집현전 중급반 19조

Universal Sentence Encoder (2018)

발표일 2021년 10월 24일

발표자 우연수, 하성진, 김진환, 하상천

목차

Universal Sentence Encoder(2018)

Universal Sentence Encoder (2018)

1. Introduction

01 Introduction

- 논문 개요

논문 정보

- ✓ 제목 : **Universal Sentence Encoder**
- ✓ 저자 : Daniel Cer 외 Google Team
- ✓ 학회 : Empirical Methods in Natural Language Processing(EMNLP) 2018
- ✓ 링크 : https://arxiv.org/pdf/1803.11175.pdf

ABSTRACT

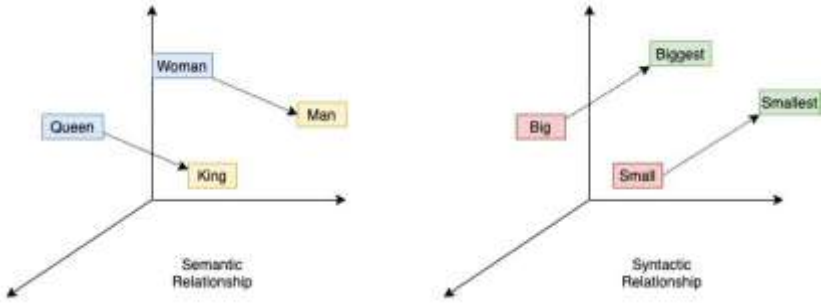
- ✓ 문장을 embedding vector로 encoding하는 모델, 특히 여러 NLP 태스크에 적용할 수 있는 전이 학습을 목표함. pre-trained word embedding을 활용한 단어 수준의 전이학습 baseline과 전이학습을 사용하지 않는 baseline를 활용해 비교 실험 실시함. **문장 embedding을 활용한 전이학습**이 단어 수준의 전이학습보다 뛰어난 결과를 보여주고, 적은 양의 지도 학습 데이터로도 좋은 성능을 나타냄.

01 Introduction

▪ 배경지식

☑ Word2vec

- ✓ Efficient Estimation of Word Representations in Vector Space(2013)
- ✓ 단어를 word embedding 으로 표현
- ✓ CBOW 모델(주변 단어로부터 목표 단어 예측)과 Skip-gram 모델(중심 단어로부터 주변 단어 예측)



☑ GloVe

- ✓ Global Vectors for Word Representation(2014)
- ✓ Word2vec : 주변 단어 중심으로 학습함으로써 말뭉치의 전체적인 통계 정보를 반영하지 못하는 문제
- ✓ 중심 단어와 주변 단어의 내적이 전체 말뭉치에서의 **동시 발생 확률**이 되도록 하는 목적 함수 설정

01 Introduction

▪ 배경지식

☑ Sentence Embedding

- ✓ 문장 수준의 embedding 기법
- ✓ **Doc2Vec(2014)**
  - ○ Word2vec 에서 확장된 개념으로, 문장·문단·문서 단위로 vector 계산
  - ○ paragraph ID를 토큰으로 사용
  - ○ PV-DM 모델(문장 다음에 오는 단어 예측)과 PV-DBOW 모델(paragraph ID만을 가지고 단어 예측)
- ✓ **InferSent(2018)**
  - ○ entailment/contradiction/neutral로 라벨링된 영어 자연어 데이터로 지도 학습된 모델
  - ○ 사전학습된 word embedding으로 GloVe 활용하는 버전1, fastText 활용하는 버전2
- ✓ **Sentence-BERT(2019)**
  - ○ Siamese Network를 사용하여 BERT를 변형한 Bi-encoder 구조
  - ○ 10,000개의 문장을 고정된 사이즈의 벡터로 표현
  - ○ Augmented SBERT (NACCL 2021)

