

SentEval: An Evaluation Toolkit for Universal Sentence Representations

Alexis Conneau* and Douwe Kiela

Facebook Artificial Intelligence Research
{aconneau,dkiela}@fb.com

Abstract

We introduce SentEval, a toolkit for evaluating the quality of universal sentence representations. SentEval encompasses a variety of tasks, including binary and multi-class classification, natural language inference and sentence similarity. The set of tasks was selected based on what appears to be the community consensus regarding the appropriate evaluations for universal sentence representations. The toolkit comes with scripts to download and preprocess datasets, and an easy interface to evaluate sentence encoders. The aim is to provide a fairer, less cumbersome and more centralized way for evaluating sentence representations.

Keywords: representation learning, evaluation

1. Introduction

Following the recent word embedding upheaval, one of NLP’s next challenges has become the hunt for universal general-purpose sentence representations. What distinguishes these representations, or embeddings, is that they are not necessarily trained to perform well on one specific task. Rather, their value lies in their transferability, i.e., their ability to capture information that can be of use in any kind of system or pipeline, on a variety of tasks.

Word embeddings are particularly useful in cases where there is limited training data, leading to sparsity and poor vocabulary coverage, which in turn lead to poor generalization capabilities. Similarly, sentence embeddings (which are often built on top of word embeddings) can be used to further increase generalization capabilities, composing unseen combinations of words and encoding grammatical constructions that are not present in the task-specific training data. Hence, high-quality universal sentence representations are highly desirable for a variety of downstream NLP tasks.

The evaluation of general-purpose word and sentence embeddings has been problematic (Chiu et al., 2016; Faruqui et al., 2016), leading to much discussion about the best way to go about it¹. On the one hand, people have measured performance on intrinsic evaluations, e.g. of human judgments of word or sentence similarity ratings (Agirre et al., 2012; Hill et al., 2016b) or of word associations (Vulić et al., 2017). On the other hand, it has been argued that the focus should be on downstream tasks where these representations would actually be applied (Ettinger et al., 2016; Nayak et al., 2016). In the case of sentence representations, there is a wide variety of evaluations available, many from before the “embedding era”, that can be used to assess representational quality on that particular task. Over the years, something of a consensus has been established, mostly based on the evaluations in seminal papers such as SkipThought (Kiros et al., 2015), concerning what evaluations to use. Recent works in which various alternative sentence encoders are compared use a similar set of tasks

(Hill et al., 2016a; Conneau et al., 2017).

Implementing pipelines for this large set of evaluations, each with its own peculiarities, is cumbersome and induces unnecessary wheel reinventions. Another well-known problem with the current status quo, where everyone uses their own evaluation pipeline, is that different preprocessing schemes, evaluation architectures and hyperparameters are used. The datasets are often small, meaning that minor differences in evaluation setup may lead to very different outcomes, which implies that results reported in papers are not always fully comparable.

In order to overcome these issues, we introduce SentEval²: a toolkit that makes it easy to evaluate universal sentence representation encoders on a large set of evaluation tasks that has been established by community consensus.

2. Aims

The aim of SentEval is to make research on universal sentence representations fairer, less cumbersome and more centralized. To achieve this goal, SentEval encompasses the following:

- one central set of evaluations, based on what appears to be community consensus;
- one common evaluation pipeline with fixed standard hyperparameters, apart from those tuned on validation sets, in order to avoid discrepancies in reported results; and
- easy access for anyone, meaning: a straightforward interface in Python, and scripts necessary to download and preprocess the relevant datasets.

In addition, we provide examples of models, such as a simple bag-of-words model. These could potentially also be used to extrinsically evaluate the quality of word embeddings in NLP tasks.

