

# An Embarrassingly Simple Approach for Transfer Learning from Pretrained Language Models

Alexandra Chronopoulou<sup>1</sup>, Christos Baziotis<sup>1</sup>, Alexandros Potamianos<sup>1,2,3</sup>

<sup>1</sup>School of ECE, National Technical University of Athens, Athens, Greece

<sup>2</sup>Signal Analysis and Interpretation Laboratory (SAIL), USC, Los Angeles, USA

<sup>3</sup>Behavioral Signal Technologies, Los Angeles, USA

el12068@central.ntua.gr, cbaziotis@mail.ntua.gr

potam@central.ntua.gr

## Abstract

A growing number of state-of-the-art transfer learning methods employ language models pretrained on large generic corpora. In this paper we present a conceptually simple and effective transfer learning approach that addresses the problem of catastrophic forgetting. Specifically, we combine the task-specific optimization function with an auxiliary language model objective, which is adjusted during the training process. This preserves language regularities captured by language models, while enabling sufficient adaptation for solving the target task. Our method does not require pre-training or finetuning separate components of the network and we train our models end-to-end in a single step. We present results on a variety of challenging affective and text classification tasks, surpassing well established transfer learning methods with greater level of complexity.

## 1 Introduction

Pretrained word representations captured by Language Models (LMs) have recently become popular in Natural Language Processing (NLP). Pretrained LMs encode contextual information and high-level features of language, modeling syntax and semantics, producing state-of-the-art results across a wide range of tasks, such as named entity recognition (Peters et al., 2017), machine translation (Ramachandran et al., 2017) and text classification (Howard and Ruder, 2018).

However, in cases where contextual embeddings from language models are used as additional features (e.g. ELMo (Peters et al., 2018)), results come at a high computational cost and require task-specific architectures. At the same time, approaches that rely on fine-tuning a LM to the task at hand (e.g. ULMFiT (Howard and Ruder, 2018)) depend on pretraining the model on an extensive vocabulary and on employing a sophisticated

slanted triangular learning rate scheme to adapt the parameters of the LM to the target dataset.

We propose a simple and effective transfer learning approach, that leverages LM contextual representations and does not require any elaborate scheduling schemes during training. We initially train a LM on a Twitter corpus and then transfer its weights. We add a task-specific recurrent layer and a classification layer. The transferred model is trained end-to-end using an auxiliary LM loss, which allows us to explicitly control the weighting of the pretrained part of the model and ensure that the distilled knowledge it encodes is preserved.

Our contributions are summarized as follows:

- 1) We show that transfer learning from language models can achieve competitive results, while also being intuitively simple and computationally effective.
- 2) We address the problem of catastrophic forgetting, by adding an auxiliary LM objective and using an unfreezing method.
- 3) Our results show that our approach is competitive with more sophisticated transfer learning methods. We make our code widely available.<sup>1</sup>

## 2 Related Work

Unsupervised pretraining has played a key role in deep neural networks, building on the premise that representations learned for one task can be useful for another task. In NLP, pretrained word vectors (Mikolov et al., 2013; Pennington et al., 2014) are widely used, improving performance in various downstream tasks, such as part-of-speech tagging (Collobert et al., 2011) and question answering (Xiong et al., 2016). These pretrained word vectors serve as initialization of the embedding layer and remain frozen during training, while our pretrained language model also initializes the hidden layers of the model and is fine-tuned to each

<sup>1</sup>[github.com/alexandra-chron/siat1](https://github.com/alexandra-chron/siat1)

classification task.

Aiming to learn from unlabeled data, Dai and Le (2015) use unsupervised objectives such as sequence autoencoding and language modeling for as pretraining methods. The pretrained model is then fine-tuned to the target task. However, the fine-tuning procedure of the language model to the target task does not include an auxiliary objective. Ramachandran et al. (2017) also pretrain encoder-decoder pairs using language models and fine-tune them to a specific task, using an auxiliary language modeling objective to prevent catastrophic forgetting. This approach, nevertheless, is only evaluated on machine translation tasks; moreover, the seq2seq (Sutskever et al., 2014) and language modeling losses are weighted equally throughout training. By contrast, we propose a weighted sum of losses, where the language modeling contribution gradually decreases. ELMo embeddings (Peters et al., 2018) are obtained from language models and improve the results in a variety of tasks as additional contextual representations. However, ELMo embeddings rely on character-level models, whereas our approach uses a word-level LM. They are, furthermore, concatenated to pretrained word vectors and remain fixed during training. We instead propose a fine-tuning procedure, aiming to adjust a generic architecture to different end tasks.

Moreover, BERT (Devlin et al., 2018) pretrains language models and fine-tunes them on the target task. An auxiliary task (next sentence prediction) is used to enhance the representations of the LM. BERT fine-tunes masked bi-directional LMs. Nevertheless, we are limited to a uni-directional model. Training BERT requires vast computational resources, while our model only requires 1 GPU. We note that our approach is not orthogonal to BERT and could be used to improve it, by adding an auxiliary LM objective and weighing its contribution.

