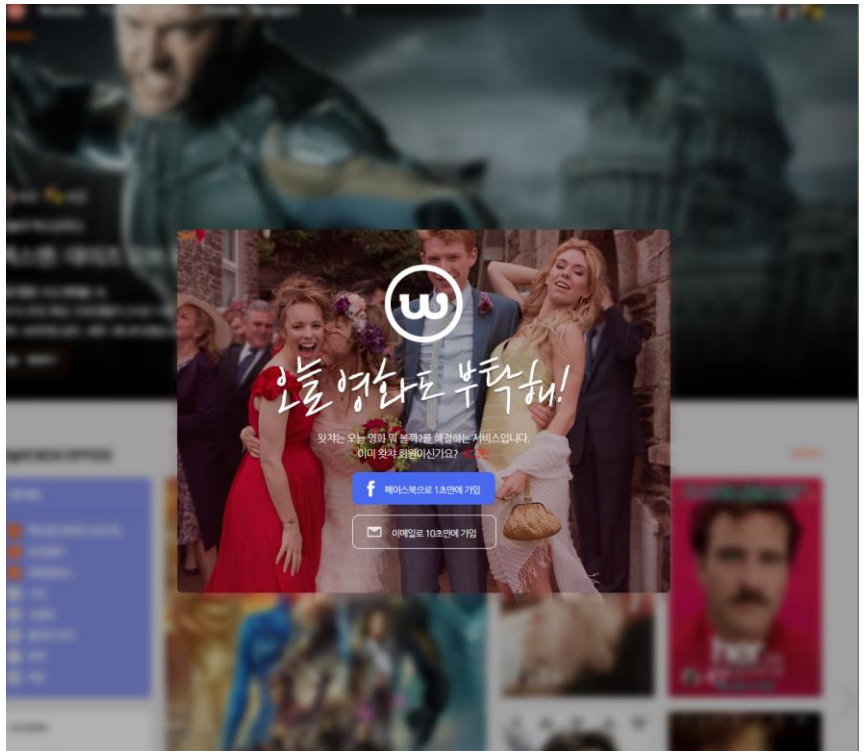
CMD. “make positive opinion”
OUTPUT. **“The film is full of imagination .”**

데이터 설명 및 전처리

WHATCHA 영화 리뷰 데이터

- 7년차 국내 기업 *"프로그램스"* 산하 영화 리뷰 서비스
- 과거에 *비영리 연구 목적으로* Web Crawling 수행
- 100% 한국 언어권 데이터
- 수집해 주신 분

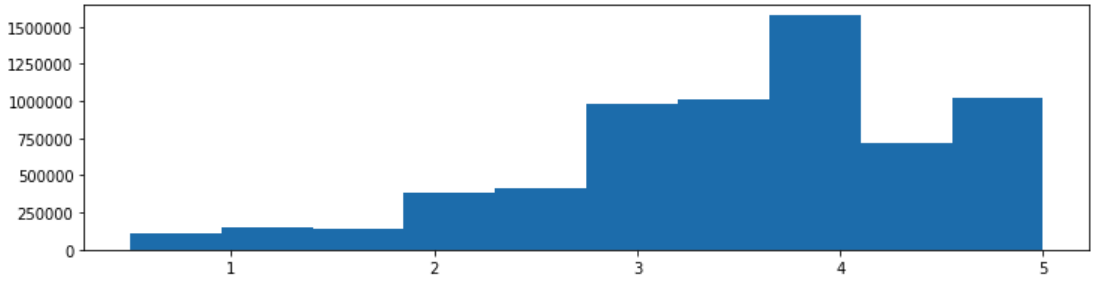
  - DSBA Lab 서덕성 (2017 졸업, SK)



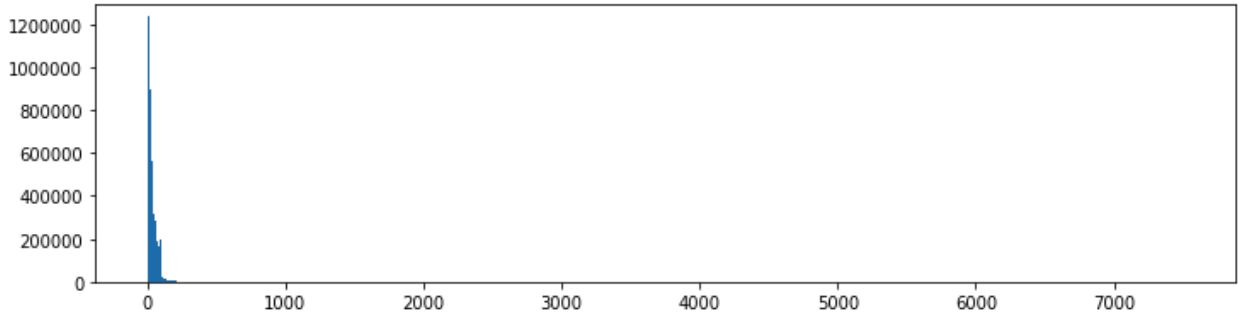
데이터 설명 및 전처리

WHATCHA 영화 리뷰 데이터

- 총 6,569,690 개의 영화 리뷰 데이터
- 1개의 데이터는 다음 요소로 이루어짐
  - **리뷰 문장**
  - **0.5 ~ 5점 10개 척도로 구분된 평점**



