Universal Sentence Encoder for English

Daniel Cer[a†], Yinfei Yang[a†], Sheng-yi Kong[a], Nan Hua[a], Nicole Limtiaco[b], Rhomni St. John[a], Noah Constant[a], Mario Guajardo-Céspedes[a], Steve Yuan[c], Chris Tar[a], Yun-Hsuan Sung[a], Brian Strope[a], Ray Kurzweil[a]

[a]Google AI
Mountain View, CA

[b]Google AI
New York, NY

[c]Google
Cambridge, MA

Abstract

We present easy-to-use TensorFlow Hub sentence embedding models having good task transfer performance. Model variants allow for trade-offs between accuracy and compute resources. We report the relationship between model complexity, resources, and transfer performance. Comparisons are made with baselines without transfer learning and to baselines that incorporate word-level transfer. Transfer learning using sentence-level embeddings is shown to outperform models without transfer learning and often those that use only word-level transfer. We show good transfer task performance with minimal training data and obtain encouraging results on word embedding association tests (WEAT) of model bias.

1 Introduction

We present easy-to-use sentence-level embedding models with good transfer task performance even when using remarkably little training data.[1] Model engineering characteristics allow for trade-offs between accuracy versus memory and compute resource consumption.

2 Model Toolkit

Models are implemented in TensorFlow (Abadi et al., 2016) and are made publicly available on TensorFlow Hub.[2] Listing 1 provides an example

```
import tensorflow_hub as hub

embed = hub.Module("https://tfhub.dev/google/"
    "universal-sentence-encoder/2")
embedding = embed(["Hello World!"])
```

Listing 1: Python sentence embedding code.

code snippet to compute a sentence-level embedding from a raw untokenized input string.[3] The resulting embedding can be used directly or incorporated into a downstream model for a specific task.[4]

3 Encoders

Two sentence encoding models are provided: (i) transformer (Vaswani et al., 2017), which achieves high accuracy at the cost of greater resource consumption; (ii) deep averaging network (DAN) (Iyyer et al., 2015), which performs efficient inference but with reduced accuracy.

3.1 Transformer

The transformer sentence encoding model constructs sentence embeddings using the encoding sub-graph of the transformer architecture (Vaswani et al., 2017). The encoder uses attention to compute context aware representations of words in a sentence that take into account both the ordering and identity of other words. The context aware word representations are averaged together to obtain a sentence-level embedding.

We train for broad coverage using multi-task learning, with the same encoding model supporting multiple downstream tasks. The task types include: a Skip-Thought like task (Kiros et al.,

† Corresponding authors:
{cer, yinfeiy}@google.com

[1]We describe our publicly released models. See Yang et al. (2018) and Henderson et al. (2017) for additional architectural details of models similar to those presented here.

[2] https://www.tensorflow.org/hub/, Apache 2.0 license, with models available as saved TF graphs.

[3]Basic text preprocessing and white-space tokenization is performed internally. Preprocessing lowercases the text and removes punctuation. OOV items are handled using string hashing to index into 400,000 OOV embeddings.

[4]Visit https://colab.research.google.com/ to try the code snippet in Listing 1. Example code and documentation is available on the TF Hub website.

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (System Demonstrations), pages 169–174
Brussels, Belgium, October 31–November 4, 2018.

2015);[5] conversational response prediction (Henderson et al., 2017); and a select supervised classification task that improves sentence embeddings.[6] The transformer encoder achieves the best transfer performance. However, this comes at the cost of compute time and memory usage scaling dramatically with sentence length.

3.2 Deep Averaging Network (DAN)

The DAN sentence encoding model begins by averaging together word and bi-gram level embeddings. Sentence embeddings are then obtain by passing the averaged representation through a feedforward deep neural network (DNN). The DAN encoder is trained similar to the transformer encoder. Multitask learning trains a single DAN encoder to support multiple downstream tasks. An advantage of the DAN encoder is that compute time is linear in the length of the input sequence. Similar to Iyyer et al. (2015), our results demonstrate that DANs achieve strong baseline performance on text classification tasks.