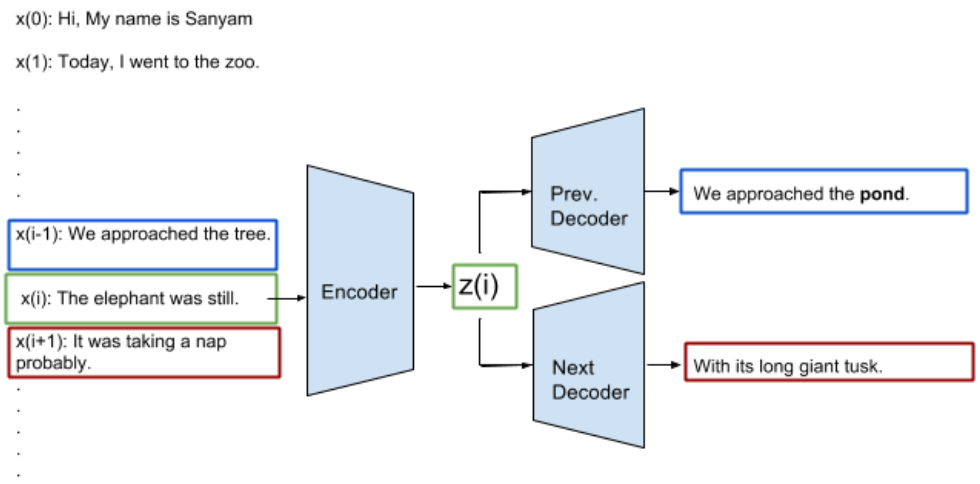
Universal Sentence Encoder(2018)

01 Introduction

▪ 배경지식

☑ Skip-Thoughts

- ✓ 고정된 길이로 문장을 표현하는 비지도 방식의 신경망 모델
- ✓ 자연어 말뭉치에서 **문장의 순서**를 정보로 사용
- ✓ 감성 분류, 유사도 계산 등 여러 downstream tasks에 사용할 수 있음
- ✓ 문장을 sequential하게 처리하는 recurrent network 사용 (GRU, LSTM)
- ✓ Encoder : i번째 문장 x(i)를 받아 고정된 길이의 표현 z(i)으로 생성
- ✓ Decoder
  - ○ Prev. : embedding z(i)를 받아 문장 x(i-1)를 생성
  - ○ Next : embedding z(i)를 받아 문장x(i+1)를 생성



01 Introduction

- Paper review

☑ 1. Introduction

○ 제한적인 NLP 학습 데이터, 비용이 많이 드는 annotation 작업
- word2vec, GloVe와 같은 pre-trained **word embeddings**를 사용해 한계를 극복하려는 시도
- 최근에는 pre-trained **sentence level embedding**을 활용한 transfer task
- 본 논문에서는 sentence embedding을 위한 두 개 모델을 제시하고, 다양한 사이즈의 transfer task 학습 데이터로 실험함으로써 **transfer task performance와 훈련셋 사이즈 간의 관계**를 파악

○ 본 논문의 sentence embedding을 사용해, 매우 적은 학습 데이터로도 좋은 성능
- 다양한 길이의 문장을 활용하여 컴퓨팅 리소스 측면에서 비교 실험

사전학습된 문장 임베딩

Universal Sentence Encoder (2018)

2. Encoders

02 Encoding

- Model Toolkit

☑ Model Design

✓ 서로 다른 설계 목표를 가지고 두가지 종류의 모델링을 진행

✓ 서로 다른 특징을 만들어낸 모델의 근본적 차이
- 임베딩을 이루는 구성 요소가 다름

✓ 임베딩 구성요소를 이루는 방식 2가지
- Unorderd
  - 단어의 순서나 문장의 구조를 고려하지 않은 채, 출현빈도(frequency)에 초점을 맞춤
  - 구성요소가 단순한만큼, 컴퓨팅 자원 소모가 적고 학습시간도 적음
  - 문장의 길이에 따라 학습시간 컴퓨팅 자원 소모 및 학습시간 선형적으로 증가
- Syntactic
  - 단어의 순서나 문장 구원을 요조 등 다양한 요소를 고려하여 임베딩을 진행
  - 그만큼, 더 긴 시간의 학습 시간 및 컴퓨팅 자구
  - 문장의 길이가 증가할경우, 급진적으로 컴퓨팅 자원 소모 및 학습 시간 증가

02 Encoding

- Model Toolkit

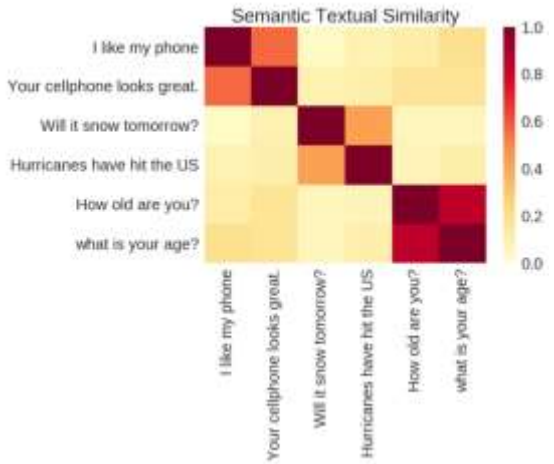
☑ Two Model

✓ 첫번째 모델: Transformer를 사용한 모델
- 높은 성능을 목표
- 높은 정확도, 높은 컴퓨팅 자원 소모

✓ 두번째 모델: DAN(Deep Average Network)를
- 효율적인 구동 및 운영을 목표
- 조금 낮은 정확도, 낮은 컴퓨팅 자원 소모