<평점 - 리뷰 수 히스토그램>

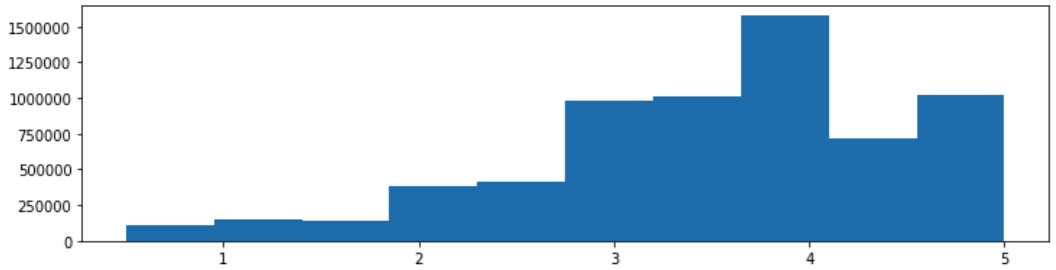


<리뷰 길이 히스토그램>

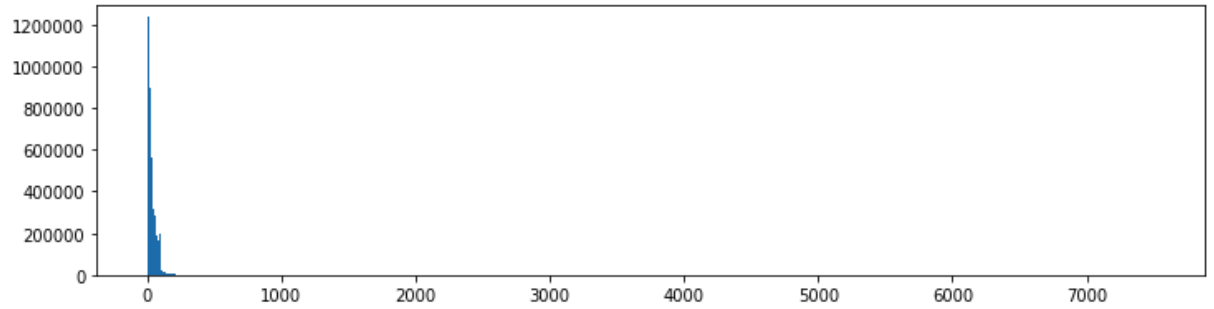
데이터 설명 및 전처리

WHATCHA 영화 리뷰 데이터

- 평점 - 리뷰 수 히스토그램
  - 긍정/부정을 결정할 평점의 **임계 값**을 찾기 위해 사용



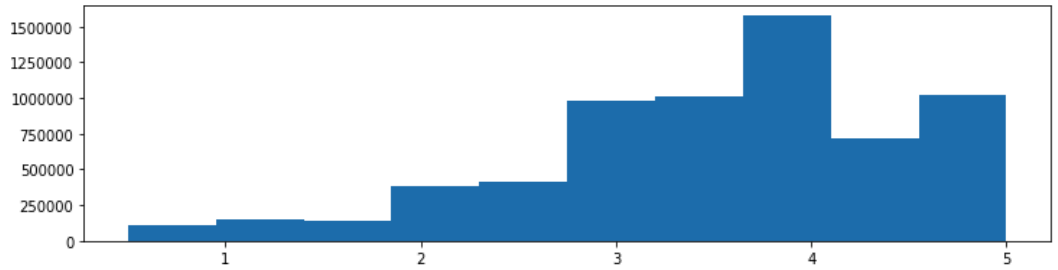
- 리뷰 길이 - 리뷰 수 히스토그램
  - 효율적인 메모리 관리를 위해 최적의 **길이 제한**을 위해 사용



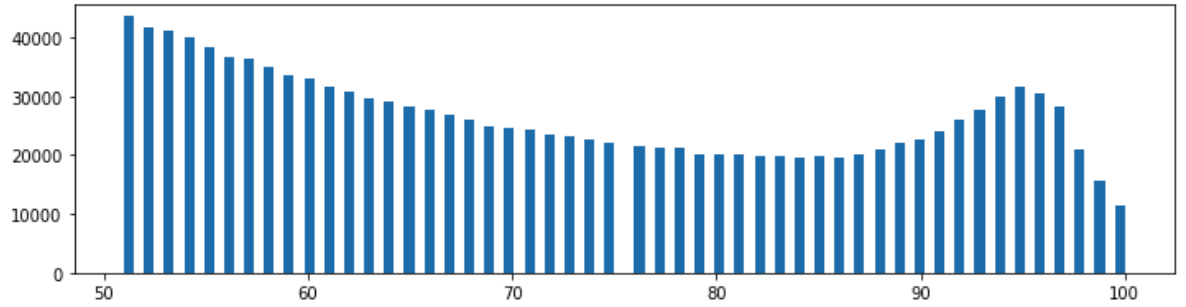
데이터 설명 및 전처리

WHATCHA 영화 리뷰 데이터

- 평점 - 리뷰 수 히스토그램
  - 긍정/부정을 결정할 평점의 임계 값을 찾기 위해 사용



- 리뷰 길이 - 리뷰 수 히스토그램
  - 효율적인 메모리 관리를 위해 최적의 길이 제한을 위해 사용
  - **글자 길이 51~100 리뷰만 이용 (1,328,875개의 문장 사용)**



데이터 설명 및 전처리

Regex

- 비정상 데이터 삭제, 글자 길이 51~100 리뷰만 남김
  1,328,875개

```
if not 50 < len(text) <= 100:
    skip = True
if not skip and re.compile(r'[a-zA-Zㄱ-ㅎㅏ-ㅣ]').search(text):
    skip = True
if not skip and re.compile(r'\w{10}').search(text):
    skip = True
if not skip and re.compile(r'(.+?)\1{3}').search(text):
    skip = True
```

데이터 설명 및 전처리

Regex

- 명확한 음절이 아닌 문자를 가진 데이터 삭제
  질 좋은 학습 데이터 확보를 위해

```
if not 50 < len(text) <= 100:
    skip = True
if not skip and re.compile(r'[a-zA-Zㄱ-ㅎㅏ-ㅣ]').search(text):
    skip = True
if not skip and re.compile(r'\w{10}').search(text):
    skip = True
if not skip and re.compile(r'(.+?)\1{3}').search(text):
    skip = True
```

데이터 설명 및 전처리

Regex

- 한 단어가 10글자를 넘어가는 데이터 삭제
  띄어쓰기 안 한 리뷰 차단

```
if not 50 < len(text) <= 100:
    skip = True
if not skip and re.compile(r'[a-zA-Zㄱ-ㅎㅏ-ㅣ]').search(text):
    skip = True
if not skip and re.compile(r'\w{10}').search(text):
    skip = True
if not skip and re.compile(r'(.+?)\1{3}').search(text):
    skip = True
```

데이터 설명 및 전처리

Regex

- 동일한 시퀀스가 4번 이상 반복되는 데이터 삭제
  의미 없는 도배 글 차단

```
if not 50 < len(text) <= 100:
    skip = True
if not skip and re.compile(r'[a-zA-Zㄱ-ㅎㅏ-ㅣ]').search(text):
    skip = True
if not skip and re.compile(r'\w{10}').search(text):
    skip = True
if not skip and re.compile(r'(.+?)\1{3}').search(text):
    skip = True
```

데이터 설명 및 전처리

Regex

- 그 외

모든 숫자와 숫자만의 나열은 0으로 변경

특수 문자 공백으로

공백 통일

등등...