3. Evaluations

Our aim is to obtain general-purpose sentence embeddings that capture generic information, which should be useful

*LIUM, Université Le Mans

¹See also recent workshops on evaluating representations for NLP, e.g. RepEval: <https://repeval2017.github.io/>

²<https://github.com/facebookresearch/SentEval>

name	N	task	C	examples	label(s)
MR	11k	sentiment (movies)	2	“Too slow for a younger crowd , too shallow for an older one.”	neg
CR	4k	product reviews	2	“We tried it out christmas night and it worked great .”	pos
SUBJ	10k	subjectivity/objectivity	2	“A movie that doesn’t aim too high , but doesn’t need to.”	subj
MPQA	11k	opinion polarity	2	“don’t want”; “would like to tell”;	neg, pos
TREC	6k	question-type	6	“What are the twin cities ?”	LOC:city
SST-2	70k	sentiment (movies)	2	“Audrey Tautou has a knack for picking roles that magnify her [...]”	pos
SST-5	12k	sentiment (movies)	5	“nothing about this movie works.”	0

Table 1: **Classification tasks.** C is the number of classes and N is the number of samples.


name	N	task	output	premise	hypothesis	label
SNLI	560k	NLI	3	“A small girl wearing a pink jacket is riding on a carousel.”	“The carousel is moving.”	entailment
SICK-E	10k	NLI	3	“A man is sitting on a chair and rubbing his eyes”	“There is no man sitting on a chair and rubbing his eyes”	contradiction
SICK-R	10k	STS	[0, 5]	“A man is singing a song and playing the guitar”	“A man is opening a package that contains headphones”	1.6
STS14	4.5k	STS	[0, 5]	“Liquid ammonia leak kills 15 in Shanghai”	“Liquid ammonia leak kills at least 15 in Shanghai”	4.6
MRPC	5.7k	PD	2	“The procedure is generally performed in the second or third trimester.”	“The technique is used during the second and, occasionally, third trimester of pregnancy.”	paraphrase
COCO	565k	ICR	sim		“A group of people on some horses riding through the beach.”	rank

Table 2: **Natural Language Inference and Semantic Similarity tasks.** NLI labels are contradiction, neutral and entailment. STS labels are scores between 0 and 5. PD=paraphrase detection, ICR=image-caption retrieval.

for a broad set of tasks. To evaluate the quality of these representations, we use them as features in various transfer tasks.

Binary and multi-class classification We use a set of binary classification tasks (see Table 1) that covers various types of sentence classification, including sentiment analysis (MR and both binary and fine-grained SST) (Pang and Lee, 2005; Socher et al., 2013), question-type (TREC) (Voorhees and Tice, 2000), product reviews (CR) (Hu and Liu, 2004), subjectivity/objectivity (SUBJ) (Pang and Lee, 2004) and opinion polarity (MPQA) (Wiebe et al., 2005). We generate sentence vectors and classifier on top, either in the form of a Logistic Regression or an MLP. For MR, CR, SUBJ and MPQA, we use nested 10-fold cross-validation, for TREC cross-validation and for SST standard validation.

Entailment and semantic relatedness We also include the SICK dataset (Marelli et al., 2014) for entailment (SICK-E), and semantic relatedness datasets including SICK-R and the STS Benchmark dataset (Cer et al., 2017). For semantic relatedness, which consists of predicting a semantic score between 0 and 5 from two input sentences, we follow the approach of Tai et al. (2015a) and learn to predict the probability distribution of relatedness scores. SentEval reports Pearson and Spearman correlation. In addition, we include the SNLI dataset (Bowman et al., 2015), a collection of 570k human-written English supporting the task of natural language inference (NLI), also known as rec-

ognizing textual entailment (RTE) which consists of predicting whether two input sentences are entailed, neutral or contradictory. SNLI was specifically designed to serve as a benchmark for evaluating text representation learning methods.

Semantic Textual Similarity While semantic relatedness requires training a model on top of the sentence embeddings, we also evaluate embeddings on the unsupervised SemEval tasks. These datasets include pairs of sentences taken from news articles, forum discussions, news conversations, headlines, image and video descriptions labeled with a similarity score between 0 and 5. The goal is to evaluate how the cosine distance between two sentences correlate with a human-labeled similarity score through Pearson and Spearman correlations. We include STS tasks from 2012 (Agirre et al., 2012), 2013⁴ (Agirre et al., 2013), 2014 (Agirre et al., 2014), 2015 (Agirre et al., 2015) and 2016 (Agirre et al., 2016). Each of these tasks includes several subtasks. SentEval reports both the average and the weighted average (by number of samples in each subtask) of the Pearson and Spearman correlations.