3.3 Encoder Training Data

Unsupervised training data are drawn from a variety of web sources. The sources 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 (Bowman et al., 2015) in order to further improve our representations (Conneau et al., 2017). Since the only supervised training data is SNLI, the models can be used for a wide range of downstream supervised tasks that do not overlap with this dataset.[7]

4 Transfer Tasks

This section presents the data used for the transfer learning experiments and word embedding association tests (WEAT): (**MR**) Movie review sentiment on a five star scale (Pang and Lee, 2005); (**CR**) Sentiment of customer reviews (Hu and Liu, 2004); (**SUBJ**) Subjectivity of movie reviews and plot summaries (Pang and Lee, 2004); (**MPQA**) Phrase opinion polarity from news data (Wiebe et al., 2005); (**TREC**) Fine grained question classification sourced from TREC (Li and Roth, 2002); (**SST**) Binary phrase sentiment classification (Socher et al., 2013); (**STS Benchmark**) Semantic textual similarity (STS) between sentence pairs scored by Pearson r with human judgments (Cer et al., 2017); (**WEAT**) Word pairs from the psychology literature on implicit association tests (IAT) that are used to characterize model bias (Caliskan et al., 2017).[8] Table 1 gives the number of samples for each transfer task.

DATASET	TRAIN	DEV	TEST
SST	67,349	872	1,821
STS Bench	5,749	1,500	1,379
TREC	5,452	-	500
MR	-	-	10,662
CR	-	-	3,775
SUBJ	-	-	10,000
MPQA	-	-	10,606

Table 1: Transfer task evaluation sets.

5 Transfer Learning Models

For sentence classification transfer tasks, the output of the sentence encoders are provided to a task specific DNN. For the pairwise semantic similarity task, the similarity of sentence embeddings u and v is assessed using $-\arccos\left(\frac{uv}{||u||\,||v||}\right)$.[9]

5.1 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. For word-level transfer, we incorporate word embeddings from a word2vec skip-gram model trained on a corpus of news data (Mikolov et al., 2013). The pretrained word embeddings are included as input to two model types: a convolutional neural network model (CNN) (Kim, 2014); a DAN. The baselines that use pretrained word embeddings allow us to contrast word- vs. sentence-level transfer. Additional baseline CNN and DAN models are trained without using any pretrained word or sentence embeddings. For reference, we compare with InferSent (Conneau et al., 2017) and

[5] The Skip-Thought like task replaces the LSTM (Hochreiter and Schmidhuber, 1997) in the original formulation with a transformer model.

[6] SNLI (Bowman et al., 2015; Conneau et al., 2017)

[7] For questions on downstream evaluations possibly overlapping with the encoder training data, visit the TF Hub discussion board, https://groups.google.com/a/tensorflow.org/d/forum/hub, or e-mail the corresponding authors.

[8] For MR, CR, SUBJ, SST, and TREC we use the preparation of the data provided by Conneau et al. (2017).

[9] arccos converts cosine similarity into an angular distance that obeys the triangle inequality. We find that angular distance performs better on STS than cosine similarity.

Skip-Thought with layer normalization (Ba et al., 2016) on sentence-classification tasks. On the STS Benchmark, we compare with InferSent and the state-of-the-art neural STS systems CNN (HCTI) (Shao, 2017) and gConv (Yang et al., 2018).

5.2 Combined Transfer Models

We explore combining the sentence and word-level transfer models by concatenating their representations prior to the classification layers. For completeness, we report results providing the classification layers with the concatenating of the sentence-level embeddings and the representations produced by baseline models that do not make use of word-level transfer learning.