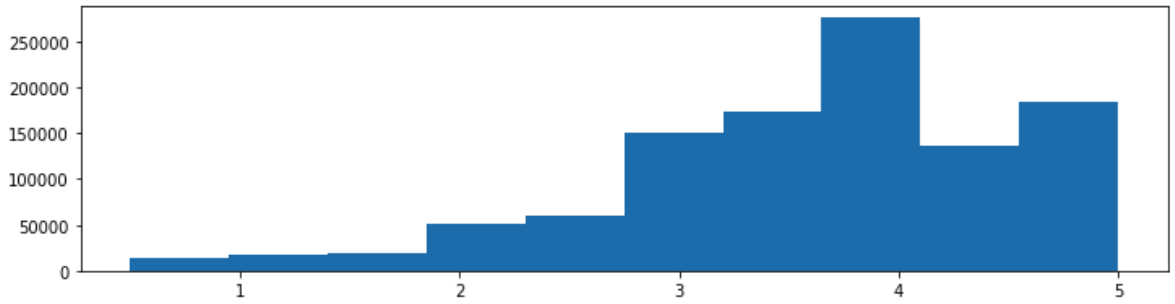
1,080,081개

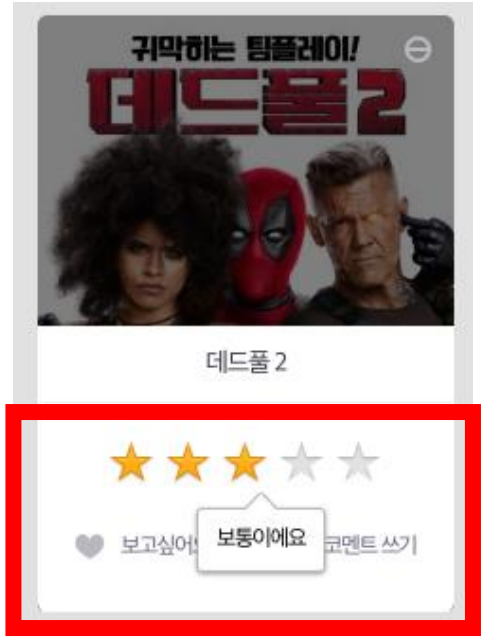
데이터 설명 및 전처리

Labeling

- 평점 3 이하 데이터 311,821개 <긍정 LABEL> [0.0, 1.0]
- 평점 3.5 ~ 4 데이터 448,757개 <중립 LABEL> [0.5, 0.5]
- 평점 4.5 이상 데이터 319,503개 <부정 LABEL> [1.0, 0.0]

개수의 표준 편차가 가장 적은 임계 값 사용 (3 분할)





데이터 설명 및 전처리

POS-Tagging

```
아버지가방에들어가신다
[('아버지', 'Noun'), ('가방', 'Noun'), ('에', 'Josa'), ('들어가신', 'Verb'), ('다', 'Eomi')]
[('아버지', 'NNG'), ('가', 'JKS'), ('방', 'NNG'), ('에', 'JKB'), ('들어가', 'VV'), ('신다', 'EP+EC')]
```

- **Mecab-KO** 사용 (실험 결과 가장 성능이 좋음)
- 동음이의어를 방지하기 위해 품사를 함께 단어에 합침.

```
아버지가방에들어가신다
아버지-NNG 가-JKS 방-NNG 에-JKB 들어가-VV 신다-EP+EC
```

데이터 설명 및 전처리

Embedding

- 동음이의어를 방지하기 위해 품사를 함께 단어에 합침.

```
아버지가방에들어가신다
아버지-NNG 가-JKS 방-NNG 에-JKB 들어가-VV 신다-EP+EC
```

- Word2Vec 알고리즘을 이용해 Embedding
  - Number of Epoch: 100
  - 제외하는 단어 없음
  - 512차원 Mapping
  - 단어가 없는 공간은 Zero Vector로 설정

```
model.wv.similar_by_word('영화-NNG')
```

```
[('작품-NNG', 0.6001042127609253),
 ('시리즈-NNG', 0.4474421441555023),
 ('스릴러-NNG', 0.441961407661438),
 ('애니메이션-NNG', 0.44161438941955566),
 ('장르-NNG', 0.41859152913093567),
 ('드라마-NNG', 0.41635411977767944),
 ('로코-NNG', 0.41328907012939453),
 ('다큐-NNG', 0.4026246666908264),
 ('애니-NNP', 0.39802104234695435),
 ('블록버스터-NNP', 0.39483362436294556)]
```

```
model.wv.similar_by_word('한국-NNP')
```

```
[('국내-NNG', 0.5803348422050476),
 ('우리나라-NNG', 0.5653473138809204),
 ('일본-NNP', 0.5039248466491699),
 ('미국-NNP', 0.46860307455062866),
 ('대한민국-NNP', 0.43743646144866943),
 ('헐리우드-NNP', 0.4243888854980469),
 ('중국-NNP', 0.42199772596359253),
 ('나라-NNG', 0.4210081696510315),
 ('국산-NNG', 0.41553574800491333),
 ('할리우드-NNP', 0.40409353375434875)]
```

데이터 설명 및 전처리

Fake Data

- 추후 모델에 사용하게 될 가짜 데이터 생성

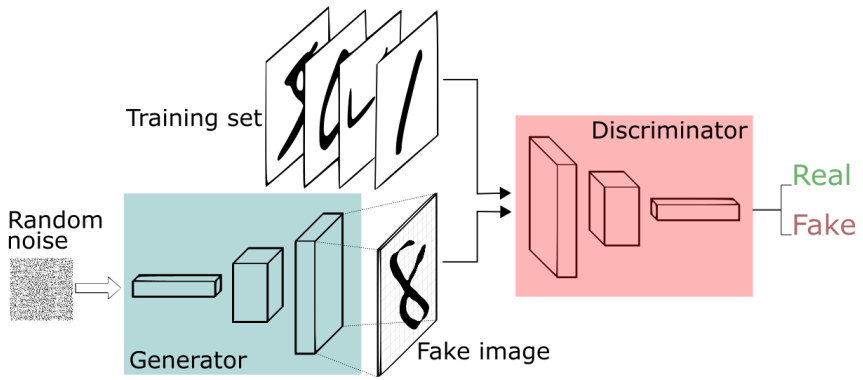
아버지-NNG 가-JKS 방-NNG 에-JKB 들어가-VV 신다-EP+EC
방-NNG 에-JKB 들어가-VV 가-JKS 아버지-NNG 신다-EP+EC

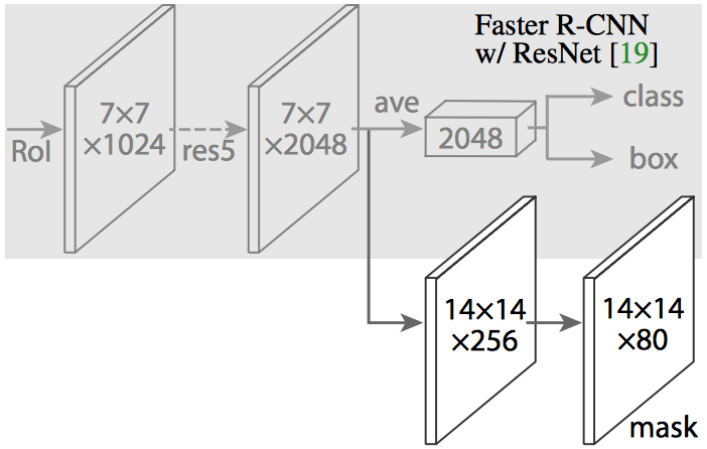
- 원 문장의 순서를 30퍼센트 이상 손상시킨 데이터 생성
  - 원본 데이터: <"REAL" LABEL>
  - 손상된 데이터: <"FAKE" LABEL>

