Universal Sentence Encoder

Cer et al.

Presented by Taekyoon
July 12th, 2019

Abstract

- We present models for encoding **sentences into embedding** vectors that specifically target transfer learning to other NLP tasks.
- We find that **transfer learning** using sentence embeddings tends to outperform word level transfer.
- Our pre-trained sentence encoding models are made freely available for download and on **TF Hub**

Introduction

- Given the high cost of annotating supervised training data, **very large training sets are usually not available** for most research or industry NLP tasks **because of limited amount of training dataset**
- In this paper, we present two models for **producing sentence embeddings** that demonstrate **good transfer** to a number of other of **other NLP tasks**

Encoders

- We introduce the model architecture for our **two** encoding models in this section
- **Transformer**
  - The model is **accomplished by using multi-task learning** whereby a single encoding model is used to feed multiple downstream tasks.
  - The supported tasks include: a **SkipThought** like task (Kiros et al., 2015) for the unsupervised learning from arbitrary running text

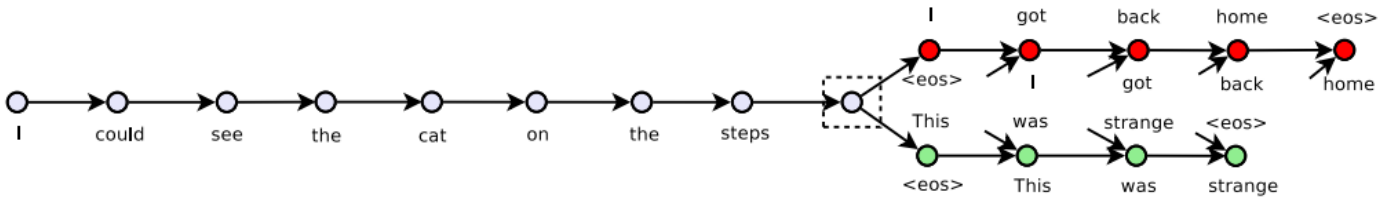
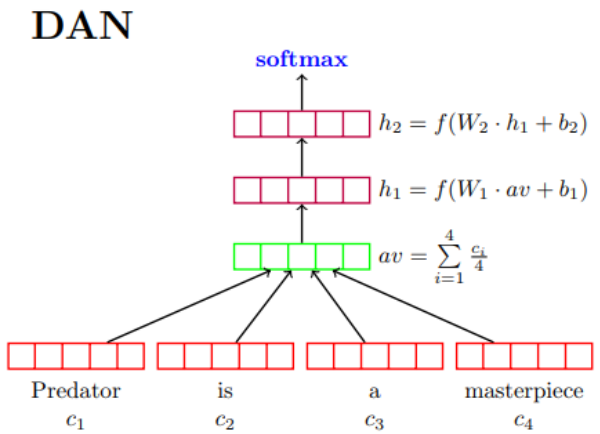


Figure 1: The skip-thoughts model. Given a tuple (s_{i-1}, s_i, s_{i+1}) of contiguous sentences, with s_i the i-th sentence of a book, the sentence s_i is encoded and tries to reconstruct the previous sentence s_{i-1} and next sentence s_{i+1}. In this example, the input is the sentence triplet *I got back home. I could see the cat on the steps. This was strange.* Unattached arrows are connected to the encoder output. Colors indicate which components share parameters. $\langle$eos$\rangle$ is the end of sentence token.

Encoders

- Deep Averaging Network (DAN)
  - The second encoding model makes use of a deep averaging network (DAN)
  - The primary advantage of the DAN encoder is that **compute time is linear in the length of the input sequence**
  - **DANs achieve strong baseline performance on text classification tasks**



Encoders

- Training Data
  - **Unsupervised training data** for the sentence encoding models are Wikipedia, web news, web question-answer pages and discussion forums.
  - We **augment unsupervised learning** with training on supervised data from the Stanford Natural Language Inference (**SNLI**) corpus

Transfer Tasks

- **Word Embedding Association Test (WEAT)**
  - **MR** : **Movie review** snippet sentiment on a five star scale
  - **CR** : Sentiment of sentences mined from **customer reviews**
  - **SUBJ** : **Subjectivity of sentences** from movie reviews and plot summaries (Pang and Lee, 2004).
  - **MPQA** : **Phrase level opinion polarity** from news data (Wiebe et al., 2005).
  - **TREC** : Fine grained **question classification** sourced from TREC (Li and Roth, 2002).
  - **SST** : Binary **phrase level sentiment classification** (Socher et al., 2013).
  - **STS** Benchmark : **Semantic textual similarity** (STS) between sentence pairs scored by Pearson correlation with human judgements
  - **WEAT** : **Word pairs** from the psychology literature on implicit association tests (IAT) that are used to characterize model bias

Transfer Learning Models

- For sentence classification transfer tasks, the output of the transformer and DAN sentence encoders are provided to a task specific DNN
- For the pairwise semantic similarity task, we directly assess the similarity of the sentence embeddings produced by our two encoders

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

Transfer Learning Models

- Baselines
  - For each transfer task, we include baselines that only make use of word level transfer and baselines that make use of no transfer learning at all
  - The pretrained word embeddings are included as input to two model types: a convolutional neural network models (CNN) (Kim, 2014); a DAN
  - Additional baseline CNN and DAN models are trained without using any pretrained word or sentence embeddings

Transfer Learning Models

- Combined Transfer Models
  - We explore combining the sentence and word level transfer models by **concatenating their representations** prior to feeding the combined representation **to the transfer task classification layers**
  - For completeness, we also explore **concatenating the representations from sentence level transfer models with the baseline models** that do not make use of word level transfer learning

Experiments

- Transfer task model hyperparamaters are tuned using a combination of Vizier (Golovin et al.) and light manual tuning. When available, model hyperparameters are tuned using task dev sets
- Otherwise, hyperparameters are tuned by cross validation on the task training data when available or the evaluation test data when neither training nor dev data are provided
- **Training repeats ten times** for each transfer task model **with different randomly initialized weights** and we report evaluation results by **averaging across runs**

Results

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).