☑ Sentence Similarity Task

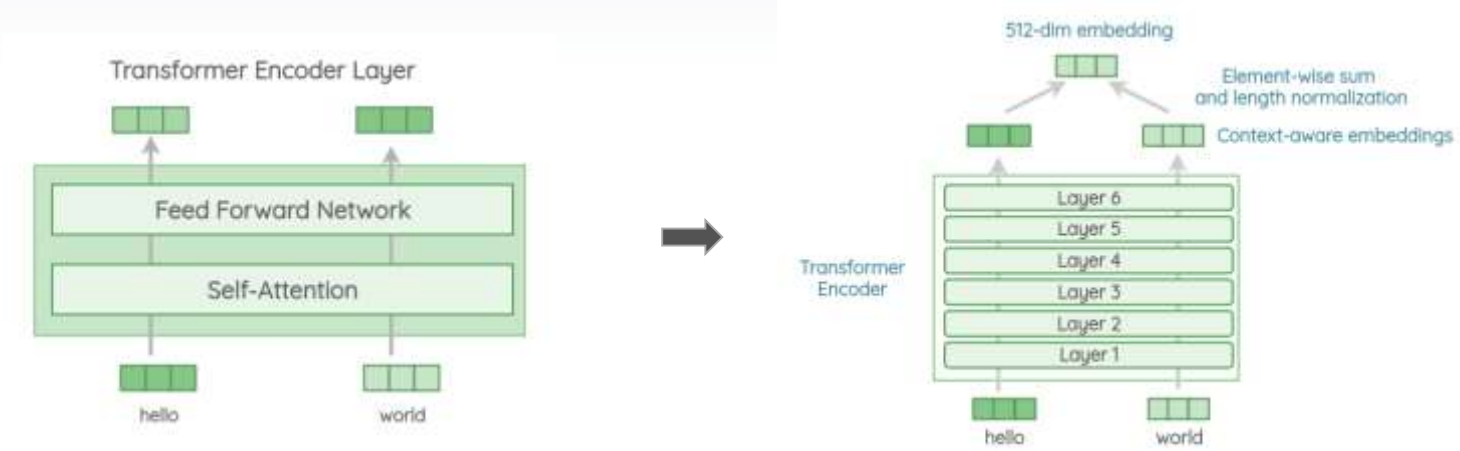
✓ 문장 유사도 테스트에서도 사용 가능
✓ semantic textual similarity(STS) 벤치마크에서 매우 훌륭한 성능을 보였음
✓ Gradient based update를 통해 특정 태스크에 최적화될 수 있음



02 Encoding

▪ Transformer

☑ [Synthetic] Transformer based sentence encoder



✓ 모델 특징
- 트랜스포머 아키텍처 기반 서브 그래프 문장 임베딩을 진행
- 문장에서의 순서 및 단어의 특징(identity)을 함께 고려
- 어텐션 방식을 이용해 context aware representation을 생성

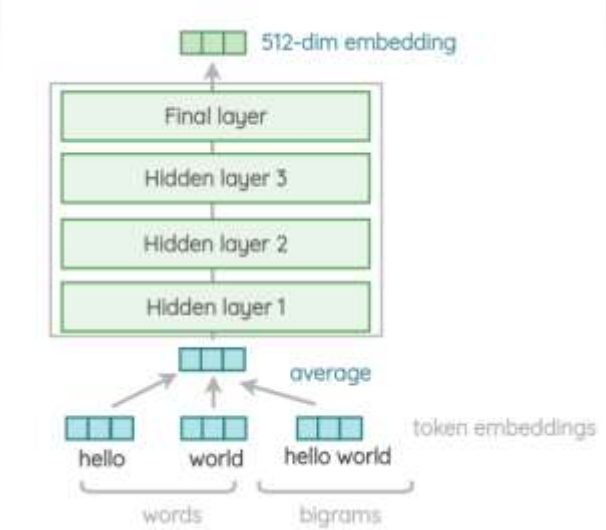
✓ 공통적 사항
- 문맥 벡터들은 고정된 길이의 문장 인코딩 벡터로 변환됨
- 인풋은 PTB 토크나이저로 토큰화된 String이며, 아웃풋은 512차원 문장 임베딩된 벡터