모델 구축

Related Works

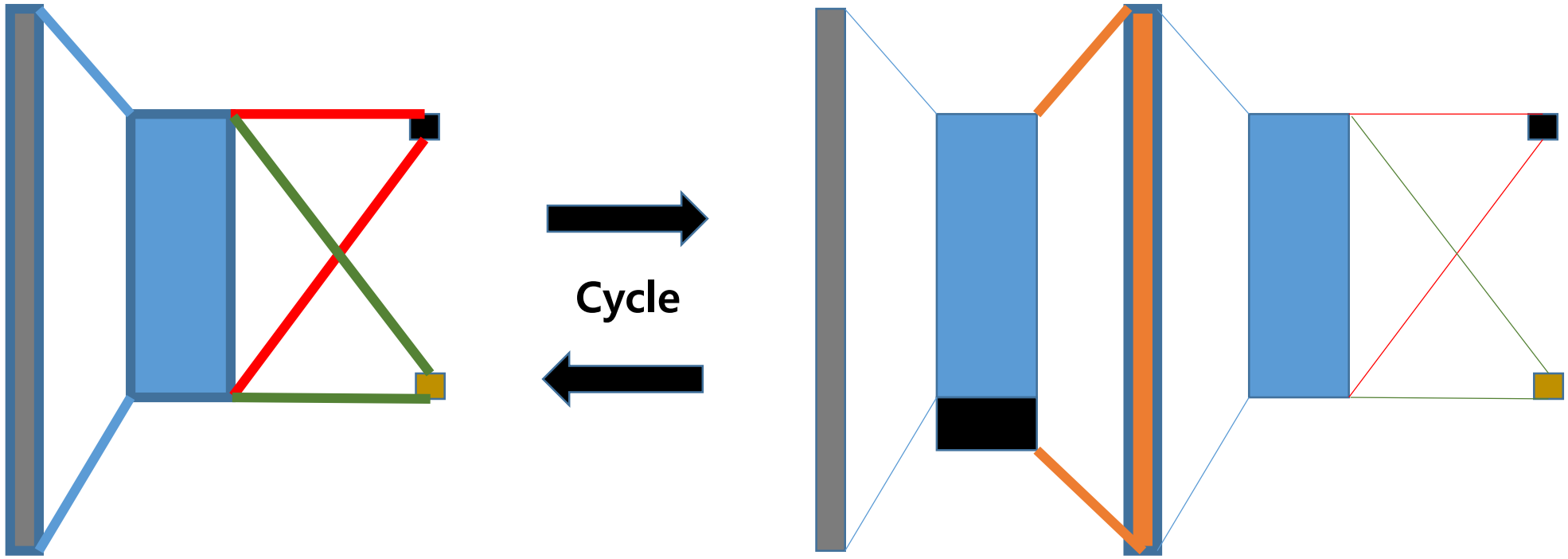
1. Long short-term memory. – Hochreiter & Schmidhuber, 1997
2. Efficient Estimation of Word Representations in Vector Space - Mikolov et al., 2013
3. Generative Adversarial Nets (GANs) - Goodfellow et al., 2014
4. Toward Controlled Generation of Text – Hu et al., 2017
5. Faster-RCNN - Girshick, 2015





모델 구축

Overview



모델 구축

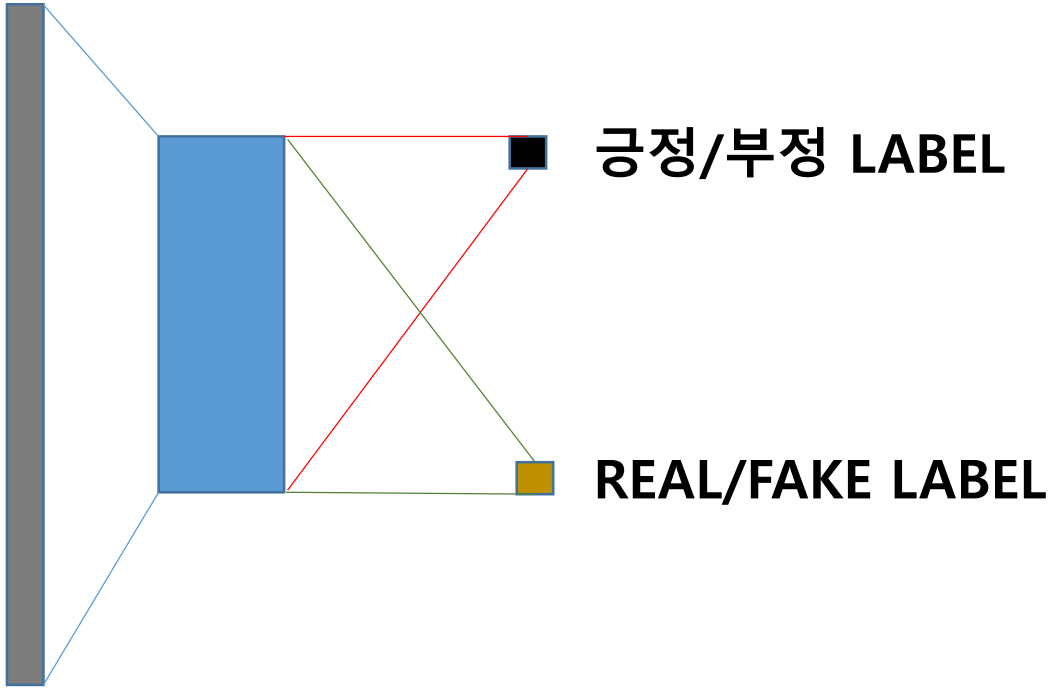
Overview: WAKE

INPUT

Batch Size 256

Real문장과 Fake문장

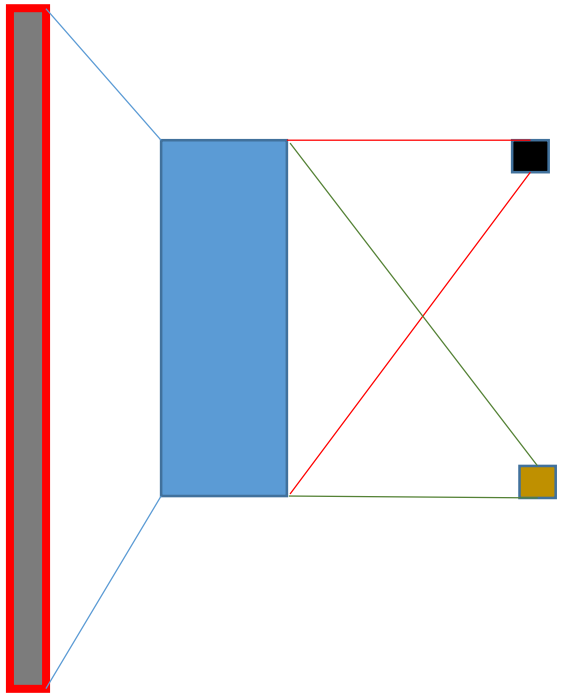
긍정/부정 정답 Label
Fake/Real 정답 Label



WAKE
(Init)