6 Experiments

Experiments use our most recent transformer and DAN encoding models.[10] Transfer task model hyperparamaters are tuned using a combination of Vizier (Golovin et al., 2017) and light manual tuning. When available, model hyperparameters are tuned using task dev sets. Otherwise, hyperparameters are tuned by cross-validation on task training data or the evaluation test data when neither training nor dev data are provided. Training repeats ten times for each task with randomly initialized weights and we report results by averaging across runs. Transfer learning is important when training data is limited. We explore using varying amounts of training data for SST. Contrasting the transformer and DAN encoders demonstrates trade-offs in model complexity and the training data required to reach a desired level of task accuracy. Finally, to assess bias in our encoders, we evaluate the strength of biased model associations on WEAT. We compare to Caliskan et al. (2017) who discovered that word embeddings reproduce human-like biases on implicit association tasks.

7 Results

Table 2 presents results on classification tasks. Using transformer sentence-level embeddings alone outperforms InferSent on MR, SUBJ, and TREC. The transformer sentence encoder also strictly outperforms the DAN encoder. Models that make use of just the transformer sentence-level embeddings tend to outperform all models that only use word-level transfer, with the exception of TREC and SST, on which CNN_{w2v} performs better. Transfer learning with DAN sentence embeddings tends to outperform a DAN with word-level transfer, except on MR and SST. Models with sentence- and word-level transfer often outperform similar models with sentence-level transfer alone.

[10] universal-sentence-encoder/2 (DAN); universal-sentence-encoder-large/3 (Transformer).

MODEL	MR	CR	SUBJ	MPQA	TREC	SST
Sentence Embedding Transfer Learning						
U_T	**82.2**	**84.2**	**95.5**	**88.1**	**93.2**	**83.7**
U_D	72.2	78.5	92.1	86.9	88.1	77.5
Word Embedding Transfer Learning						
CNN_{w2v}	**75.1**	**80.5**	**91.1**	80.3	**96.6**	**84.1**
DAN_{w2v}	74.7	75.3	90.2	**82.1**	83.5	80.6
*Sentence Embedding Transfer Learning + DNN/CNN **with** word-level transfer*						
U_T+CNN_{w2v}	80.1	85.2	**95.8**	**88.4**	**98.7**	85.3
U_T+DAN_{w2v}	**81.4**	**86.4**	93.7	87.5	97.0	**86.0**
U_D+CNN_{w2v}	76.7	82.0	91.2	85.2	97.1	85.1
U_D+DAN_{w2v}	76.4	81.0	94.0	88.0	92.6	82.2
*Sentence Embedding Transfer Learning + DNN/CNN **without** word-level transfer*						
U_T+CNN_{rnd}	**82.7**	**88.6**	93.6	**87.8**	98.5	**88.9**
U_T+DAN_{rnd}	80.6	84.8	**94.3**	86.0	**98.6**	86.2
U_D+CNN_{rnd}	78.0	82.9	90.2	**87.8**	96.2	83.2
U_D+DAN_{rnd}	76.4	84.9	94.0	85.3	98.1	86.2
Baselines with No Transfer Learning						
CNN_{rnd}	**76.5**	81.0	89.6	**82.2**	**97.9**	**85.0**
DAN_{rnd}	74.6	**81.2**	**91.8**	79.9	93.9	82.0
Prior Work						
InferSent	**81.1**	**86.3**	92.4	**90.2**	88.2	84.6
Skip Thght	79.4	83.1	**93.7**	89.3	-	-

Table 2: Classification tasks. U_T uses the transformer encoder for transfer learning, while U_D uses the DAN encoder. DAN/CNN_{w2v} use pre-trained w2v emb. DAN/CNN_{rnd} train rand. init. word emb. on the final classification task.

MODEL	DEV	TEST
Transformer Encoder	**0.802**	**0.766**
DAN Encoder	0.760	0.717
Prior Work		
gConv (Yang et al., 2018)	**0.835**	**0.808**
CNN (HCTI) (Shao, 2017)	0.834	0.784
InferSent (Conneau et al., 2017)	0.801	0.758

Table 3: STS Benchmark Pearson's *r*. Our prior gConv model (Yang et al., 2018) is a variant of our TF Hub transformer model tuned to STS.