Paraphrase detection The Microsoft Research Paraphrase Corpus (MRPC) (Dolan et al., 2004) is composed of pairs of sentences which have been extracted from news sources on the Web. Sentence pairs have been human-annotated according to whether they capture a paraphrase/semantic equivalence relationship. We use the same

³Antonio Rivera - CC BY 2.0 - flickr

⁴Due to License issues, we do not include the SMT subtask.

approach as with SICK-E, except that our classifier has only 2 classes, i.e., the aim is to predict whether the sentences are paraphrases or not.

Caption-Image retrieval The caption-image retrieval task evaluates joint image and language feature models (Lin et al., 2014). The goal is either to rank a large collection of images by their relevance with respect to a given query caption (Image Retrieval), or ranking captions by their relevance for a given query image (Caption Retrieval). The COCO dataset provides a training set of 113k images with 5 captions each. The objective consists of learning a caption-image compatibility score $\mathcal{L}_{\text{cir}}(x, y)$ from a set of aligned image-caption pairs as training data. We use a pairwise ranking-loss $\mathcal{L}_{\text{cir}}(x, y)$:

$$\sum_y \sum_k \max(0, \alpha - s(Vy, Ux) + s(Vy, Ux_k)) + \sum_x \sum_{k'} \max(0, \alpha - s(Ux, Vy) + s(Ux, Vy_{k'})),$$

where (x, y) consists of an image y with one of its associated captions x , $(y_k)_k$ and $(y_{k'})_{k'}$ are negative examples of the ranking loss, α is the margin and s corresponds to the cosine similarity. U and V are learned linear transformations that project the caption x and the image y to the same embedding space. We measure Recall@K, with $K \in \{1, 5, 10\}$, i.e., the percentage of images/captions for which the corresponding caption/image is one of the first K retrieved; and median rank. We use the same splits as Karpathy and Fei-Fei (2015), i.e., we use 113k images (each containing 5 captions) for training, 5k images for validation and 5k images for test. For evaluation, we split the 5k images in 5 random sets of 1k images on which we compute the mean $R@1$, $R@5$, $R@10$ and median (Med r) over the 5 splits. We include 2048-dimensional pretrained ResNet-101 (He et al., 2016) features for all images.

4. Usage and Requirements

Our evaluations comprise **two different types: ones where we need to learn on top of the provided sentence representations (e.g. classification/regression) and ones where we simply take the cosine similarity between the two representations, as in the STS tasks**. In the binary and multi-class classification tasks, we fit either a Logistic Regression classifier or an MLP with one hidden layer on top of the sentence representations. For the natural language inference tasks, where we are given two sentences u and v , we provide the classifier with the input $\langle u, v, |u - v|, u * v \rangle$. To fit the Pytorch models, we use Adam (Kingma and Ba, 2014), with a batch size 64. We tune the L2 penalty of the classifier with grid-search on the validation set. When using SentEval, two functions should be implemented by the user:

- `prepare(params, dataset)`: sees the whole dataset and applies any necessary preprocessing, such as constructing a lookup table of word embeddings (this function is optional); and
- `batcher(params, batch)`: given a batch of input sentences, returns an array of the sentence embeddings for the respective inputs.

The main `batcher` function allows the user to encode text sentences using any Python framework. For example, the `batcher` function might be a wrapper around a model written in Pytorch, TensorFlow, Theano, DyNet, or any other framework⁵. To illustrate the use, here is an example of what an evaluation script looks like, having defined the `prepare` and `batcher` functions:

```
import senteval
se = senteval.engine.SE(
    params, batcher, prepare)
transfer_tasks = ['MR', 'CR']
results = se.eval(transfer_tasks)
```

Parameters Both functions make use of a `params` object, which contains the settings of the network and the evaluation. SentEval has several parameters that influence the evaluation procedure. These include the following:

- `task_path` (str, required): path to the data.
- `seed` (int): random seed for reproducibility.
- `batch_size` (int): size of minibatch of text sentences provided to batcher (sentences are sorted by length).
- `kfold` (int): k in the kfold-validation (default: 10).

The default config is:

```
params = {'task_path': PATH_TO_DATA,
          'usepytorch': True,
          'kfold': 10}
```

We also give the user the ability to customize the classifier used for the classification tasks.