모델 구축

Overview: WAKE

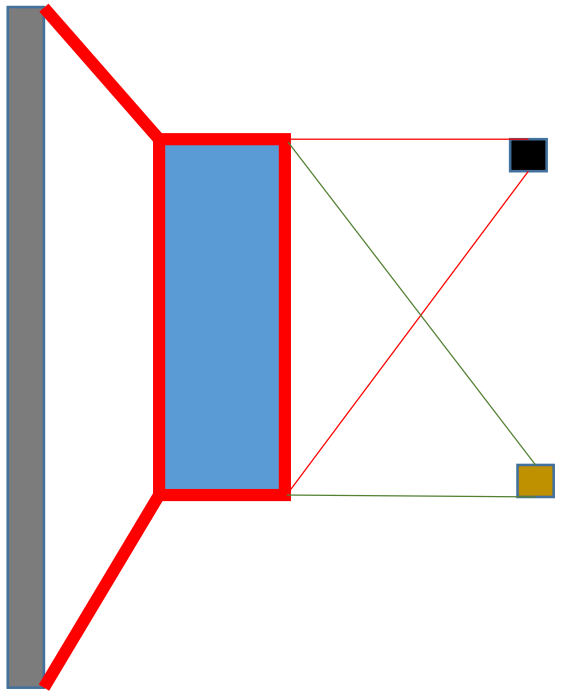


Lookup Table Variable

RETURN 64 by 512 ArrayList

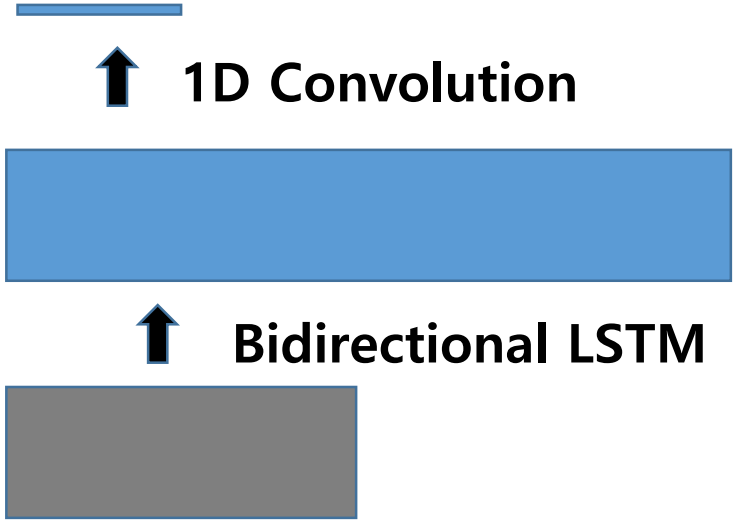
모델 구축

Overview: WAKE



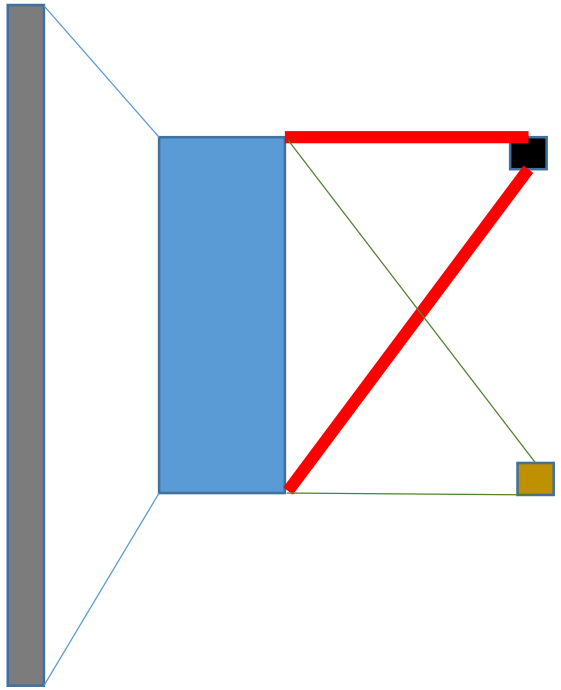
Embedding to Latent Space

RETURN 1 by 512 ArrayList



모델 구축

Overview: WAKE



Style Discriminator

RETURN 1 by 2 Softmax

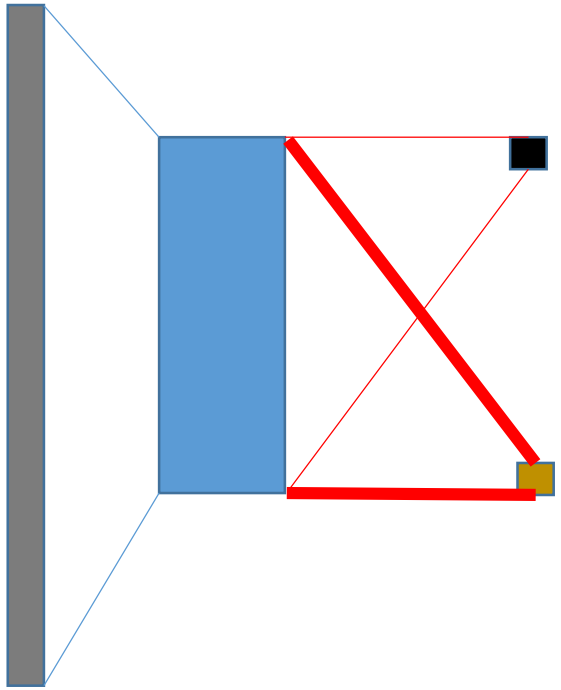
3 Fully Connected Layers