Table 3 compares our models to strong baselines on the STS Benchmark. Our transformer embeddings outperform the sentence representations produced by InferSent. Moreover, computing similarity scores by directly comparing the representations produced by our encoders approaches

the performance of state-of-the-art neural models whose representations are fit to the STS task.

Table 4 illustrates transfer task performance for varying amounts of training data. With small quantities of training data, sentence-level transfer achieves surprisingly good performance. Using only 1k labeled examples and the transformer embeddings for sentence-level transfer *surpasses* the performance of transfer learning using InferSent on the full training set of 67.3k examples. Training with 1k labeled examples and the transformer sentence embeddings surpasses word-level transfer using the full training set, CNN_{w2v}, and approaches the performance of the best model without transfer learning trained on the complete dataset, CNN_{rnd}@67.3k. Transfer learning is not always helpful when there is enough task training data. However, we observe that our best performing model still makes use of transformer sentence-level transfer but combined with a CNN with no word-level transfer, U_T+CNN_{rnd}.

Table 5 contrasts Caliskan et al. (2017)'s findings on bias within GloVe embeddings with results from the transformer and DAN encoders. Similar to GloVe, our models reproduce human associations between flowers vs. insects and pleasantness vs. unpleasantness. However, our models demonstrate weaker associations than GloVe for probes targeted at revealing ageism, racism and sexism.[11] Differences in word association patterns can be attributed to training data composition and the mixture of tasks used to train the representations.

8 Resource Usage

This section describes memory and compute resource usage for the transformer and DAN sentence encoding models over different batch sizes and sentence lengths. Figure 1 plots model resource consumption against sentence length.[12]

Compute Usage The transformer model time complexity is $O(n^2)$ in sentence length, while the DAN model is $O(n)$. As seen in Figure 1 (a-b), for short sentences, the transformer encoding model is only moderately slower than the much simpler DAN model. However, compute time for transformer increases noticeably with sentence length. In contrast, the compute time for the DAN model stays nearly constant across different lengths. When running on GPU, even for large batches and longer sentence lengths, the transformer model still achieves performance that can be used within an interactive systems.

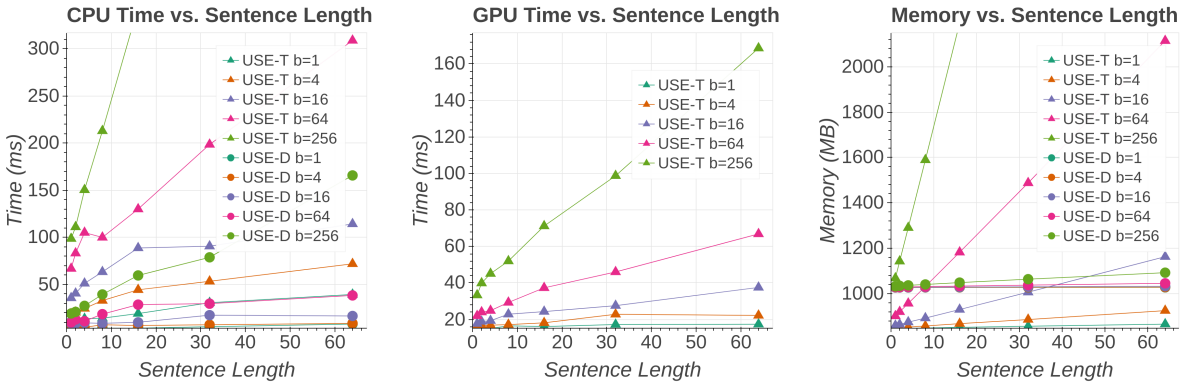
Memory Usage The transformer model space complexity also scales quadratically, $O(n^2)$, in sentence length, while the DAN is linear, $O(n)$. Similar to compute usage, memory for the transformer model increases quickly with sentence length, while the memory for the DAN model remains nearly constant. For the DAN model, memory is dominated by the parameters used to store the model unigram and bigram embeddings. Since the transformer model only stores unigrams, for