*PTB: Penn Treebank

02 Encoding

▪ Deep Averaging Network(DAN)

☑ [Unorderd] DAN based sentence encoder



✓ 모델 특징
- 단어와 바이그램의 임베딩을 평균내어 인풋 임베딩을 생성
- 최종적으로, 인풋을 DNN 모델에 통과시켜 문장 임베딩을 생성

✓ 공통적 사항
- 문맥 벡터들은 고정된 길이의 문장 인코딩 벡터로 변환됨
- 인풋은 PTB 토크나이저로 토큰화된 문장이며, 아웃풋은 512차원 문장 임베딩된 벡터

02 Encoding

▪ Training

☑ Multitask Learning

- ✓ 하나의 임베딩 모델로 다양한 다운스트림 태스크를 진행할 수 있도록 설계
- ✓ 지원하는 태스크
  - ○ [비지도학습] Skip Thought와 비슷한 형태의 태스크
  - ○ [지도 학습 - 대화] input - response 형태의 대화형 태스크
  - ○ [지도 학습 - 분류] classifiaction 기반의 태스크

☑ Training Data

- ✓ 비지도 학습에 필요한 데이터 수집
- ✓ 위키피디아, 웹뉴스, QnA, 토론 포럼 등 다양한 웹소스로부터 수집하여 학습
- ✓ 비지도 학습 성능 향상을 위해 지도 학습 데이터도 학습
  - ○ SNLI(Stanford Natural Laguage Inference) corpus 사용
  - ○ 전이 학습 성능 향상에 기여할 것으로 기대

Universal Sentence Encoder (2018)

3. Experiments

03 Experiments

- Transfer tasks

1. MR : Movie review snippet **sentiment** on a five star scale

2. CR : Sentiment of **sentences** mined from customer review

3. SUBJ : Subjectivity of **sentences** from movie reviews and plot summaries

4. MPQA : Phrase level opinion polarity from news data

5. TREC : Fine grained question classification sourced from TREC

03 Experiments

- Transfer tasks

6. SST : Binary phrase level **sentiment** classification

7. STS Benchmark : **Semantic textual similarity** (STS) between sentence pairs scored by Pearson correlation with human judgments

    - Using arccos to convert the cosine similarity into an angular distance

$$\text{sim}(\mathbf{u}, \mathbf{v}) = \left(1 - \arccos\left(\frac{\mathbf{u} \cdot \mathbf{v}}{||\mathbf{u}||\ ||\mathbf{v}||}\right)/\pi\right)$$

8. WEAT : Word pairs from the psychology literature on implicit association tests (IAT) that are used to characterize model bias

03 Experiments

1) Model performance on **transfer tasks**

2) Task performance on SST for **varying amounts of training data**

3) Model **resource usage** for both USE_D and USE_T at different batch sizes and sentence lengths

4) Word Embedding Association Tests (WEAT) for GloVe and the universal encoder

03 Experiments

- Model performance on transfer tasks

Model	MR	CR	SUBJ	MPQA	TREC	SST	STS Bench (dev / test)
Sentence & Word Embedding Transfer Learning							
USE_D+DAN (w2v w.e.)	77.11	81.71	93.12	87.01	94.72	82.14	–
USE_D+CNN (w2v w.e.)	78.20	82.04	93.24	85.87	97.67	85.29	–
USE_T+DAN (w2v w.e.)	81.32	86.66	93.90	88.14	95.51	86.62	–
USE_T+CNN (w2v w.e.)	81.18	87.45	93.58	87.32	98.07	86.69	–
Sentence Embedding Transfer Learning							
USE_D	74.45	80.97	92.65	85.38	91.19	77.62	0.763 / 0.719 (r)
USE_T	81.44	87.43	93.87	86.98	92.51	85.38	0.814 / 0.782 (r)
USE_D+DAN (lrn w.e.)	77.57	81.93	92.91	85.97	95.86	83.41	–
USE_D+CNN (lrn w.e.)	78.49	81.49	92.99	85.53	97.71	85.27	–
USE_T+DAN (lrn w.e.)	81.36	86.08	93.66	87.14	96.60	86.24	–
USE_T+CNN (lrn w.e.)	81.59	86.45	93.36	86.85	97.44	87.21	–
Word Embedding Transfer Learning							
DAN (w2v w.e.)	74.75	75.24	90.80	81.25	85.69	80.24	–
CNN (w2v w.e.)	75.10	80.18	90.84	81.38	97.32	83.74	–
Baselines with No Transfer Learning							
DAN (lrn w.e.)	75.97	76.91	89.49	80.93	93.88	81.52	–
CNN (lrn w.e.)	76.39	79.39	91.18	82.20	95.82	84.90	–