Activation Function: Sigmoid

Dropout Rate: 0.5

모델 구축

Overview: WAKE



Fake Discriminator

RETURN 1 by 2 Softmax

3 Fully Connected Layers

Activation Function: Sigmoid

Dropout Rate: 0.5

모델 구축

Overview: SLEEP

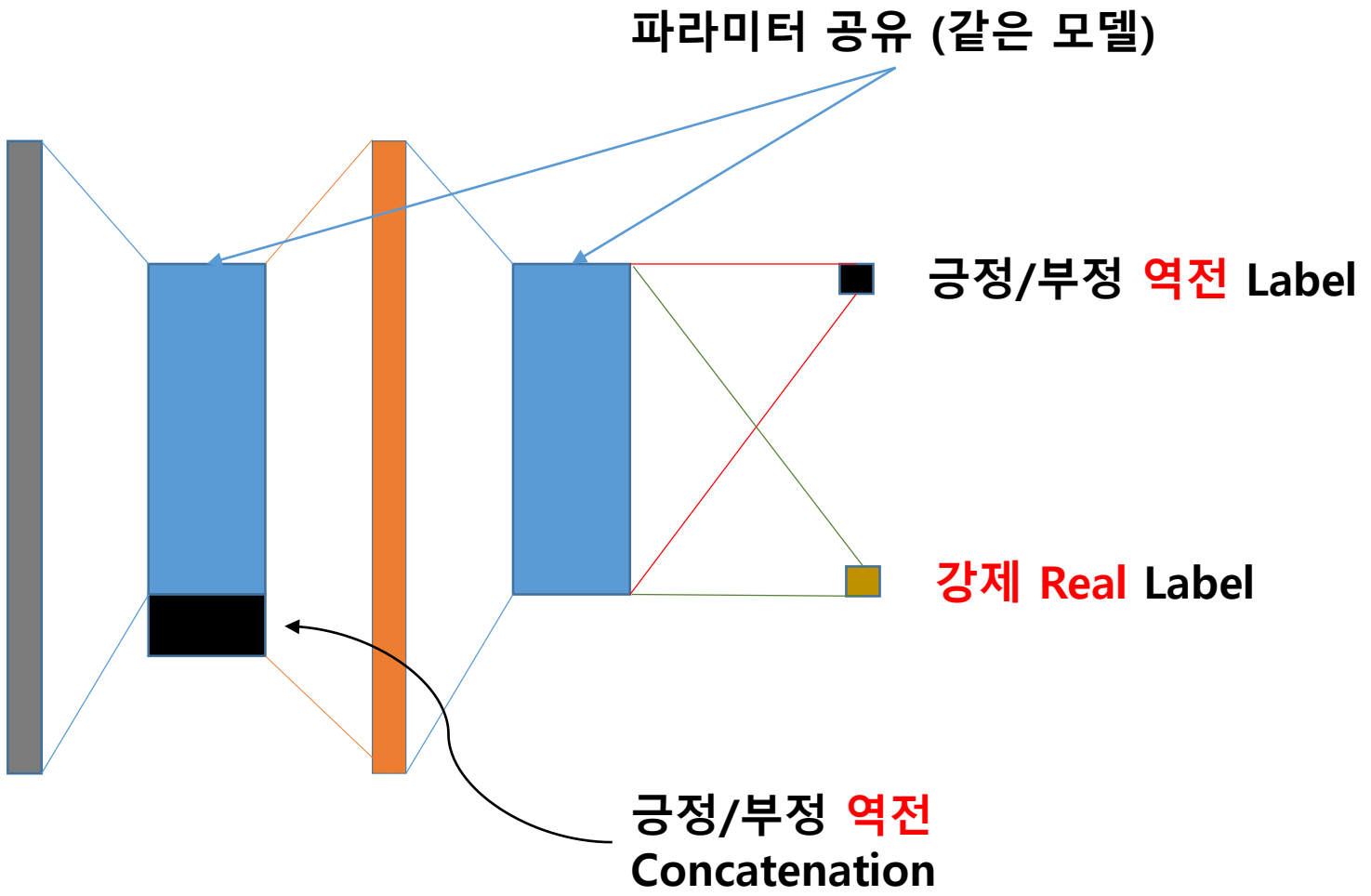
INPUT

Batch Size 256

Real문장

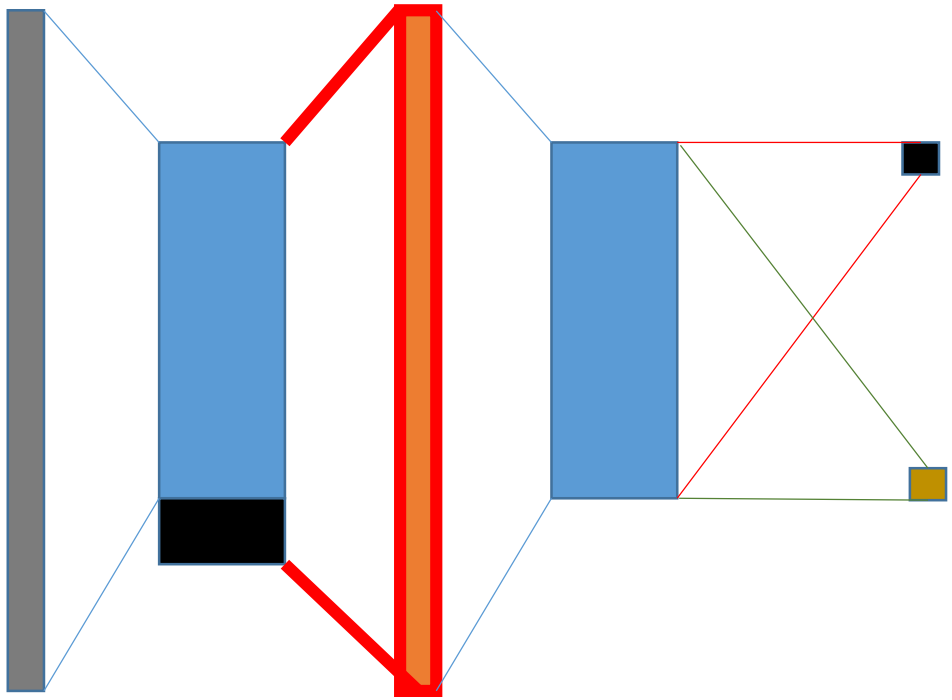
긍정/부정 역전 Label
강제 Real Label

생성모델은
차원 축소 과정이 없음
(원본 Form을 지켜야 해서)