Classifier To be comparable to the results published in the literature, users should use the following parameters for Logistic Regression:

```
params['classifier'] =
    {'nhid': 0, 'optim': 'adam',
     'batch_size': 64, 'tenacity': 5,
     'epoch_size': 4}
```

The parameters of the classifier include:

- `nhid` (int): number of hidden units of the MLP; if `nhid > 0`, a Multi-Layer Perceptron with one hidden layer and a Sigmoid nonlinearity is used.
- `optim` (str): classifier optimizer (default: adam).
- `batch_size` (int): batch size for training the classifier (default: 64).
- `tenacity` (int): stopping criterion; maximum number of times the validation error does not decrease.
- `epoch_size` (int): number of passes through the training set for one epoch.

⁵Or any other programming language, as long as the vectors can be passed to, or loaded from, code written in Python.

Model	MR	CR	SUBJ	MPQA	SST-2	SST-5	TREC	MRPC	SICK-E
<i>Representation learning (transfer)</i>									
GloVe LogReg	77.4	78.7	91.2	87.7	80.3	44.7	83.0	72.7/81.0	78.5
GloVe MLP	77.7	79.9	92.2	88.7	82.3	45.4	85.2	73.0/80.9	79.0
fastText LogReg	78.2	80.2	91.8	88.0	82.3	45.1	83.4	74.4/82.4	78.9
fastText MLP	78.0	81.4	92.9	88.5	84.0	45.1	85.6	74.4/82.3	80.2
SkipThought	79.4	83.1	93.7	89.3	82.9	-	88.4	72.4/81.6	79.5
InferSent	81.1	86.3	92.4	90.2	84.6	46.3	88.2	76.2/83.1	86.3
<i>Supervised methods directly trained for each task (no transfer)</i>									
SOTA	83.1 ¹	86.3 ¹	95.5 ¹	93.3 ¹	89.5 ²	52.4 ²	96.1 ²	80.4/85.9 ³	84.5 ⁴

Table 3: Transfer test results for various baseline methods. We include supervised results trained directly on each task (no transfer). Results ¹ correspond to AdaSent (Zhao et al., 2015), ² to BLSTM-2DCNN (Zhou et al., 2016), ³ to TF-KLD (Ji and Eisenstein, 2013) and ⁴ to Illinois-LH system (Lai and Hockenmaier, 2014).

- dropout (float): dropout rate in the case of MLP.

For use cases where there are multiple calls to SentEval, e.g when evaluating the sentence encoder at every epoch of training, we propose the following prototyping set of parameters, which will lead to slightly worse results but will make the evaluation significantly faster:

```
params['classifier'] =
    {'nhid': 0, 'optim': 'rmsprop',
     'batch_size': 128, 'tenacity': 3,
     'epoch_size': 2}
```

You may also pass additional parameters to the `params` object in order which will further be accessible from the `prepare` and `batcher` functions (e.g a pretrained model).

Datasets In order to obtain the data and preprocess it so that it can be fed into SentEval, we provide the `get_transfer_data.bash` script in the data directory. The script fetches the different datasets from their known locations, unpacks them and preprocesses them. We tokenize each of the datasets with the MOSES tokenizer (Koehn et al., 2007) and convert all files to UTF-8 encoding. Once this script has been executed, the `task_path` parameter can be set to indicate the path of the data directory.

Requirements SentEval is written in Python. In order to run the evaluations, the user will need to install numpy, scipy and recent versions of pytorch and scikit-learn. In order to facilitate research where no GPUs are available, we offer for the evaluations to be run on CPU (using scikit-learn) where possible. For the bigger datasets, where more complicated models are often required, for instance STS Benchmark, SNLI, SICK-R and the image-caption retrieval tasks, we recommend pytorch models on a single GPU.

5. Baselines

Several baseline models are evaluated in Table 3:

- Continuous bag-of-words embeddings (average of word vectors). We consider the most commonly used pretrained word vectors available, namely the fastText (Mikolov et al., 2017) and the GloVe (Pennington et al., 2014) vectors trained on CommonCrawl.