Table 2: Model performance on transfer tasks. *USE_T* is the universal sentence encoder (USE) using Transformer. *USE_D* is the universal encoder DAN model. Models tagged with *w2v w.e.* make use of pre-training word2vec skip-gram embeddings for the transfer task model, while models tagged with *lrn w.e.* use randomly initialized word embeddings that are learned only on the transfer task data. Accuracy is reported for all evaluations except STS Bench where we report the Pearson correlation of the similarity scores with human judgments. Pairwise similarity scores are computed directly using the sentence embeddings from the universal sentence encoder as in Eq. (1).

03 Experiments

- Task performance on SST for varying amounts of training data

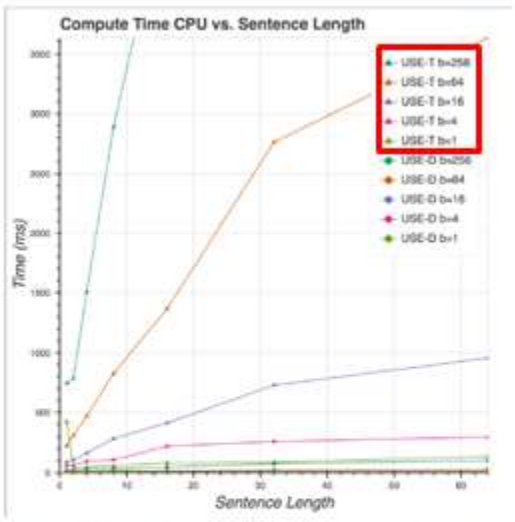
Model	SST 1k	SST 2k	SST 4k	SST 8k	SST 16k	SST 32k	SST 67.3k
Sentence & Word Embedding Transfer Learning							
USE_D+DNN (w2v w.e.)	78.65	78.68	79.07	81.69	81.14	81.47	82.14
USE_D+CNN (w2v w.e.)	77.79	79.19	79.75	82.32	82.70	83.56	85.29
USE_T+DNN (w2v w.e.)	85.24	84.75	85.05	86.48	86.44	86.38	86.62
USE_T+CNN (w2v w.e.)	84.44	84.16	84.77	85.70	85.22	86.38	86.69
Sentence Embedding Transfer Learning							
USE_D	77.47	76.38	77.39	79.02	78.38	77.79	77.62
USE_T	84.85	84.25	85.18	85.63	85.83	85.59	85.38
USE_D+DNN (lrn w.e.)	75.90	78.68	79.01	82.31	82.31	82.14	83.41
USE_D+CNN (lrn w.e.)	77.28	77.74	79.84	81.83	82.64	84.24	85.27
USE_T+DNN (lrn w.e.)	84.51	84.87	84.55	85.96	85.62	85.86	86.24
USE_T+CNN (lrn w.e.)	82.66	83.73	84.23	85.74	86.06	86.97	87.21
Word Embedding Transfer Learning							
DNN (w2v w.e.)	66.34	69.67	73.03	77.42	78.29	79.81	80.24
CNN (w2v w.e.)	68.10	71.80	74.91	78.86	80.83	81.98	83.74
Baselines with No Transfer Learning							
DNN (lrn w.e.)	66.87	71.23	73.70	77.85	78.07	80.15	81.52
CNN (lrn w.e.)	67.98	71.81	74.90	79.14	81.04	82.72	84.90

Table 3: Task performance on SST for varying amounts of training data. SST 67.3k represents the full training set. Using only 1,000 examples for training, transfer learning from USE_T is able to obtain performance that rivals many of the other models trained on the full 67.3 thousand example training set.