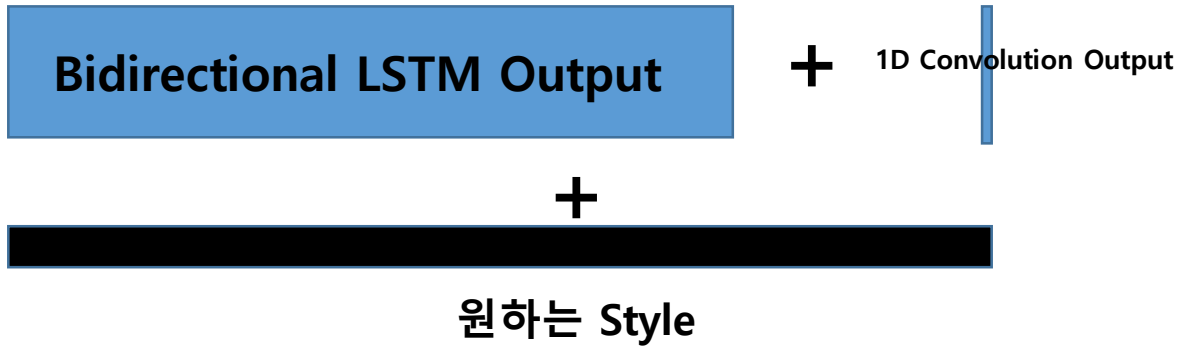


모델 구축

Overview: SLEEP



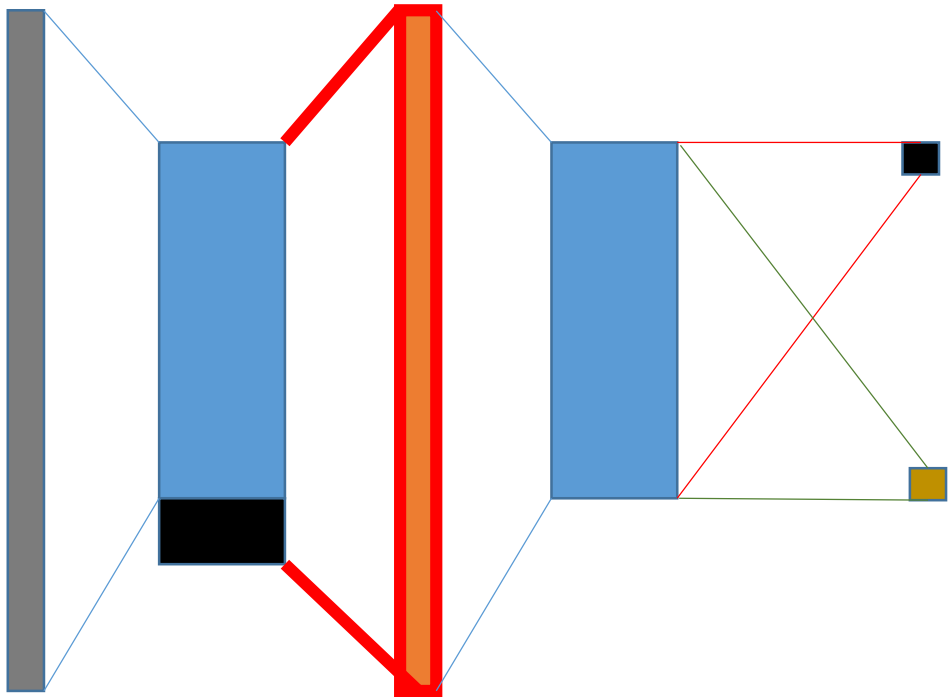
Generate Sentence



TO 128+1 by 512+2 ArrayList

모델 구축

Overview: SLEEP



Generate Sentence

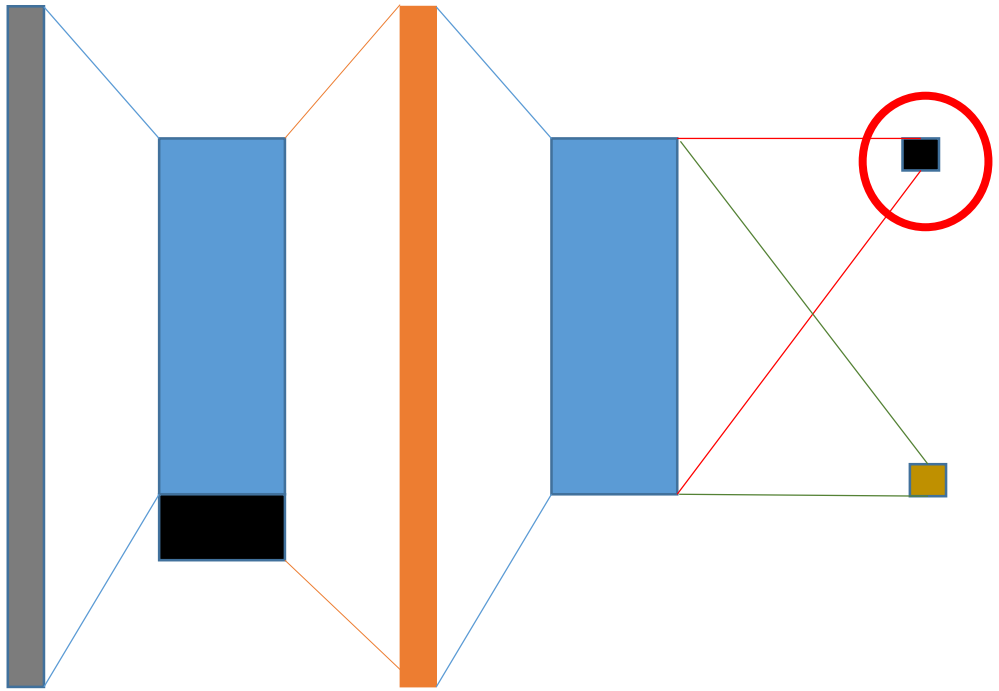
FROM 128+1 by 512+2 ArrayList

⬇ LSTM

RETURN Sentence

모델 구축

Overview: SLEEP



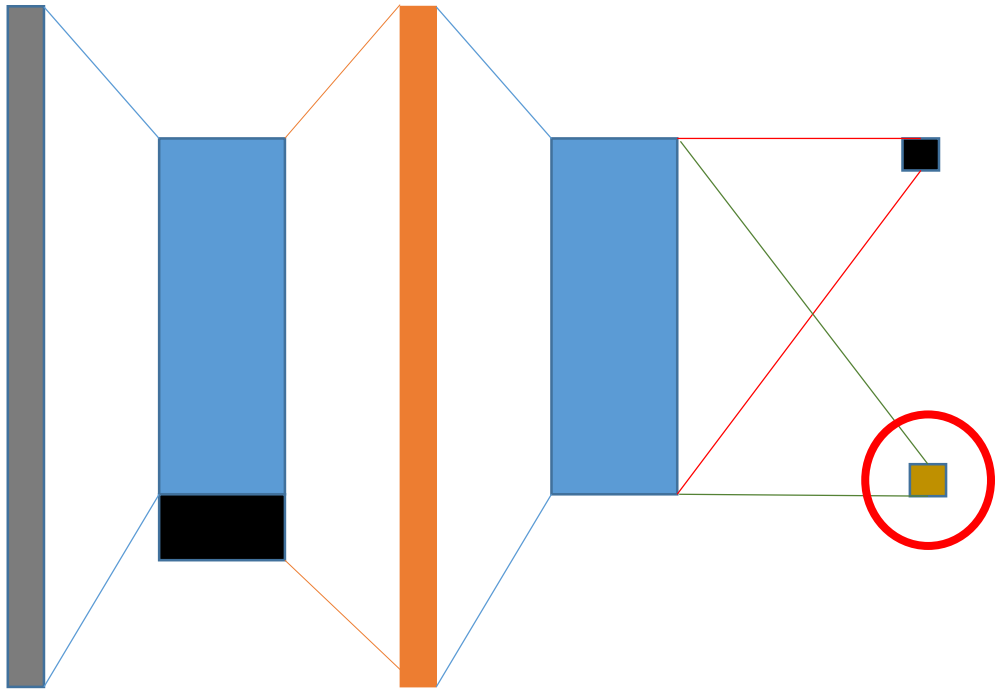
Backpropagation from Discriminator

Example

1. 긍정문을 인풋으로 넣음
2. Generator는 인풋+부정Label 정보를 이용해 문장 생성
3. Discriminator는 생성된 문장을 부정Label을 정답으로 간주하고 Loss 전파
4. 전파된 Loss를 통해 Generator학습

모델 구축

Overview: SLEEP



Backpropagation from Discriminator

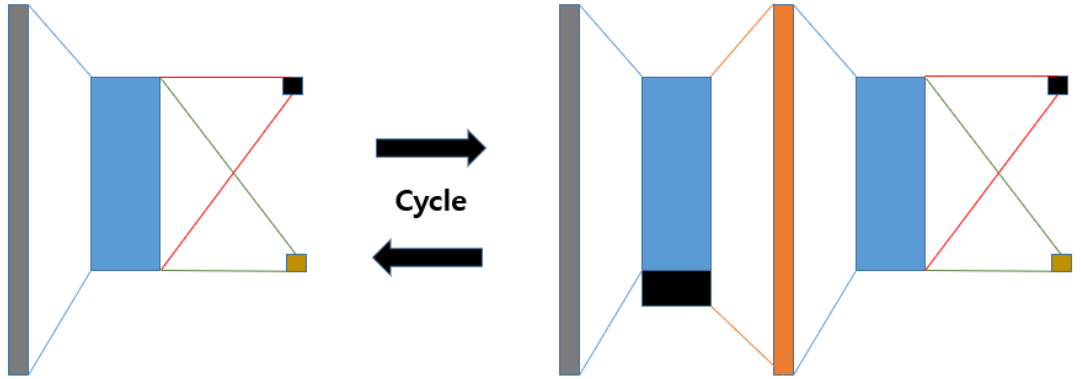
Example

1. Real 문장을 인풋으로 넣음
2. Discriminator는 생성된 문장을 Real Label을 정답으로 간주하고 Loss 전파
3. 전파된 Loss를 통해 Generator학습

모델 구축

Overview: CYCLE

1. 기존의 Fake로 Labeling된 문장을 Sleep 단계에서 만들어진 Generator가 만든 문장으로 Update
2. 1 세대 Wake 단계 수행
3. Loss와 Accuracy를 확인 한 뒤 학습이 다 안되었다면 2를 반복
4. 학습이 충분하다면 Sleep 단계 수행
5. Loss와 Accuracy를 확인 한 뒤 학습이 다 안되었다면 4를 반복
6. 개선이 없을 때까지 반복



결과 및 의의

◆ 모든 문장에서 없던 조사를 완성하는 경향

('디즈니 성품 을 알 수 있 는 영화 사실 후반부 갈등 해소 부분 에서 좋 아 버려서 아쉽 게 도 이 영화 평 을 제대로 할 수 없 게 되 었 다 나중 에 다시 찾 아 봐야 지',
 '디즈니 의 성품 을 알 수 있 는 영화 사실 후반부 의 갈등 해소 부분 에서 좋 아 버려서 아쉽 게 도 이 영화 의 평 을 제대로 할 수 없 게 되 었 다 나중 에 다시 찾 아 봐야 지 ')