- SkipThought vectors (Ba et al., 2016)

- InferSent vectors (Conneau et al., 2017)

In addition to these methods, we include the results of current state-of-the-art methods for which both the encoder and the classifier are trained on each task (no transfer). For GloVe and fastText bag-of-words representations, we report the results for Logistic Regression and Multi-Layer Perceptron (MLP). For the MLP classifier, we tune the dropout rate and the number of hidden units in addition to the L2 regularization. We do not observe any improvement over Logistic Regression for methods that already have a large embedding size (4096 for InferSent and 4800 for SkipThought). On most transfer tasks, supervised methods that are trained directly on each task still outperform transfer methods. Our hope is that SentEval will help the community build sentence representations with better generalization power that can outperform both the transfer and the supervised methods.

6. Conclusion

Universal sentence representations are a hot topic in NLP research. Making use of a generic sentence encoder allows models to generalize and transfer better, even when trained on relatively small datasets, which makes them highly desirable for downstream NLP tasks.

We introduced SentEval as a fair, straightforward and centralized toolkit for evaluating sentence representations. We have aimed to make evaluation as easy as possible: sentence encoders can be evaluated by implementing a simple Python interface, and we provide a script to download the necessary evaluation datasets. In future work, we plan to enrich SentEval with additional tasks as the consensus on the best evaluation for sentence embeddings evolves. In particular, tasks that probe for specific linguistic properties of the sentence embeddings (Shi et al., 2016; Adi et al., 2017) are interesting directions towards understanding how the encoder understands language. We hope that our toolkit will be used by the community in order to ensure that fully comparable results are published in research papers.

Model	SST'12	SST'13	SST'14	SST'15	SST'16	SICK-R	SST-B
<i>Representation learning (transfer)</i>							
GloVe BoW	52.1	49.6	54.6	56.1	51.4	79.9	64.7
fastText BoW	58.3	57.9	64.9	67.6	64.3	82.0	70.2
SkipThought-LN	30.8	24.8	31.4	31.0	-	85.8	72.1
InferSent	59.2	58.9	69.6	71.3	71.5	88.3	75.6
Char-phrase	66.1	57.2	74.7	76.1	-	-	-
<i>Supervised methods directly trained for each task (no transfer)</i>							
PP-Proj	60.0 ¹	56.8 ¹	71.3 ¹	74.8 ¹	-	86.8 ²	-

Table 4: Evaluation of sentence representations on the semantic textual similarity benchmarks. Numbers reported are Pearson correlations x100. We use the average of Pearson correlations for STS'12 to STS'16 which are composed of several subtasks. Charagram-phrase numbers were taken from (Wieting et al., 2016). Results ¹ correspond to PP-Proj (Wieting et al., 2015) and ² from Tree-LSTM (Tai et al., 2015b).