MODEL	SST 1K	SST 4K	SST 16K	SST 67.3K
Sentence Embedding Transfer Learning				
U_T	**84.8**	**84.8**	**84.8**	**83.7**
U_D	78.7	78.6	76.9	77.5
Word Embedding Transfer Learning				
CNN_{w2v}	**70.7**	73.8	**81.5**	**84.1**
DAN_{w2v}	67.5	**75.1**	78.4	80.6
*Sentence Embedding Transfer Learning + DNN/CNN **with** word-level transfer*				
U_T+CNN_{w2v}	84.9	84.9	**85.4**	85.3
U_T+DAN_{w2v}	**85.1**	**85.4**	85.0	**86.0**
U_D+CNN_{w2v}	78.6	79.7	80.9	85.1
U_D+DAN_{w2v}	78.7	79.1	81.6	82.2
*Sentence Embedding Transfer Learning + DNN/CNN **without** word-level transfer*				
U_T+CNN_{rnd}	83.1	**83.3**	84.9	**88.9**
U_T+DAN_{rnd}	**84.9**	84.2	**86.0**	86.2
U_D+CNN_{rnd}	77.5	77.9	81.3	83.2
U_D+DAN_{rnd}	78.5	78.8	82.5	86.2
Baselines with No Transfer Learning				
CNN_{rnd}	**68.9**	**74.6**	**81.5**	**85.0**
DAN_{rnd}	68.4	73.1	78.0	82.0
Prior Work				
InferSent	-	-	-	84.6

Table 4: SST performance varying the amount of training data. Model types are the same as Table 2. Using 1k examples, U_T transfer learning rivals models trained on the full training set, 67.3k.

[11] The development of our models did not target reducing bias. Researchers and developers are strongly encouraged to independently verify whether biases in their overall model or model components impacts their use case. For resources on ML fairness visit https://developers.google.com/machine-learning/fairness-overview/.

[12] All benchmark values are averaged over 25 runs that follow 5 priming runs. CPU and mem. benchmarks are performed on a machine with an Intel(R) Xeon(R) Platinum P-8136 CPU @ 2.00GHz CPU. GPU benchmarks use an Intel(R) Xeon(R) CPU E5-2696 v4 @ 2.20GHz CPU and NVIDIA Tesla P100 GPU.



(a) CPU Time vs. Sentence Length (b) GPU Time vs. Sentence Length (c) Memory vs. Sentence Length

Figure 1: Resource usage for the Universal Sentence Encoder DAN (USE-D) and Transformer (USE-T) models for different batch sizes and sentence lengths.

Target words	Attrib. words	Ref	GloVe		U. Enc. DAN		U. Enc. Trans.	
			d	p	d	p	d	p
Eur.- vs. Afr.-American names	Pleasant vs. Unpleasant	a	1.41	10^{-8}	0.36	0.04	0.22	0.14
Eur.- vs. Afr.-American names	Pleasant vs.Unpleasant from (a)	b	1.50	10^{-4}	-0.37	0.87	0.21	0.27
Eur.- vs. Afr.-American names	Pleasant vs. Unpleasant from (c)	b	1.28	10^{-3}	0.72	0.02	0.93	10^{-2}
Male vs. female names	Career vs. family	c	1.81	10^{-3}	0.02	0.48	0.95	0.03
Math vs. arts	Male vs. female terms	c	1.06	0.02	0.59	0.12	0.12	0.41
Science vs. arts	Male vs. female terms	d	1.24	10^{-2}	0.24	0.32	-0.21	0.67
Mental vs. physical disease	Temporary vs. permanent	e	1.38	10^{-2}	1.60	10^{-2}	0.42	0.23
Young vs old peoples names	Pleasant vs unpleasant	c	1.21	10^{-2}	1.01	0.02	0.06	0.46
Flowers vs. Insects	Pleasant vs. Unpleasant	a	1.50	10^{-7}	1.38	10^{-6}	1.47	10^{-7}
Instruments vs. Weapons	Pleasant vs Unpleasant	a	1.53	10^{-7}	1.44	10^{-7}	1.65	10^{-7}