Results

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

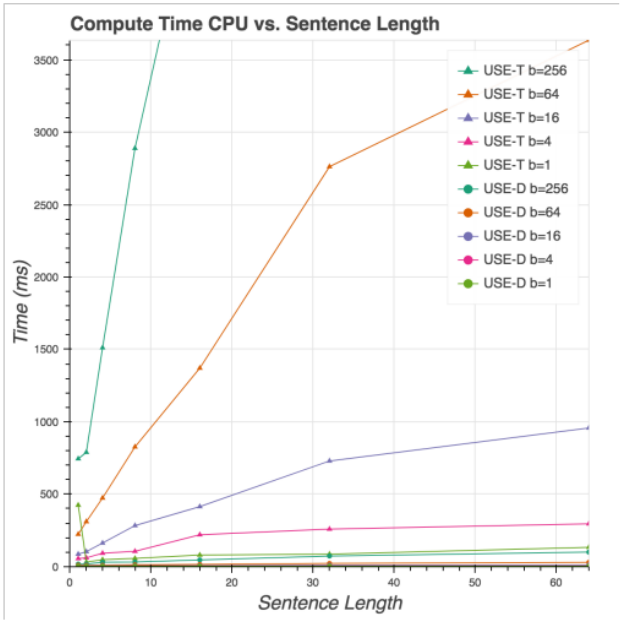
Table 3: Task performance on SST for varying amounts of training data. SST 67.3k represents the full training set. Using only 1,000 examples for training, transfer learning from USE_T is able to obtain performance that rivals many of the other models trained on the full 67.3 thousand example training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

Results

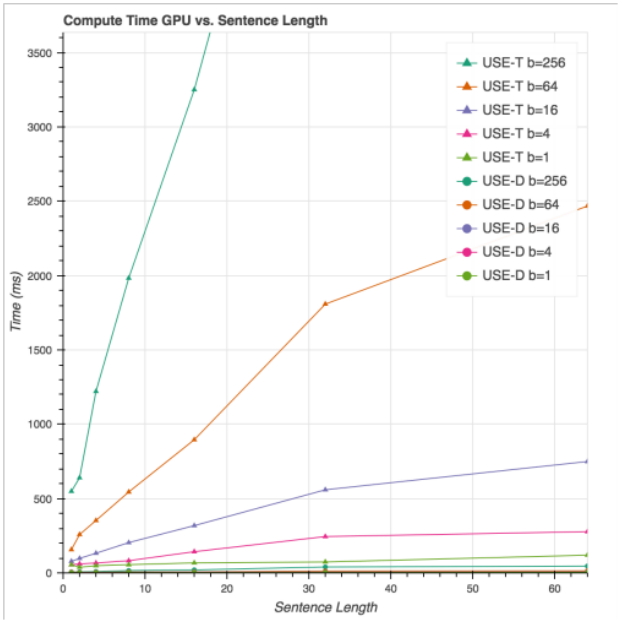
Target words	Attrib. words	Ref	GloVe		Uni. Enc. (DAN)	
			d	p	d	p
Eur.-American vs Afr.-American names	Pleasant vs. Unpleasant 1	a	1.41	10^{-8}	0.361	0.035
Eur.-American vs. Afr.-American names	Pleasant vs. Unpleasant from (a)	b	1.50	10^{-4}	-0.372	0.87
Eur.-American vs. Afr.-American names	Pleasant 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).

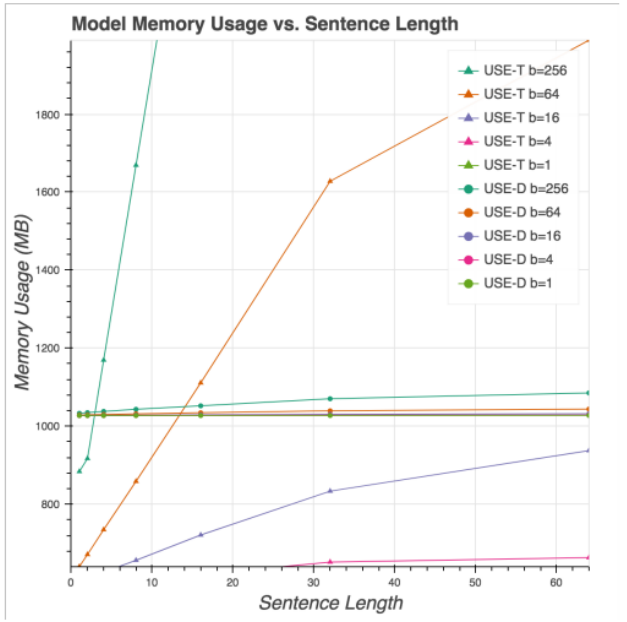
Results



(a) CPU Time vs. Sentence Length



(b) GPU Time vs. Sentence Length



(c) Memory vs. Sentence Length

Conclusion

- Models that make use of **sentence and word level transfer** achieve the **best overall performance**
- The encoding models make different **trade-offs regarding accuracy and model complexity** that should be considered when choosing the best model for a particular application