7. Bibliographical References

- Adi, Y., Kermany, E., Belinkov, Y., Lavi, O., and Goldberg, Y. (2017). Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. In *Proceedings of ICLR Conference Track*, Toulon, France. Published online: <https://openreview.net/group?id=ICLR.cc/2017/conference>.
- Agirre, E., Diab, M., Cer, D., and Gonzalez-Agirre, A. (2012). Semeval-2012 task 6: A pilot on semantic textual similarity. In *Proceedings of Semeval-2012*, pages 385–393.
- Agirre, E., Cer, D., Diab, M., Gonzalez-agirre, A., and Guo, W. (2013). sem 2013 shared task: Semantic textual similarity, including a pilot on typed-similarity. In *In *SEM 2013: The Second Joint Conference on Lexical and Computational Semantics. Association for Computational Linguistics*.
- Agirre, E., Baneab, C., Cardiec, C., Cerd, D., Diabe, M., Gonzalez-Agirre, A., Guof, W., Mihalceab, R., Rigaua, G., and Wiebeg, J. (2014). Semeval-2014 task 10: Multilingual semantic textual similarity. *SemEval 2014*, page 81.
- Agirre, E., Banea, C., Cardie, C., Cer, D. M., Diab, M. T., Gonzalez-Agirre, A., Guo, W., Lopez-Gazpio, I., Maritxalar, M., Mihalcea, R., et al. (2015). Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval@ NAACL-HLT*, pages 252–263.
- Agirre, E., Baneab, C., Cerd, D., Diabe, M., Gonzalez-Agirre, A., Mihalceab, R., Rigaua, G., Wiebef, J., and Donostia, B. C. (2016). Semeval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. *Proceedings of SemEval*, pages 497–511.
- Ba, J. L., Kiros, J. R., and Hinton, G. E. (2016). Layer normalization. *Advances in neural information processing systems (NIPS)*.
- Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. (2015). A large annotated corpus for learning natural language inference. In *Proceedings of EMNLP*.
- Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. (2017). Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.
- Chiu, B., Korhonen, A., and Pyysalo, S. (2016). Intrinsic evaluation of word vectors fails to predict extrinsic performance. In *First Workshop on Evaluating Vector Space Representations for NLP (RepEval)*.
- Conneau, A., Kiela, D., Schwenk, H., Barrault, L., and Bordes, A. (2017). Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of EMNLP*, Copenhagen, Denmark.
- Dolan, B., Quirk, C., and Brockett, C. (2004). Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of ACL*, page 350.
- Ettinger, A., Elgohary, A., and Resnik, P. (2016). Probing for semantic evidence of composition by means of simple classification tasks. In *First Workshop on Evaluating Vector Space Representations for NLP (RepEval)*, page 134.
- Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems with evaluation of word embeddings using word similarity tasks. *arXiv preprint arXiv:1605.02276*.
- He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of CVPR*.
- Hill, F., Cho, K., and Korhonen, A. (2016a). Learning distributed representations of sentences from unlabelled data. In *Proceedings of NAACL*.
- Hill, F., Reichart, R., and Korhonen, A. (2016b). Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.
- Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of SIGKDD*, pages 168–177.
- Ji, Y. and Eisenstein, J. (2013). Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Karpathy, A. and Fei-Fei, L. (2015). Deep visual-semantic alignments for generating image descriptions. In *Proceedings of CVPR*, pages 3128–3137.
- Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. In *Proceedings of the 3rd*

- International Conference on Learning Representations (ICLR)*.
- Kiros, R., Zhu, Y., Salakhutdinov, R. R., Zemel, R., Urtasun, R., Torralba, A., and Fidler, S. (2015). Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.
- Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Lai, A. and Hockenmaier, J. (2014). Illinois-lh: A denotational and distributional approach to semantics. *Proc. SemEval*, 2:5.
- Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer International Publishing.
- Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., and Zamparelli, R. (2014). A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of LREC*.
- Mikolov, T., Grave, E., Bojanowski, P., Puhersch, C., and Joulin, A. (2017). Advances in pre-training distributed word representations.
- Nayak, N., Angeli, G., and Manning, C. D. (2016). Evaluating word embeddings using a representative suite of practical tasks. In *First Workshop on Evaluating Vector Space Representations for NLP (RepEval)*, page 19.
- Pang, B. and Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of ACL*, page 271.
- Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of ACL*, pages 115–124.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, volume 14, pages 1532–1543.
- Shi, X., Padhi, I., and Knight, K. (2016). Does string-based neural MT learn source syntax? In *Proceedings of EMNLP*, pages 1526–1534, Austin, Texas.
- Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., Potts, C., et al. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of EMNLP*, pages 1631–1642.
- Tai, K. S., Socher, R., and Manning, C. D. (2015a). Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of ACL*.
- Tai, K. S., Socher, R., and Manning, C. D. (2015b). Improved semantic representations from tree-structured long short-term memory networks. *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.
- Voorhees, E. M. and Tice, D. M. (2000). Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 200–207. ACM.
- Vulić, I., Kiela, D., and Korhonen, A. (2017). Evaluation by association: A systematic study of quantitative word association evaluation. In *Proceedings of EACL*, volume 1, pages 163–175.
- Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2):165–210.
- Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2015). Towards universal paraphrastic sentence embeddings. *Proceedings of the 4th International Conference on Learning Representations (ICLR)*.
- Wieting, J., Bansal, M., Gimpel, K., and Livescu, K. (2016). Charagram: Embedding words and sentences via character n-grams. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.
- Zhao, H., Lu, Z., and Poupart, P. (2015). Self-adaptive hierarchical sentence model. In *Proceedings of the 24th International Conference on Artificial Intelligence, IJCAI'15*, pages 4069–4076. AAAI Press.
- Zhou, P., Qi, Z., Zheng, S., Xu, J., Bao, H., and Xu, B. (2016). Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*.