Table 5: WEAT for GloVe vs. our DAN and transformer encoding models. Effect size is reported as Cohen's d over the mean cosine similarity scores across grouped attribute words. Statistical significance uses one-tailed p-scores. 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).

very short sequences transformer requires almost half as much memory as the DAN model.

9 Conclusion

Our encoding models provide sentence-level embeddings that demonstrate strong transfer performance on a number of NLP tasks. The encoding models make different trade-offs regarding accuracy and model complexity that should be considered when choosing the best one for a particular application. Overall, our sentence-level embeddings tend to surpass the performance of transfer using word-level embeddings alone. Models that make use of sentence- and word-level transfer often achieve the best performance. Sentence-level transfer using our models can be exceptionally helpful when limited training data is available. The pre-trained encoding models are publicly available for research and use in industry applications that can benefit from a better understanding of natural language.

Acknowledgments

We thank our teammates from Descartes, Ai.h and other Google groups for their feedback and suggestions. Special thanks goes to Ben Packer and Yoni Halpern for implementing the WEAT assessments and discussions on model bias.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of USENIX OSDI'16*.

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization. *CoRR*, abs/1607.06450.

Marianne Bertrand and Sendhil Mullainathan. 2004. Are emily and greg more employable than lakisha and jamal? a field experiment on labor market discrimination. *The American Economic Review*, 94(4).

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.

Aylin Caliskan, Joanna J. Bryson, and Arvind Narayanan. 2017. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of SemEval-2017*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D. Sculley. 2017. Google vizier: A service for black-box optimization. In *Proceedings of KDD '17*.

Anthony G. Greenwald, Debbie E. McGhee, and Jordan L. K. Schwartz. 1998. Measuring individual differences in implicit cognition: the implicit association test. *Journal of personality and social psychology*, 74(6).

Matthew Henderson, Rami Al-Rfou, Brian Strope, Yun-Hsuan Sung, László Lukács, Ruiqi Guo, Sanjiv Kumar, Balint Miklos, and Ray Kurzweil. 2017. Efficient natural language response suggestion for smart reply. *CoRR*, abs/1705.00652.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of KDD '04*.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of ACL/IJCNLP*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *In Proceedings of NIPS*.

Xin Li and Dan Roth. 2002. Learning question classifiers. In *Proceedings of COLING '02*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*.

Lindsey L. Monteith and Jeremy W. Pettit. 2011. Implicit and explicit stigmatizing attitudes and stereotypes about depression. *Journal of Social and Clinical Psychology*, 30(5).

Brian A. Nosek, Mahzarin R. Banaji, and Anthony G. Greenwald. 2002a. Harvesting implicit group attitudes and beliefs from a demonstration web site. *Group Dynamics*, 6(1).

Brian A. Nosek, Mahzarin R. Banaji, and Anthony G Greenwald. 2002b. Math = male, me = female, therefore math me. *Journal of Personality and Social Psychology,*, 83(1).

Bo Pang and Lillian Lee. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL'04), Main Volume*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL'05*.

Yang Shao. 2017. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 130–133.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NIPS*.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation*, 39(2):165–210.

Yinfei Yang, Steve Yuan, Daniel Cer, Sheng yi Kong, Noah Constant, Petr Pilar, Heming Ge, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Learning semantic textual similarity from conversations. *Proceedings of RepL4NLP workshop at ACL*.