어순이 바뀌는 것을 방지하기 위한 Loss + 원래의 문장을 유지하기 위한 Loss 때문

띄어쓰기를 변형 하여, 띄어쓰기 교정 모델로 확장 가능성

이 외에도 다양한 스타일 변형 작업에 응용 가능

결과 및 의의

◆ 긍정/부정이 바뀌었다고 인식하는 경우

주로 예능 과 드라마 에서 활약 하 는 연예인 들 이 영화 라는 영역 을 너무 도 장난 스럽 게 침범 못 봐 주 겠 다
그러그러 기족 과 드라마 에서 활약 하 는 뜻대 들 이 영화 라는 영역 을 너무 도 장난 스럽 게 시킬라고 못 봐 주 겠 다

원래 문장과 생성된 문장 차이의 Loss를 Mean Squared Error로 계산

특정 단어가 긍정/부정 분류에 강한 영향을 미침

문장 길이 변형이 불가능한 구조

Embedding된 Vector값에 가장 가까운 단어를 구하는 새로운 방식이 필요함

결과 및 의의

◆ 모델 고도화의 필요성

선생 님 이 어머님 의 집 에서 짐 싸 서 내려올 때 안타까움 과 눈물 이 났 다 보 는 내내 너무 고통 스러웠 지만 외면 하 고 싶 지 않 았 던 작품
선생 님 어머님 집 에서 짐 싸 서 내려올 때 안타까움 과 눈물 이 났 다 보 는 내내 너무 고통 스러웠 지만 외면 하 고 싶 지 않 았 던 작품

Generator가 본 문장과 많이 비슷한 문장을 생성할 경우 스타일 변형이 어려워 짐

남자 친구 가 이 영화 를 보 자고 했 다 난 보 면 후회 할 것 같 다고 안 보 고 싶 다고 했 다 영화 가 끝난 뒤 남자 친구 는 내게 사과 했 다
양카 미츠요 가 이 영화 를 보 고 꾸짓 다 김상봉 보 면 신신당부 할 것 같 다 쩔쩔맸 보 고 싶 다 했 다 영화 가 신파치 올랭피아 강은비 천지회 는
정찰기 끼어들기 했 다

Generator가 스타일 변형을 성공한 경우 단어 선택이 부자연스러움

적합한 내쉬 균형을 찾기 어려운 구조

GAN으로 대체할 필요성이 생김

결과 및 의의

Future Works

- 실험을 통해 맞춤법 교정의 효과가 강하게 나타남. 특히 조사를 조정해주는 모습이 두드러짐. 이를 더 발전시켜 띄어쓰기와 같은 고급 맞춤법 수정 분야까지 추구

- 본 실험에서는 이진분류를 통해 긍,부정의 의미전환을 목표로 했음. 언어의 특성 상 확실하게 구분되는 이진 분류보다 연속된 분포로 접근하는 것이 타당. 최종적으로 문장이 가지는 감정의 정확한 스코어 추구 가능

감사합니다