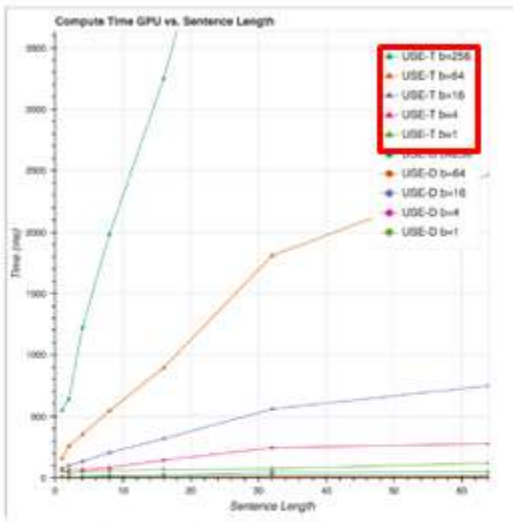
03 Experiments

- Model resource usage for both USE_D and USE_T at different batch sizes, sentence lengths

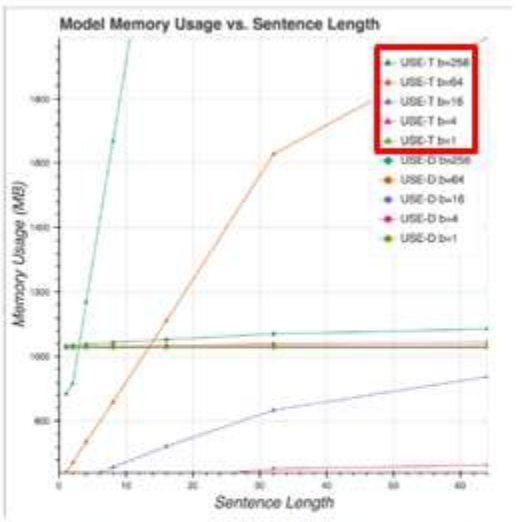
  - The transformer model space complexity also scales O(n^2)



(a) CPU Time vs. Sentence Length



(b) GPU Time vs. Sentence Length



(c) Memory vs. Sentence Length

Figure 2: Model Resource Usage for both USE_D and USE_T at different batch sizes and sentence lengths.

03 Experiments

- Word Embedding Association Tests (WEAT) for GloVe and the universal encoder

  - Weaker associations than GloVe for probes targeted at revealing at **ageism, racism and sexism**

Target words	Attrib. words	Ref	GloVe		Uni. Enc. (DAN)	
			d	p	d	p
Eur.-American vs Afr.-American names	Pleasant vs. Unpleasant 1	a	1.41	10^{-8}	0.361	0.035
Eur.-American vs. Afr.-American names	Pleasant vs. Unpleasant from (a)	b	1.50	10^{-4}	-0.372	0.87
Eur.-American vs. Afr.-American names	Pleasant vs. Unpleasant from (c)	b	1.28	10^{-3}	0.721	0.015
Male vs. female names	Career vs family	c	1.81	10^{-3}	0.0248	0.48
Math vs. arts	Male vs. female terms	c	1.06	0.018	0.588	0.12
Science vs. arts	Male vs female terms	d	1.24	10^{-2}	0.236	0.32
Mental vs. physical disease	Temporary vs permanent	e	1.38	10^{-2}	1.60	0.0027
Young vs old peoples names	Pleasant vs unpleasant	c	1.21	10^{-2}	1.01	0.022
Flowers vs. insects	Pleasant vs. Unpleasant	a	1.50	10^{-7}	1.38	10^{-7}
Instruments vs. Weapons	Pleasant vs Unpleasant	a	1.53	10^{-7}	1.44	10^{-7}

Table 4: Word Embedding Association Tests (WEAT) for GloVe and the Universal Encoder. Effect size is reported as Cohen's d over the mean cosine similarity scores across grouped attribute words. Statistical significance is reported for 1 tailed p-scores. The letters in the *Ref* column indicates the source of the IAT word lists: (a) Greenwald et al. (1998) (b) Bertrand and Mullainathan (2004) (c) Nosek et al. (2002a) (d) Nosek et al. (2002b) (e) Monteith and Pettit (2011).

03 Experiments

- Both the **transformer and DAN** based **universal encoding models** demonstrate **strong transfer performance** on a number of NLP tasks.
- The **sentence level embeddings surpass the performance** of transfer learning using **word level embeddings alone**.
- Models that make use of **sentence and word level transfer achieve the best overall performance.**
- We observe that transfer learning is **most helpful when limited training data is available** for the transfer task.
- The encoding models make different **trade-offs regarding accuracy and model complexity**

Universal Sentence Encoder (2018)

4. 실습

04 실습

1) 모델 사용 방법

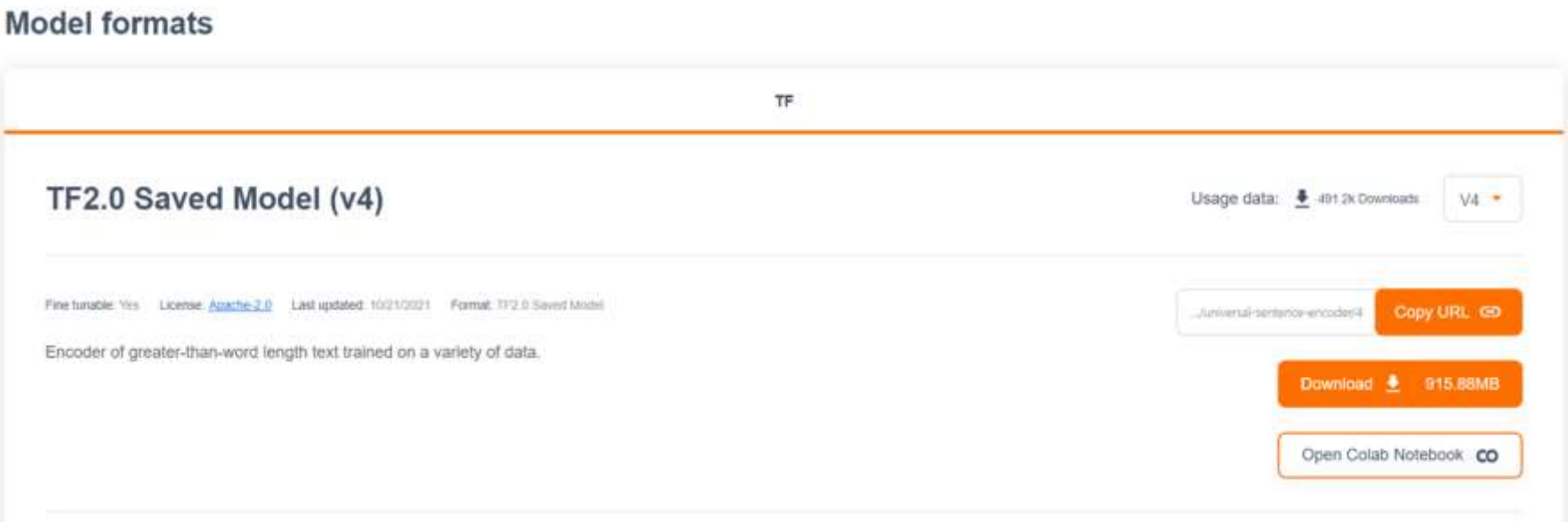
2) 문장간의 히트맵

3) Query문장과의 코사인 유사도

04 실습

- Semantic Similarity with TF-Hub Universal Sentence Encoder

  - 모델 사용 방법



https://tfhub.dev/google/universal-sentence-encoder/4

04 실습

Semantic Similarity with TF-Hub Universal Sentence Encoder

- 모델 사용 방법

```
import tensorflow_hub as hub
import tensorflow as tf

sentences = ["I ate dinner.",
             "We had a three-course meal.",
             "Brad came to dinner with us.",
             "He loves fish tacos.",
             "In the end, we all felt like we ate too much.",
             "We all agreed; it was a magnificent evening."]

model = hub.load('https://tfhub.dev/google/universal-sentence-encoder/4')
sentence_embeddings = model(sentences)

print(sentence_embeddings.shape) # (6, 512)
```

04 실습

Semantic Similarity with TF-Hub Universal Sentence Encoder

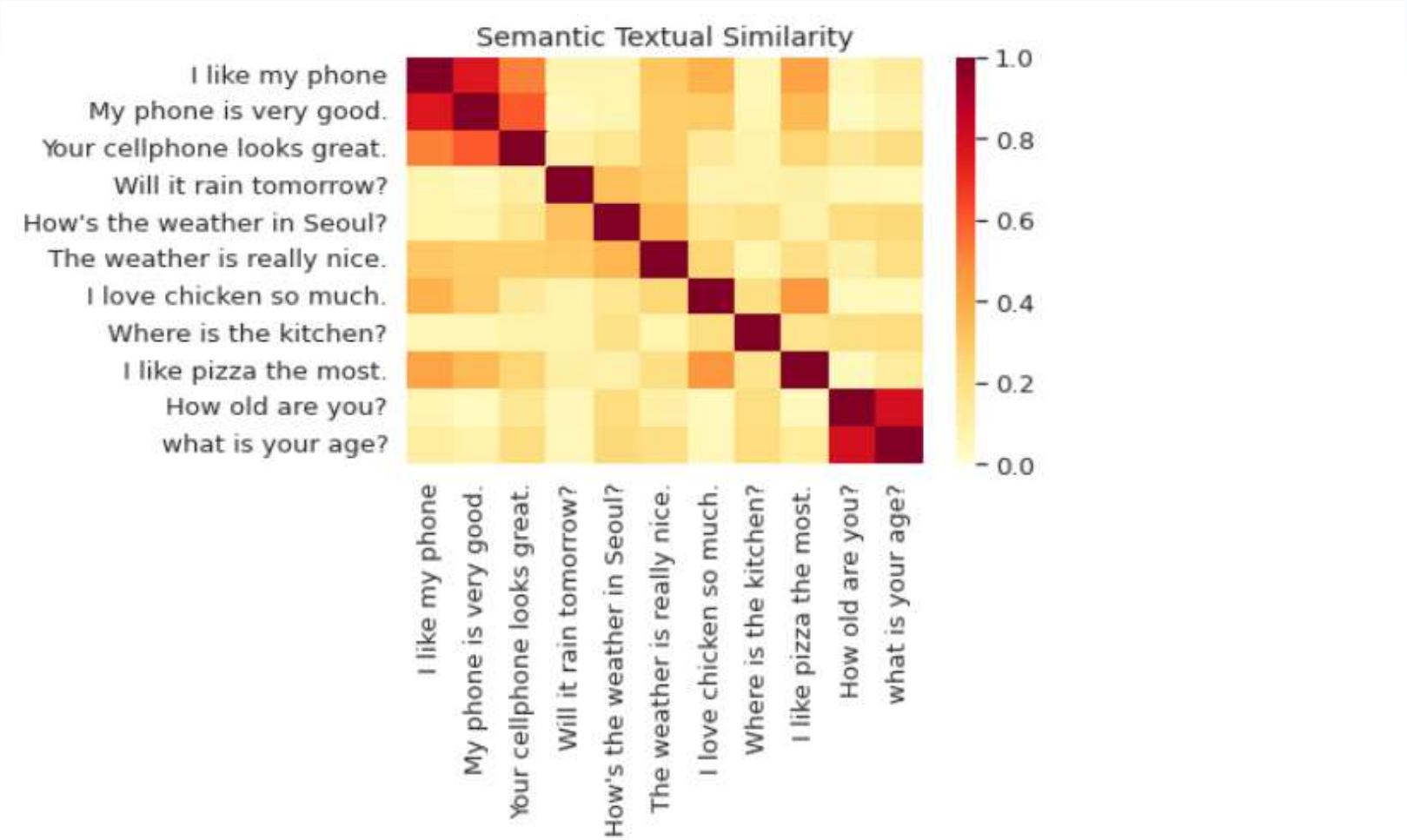
- 문장간의 히트맵

```
sentences = [
    # Smartphones
    "I like my phone",
    "My phone is very good.",
    "Your cellphone looks great.",

    # Weather
    "Will it rain tomorrow?",
    "How's the weather in Seoul?",
    "The weather is really nice.",

    # Food
    "I love chicken so much.",
    "Where is the kitchen?",
    "I like pizza the most.",

    # Asking about age
    "How old are you?",
    "what is your age?",
]

run_and_plot(sentences)
```

04 실습

- Semantic Similarity with TF-Hub Universal Sentence Encoder

  - 문장간의 히트맵



04 실습

Semantic Similarity with TF-Hub Universal Sentence Encoder

- Query문장과의 코사인 유사도

```
query = "I had pizza and pasta."

sentences = [
    "I ate dinner.",
    "We had a three-course meal.",
    "Brad came to dinner with us.",
    "He loves fish tacos.",
    "In the end, we all felt like we ate too much.",
    "We all agreed; it was a magnificent evening.",
]
```

04 실습

- Semantic Similarity with TF-Hub Universal Sentence Encoder

  - Query문장과의 코사인 유사도

```
print("Query = I had pizza and pasta.")

for sent in sentences:
    sim = cosine(query_vec, embed([sent])[0])
    print("Sentence = ", sent, "₩t similarity = ", sim)
```

```
Query = I had pizza and pasta.
Sentence =  I ate dinner.        similarity =  [0.46866423]
Sentence =  We had a three-course meal.          similarity =  [0.35643074]
Sentence =  Brad came to dinner with us.         similarity =  [0.2033895]
Sentence =  He loves fish tacos.         similarity =  [0.16515437]
Sentence =  In the end, we all felt like we ate too much.        similarity =  [0.1498742]
Sentence =  We all agreed; it was a magnificent evening.         similarity =  [0.0584359]
```


Thank you