Towards the same direction, ULMFiT (Howard and Ruder, 2018) shows impressive results on a variety of tasks by employing pretrained LMs. The proposed pipeline requires three distinct steps, that include (1) pretraining the LM, (2) fine-tuning it on a target dataset with an elaborate scheduling procedure and (3) transferring it to a classification model. Our proposed model is closely related to ULMFiT. However, ULMFiT trains a LM and fine-tunes it to the target dataset, before transferring it to a classification model. While fine-tuning

the LM to the target dataset, the metric (e.g. accuracy) that we intend to optimize cannot be observed. We propose adopting a multi-task learning perspective, via the addition of an auxiliary LM loss to the transferred model, to control the loss of the pretrained and the new task simultaneously. The intuition is that we should avoid catastrophic forgetting, but at the same time allow the LM to distill the knowledge of the prior data distribution and keep the most useful features.

Multi-Task Learning (MTL) via hard parameter sharing (Caruana, 1993) in neural networks has proven to be effective in many NLP problems (Collobert and Weston, 2008). More recently, alternative approaches have been suggested that only share parameters across lower layers (Sogaard and Goldberg, 2016). By introducing part-of-speech tags at the lower levels of the network, the proposed model achieves competitive results on chunking and CCG super tagging. Our auxiliary language model objective follows this line of thought and intends to boost the performance of the higher classification layer.

### 3 Our Model

We introduce **SiATL**, which stands for **Single-step Auxiliary loss Transfer Learning**. In our proposed approach, we first train a LM. We then transfer its weights and add a task-specific recurrent layer to the final classifier. We also employ an auxiliary LM loss to avoid catastrophic forgetting.

**LM Pretraining.** We train a word-level language model, which consists of an embedding LSTM layer (Hochreiter and Schmidhuber, 1997), 2 hidden LSTM layers and a linear layer. We want to minimize the negative log-likelihood of the LM:

$$L(\hat{p}) = -\frac{1}{N} \sum_{n=1}^N \sum_{t=1}^{T^n} \log \hat{p}(x_t^n | x_1^n, \dots, x_{t-1}^n) \quad (1)$$

where  $\hat{p}(x_t^n | x_1^n, \dots, x_{t-1}^n)$  is the distribution of the  $t^{th}$  word in the  $n^{th}$  sentence given the  $t - 1$  words preceding it and  $N$  is total number of sentences.

**Transfer & auxiliary loss.** We transfer the weights of the pretrained model and add one LSTM with a self-attention mechanism (Lin et al., 2017; Bahdanau et al., 2015).

In order to adapt the contribution of the pretrained model to the task at hand, we introduce an auxiliary LM loss during training. The joint loss is the

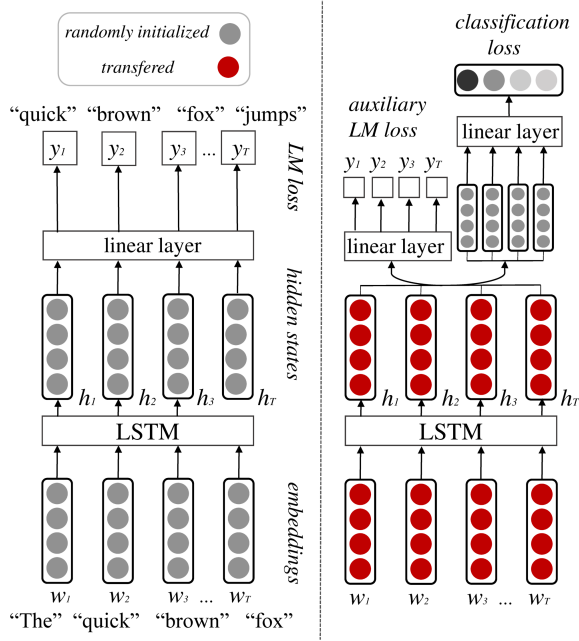


Figure 1: High-level overview of our proposed TL architecture. We transfer the pretrained LM add an extra recurrent layer and an auxiliary LM loss.

weighted sum of the task-specific loss  $L_{task}$  and the auxiliary LM loss  $L_{LM}$ , where  $\gamma$  is a weighting parameter to enable adaptation to the target task but at the same time keep the useful knowledge from the source task. Specifically:

$$L = L_{task} + \gamma L_{LM} \quad (2)$$

**Exponential decay of  $\gamma$ .** An advantage of the proposed TL method is that the contribution of the LM can be explicitly controlled in each training epoch. In the first few epochs, the LM should contribute more to the joint loss of SiATL so that the task-specific layers adapt to the new data distribution. After the knowledge of the pretrained LM is transferred to the new domain, the task-specific component of the loss function is more important and  $\gamma$  should become smaller. This is also crucial due to the fact that the new, task-specific LSTM layer is randomly initialized. Therefore, by back-propagating the gradients of this layer to the pretrained LM in the first few epochs, we would add noise to the pretrained representation. To avoid this issue, we choose to initially pay attention to the LM objective and gradually focus on the classification task. In this paper, we use an exponential decay for  $\gamma$  over the training epochs.

**Sequential Unfreezing.** Instead of fine-tuning all the layers simultaneously, we propose unfreezing

them sequentially, according to Howard and Ruder (2018); Chronopoulou et al. (2018). We first fine-tune only the extra, randomly initialized LSTM and the output layer for  $n - 1$  epochs. At the  $n^{th}$  epoch, we unfreeze the pretrained hidden layers. We let the model fine-tune, until epoch  $k - 1$ . Finally, at epoch  $k$ , we also unfreeze the embedding layer and let the network train until convergence. The values of  $n$  and  $k$  are obtained through grid search. We find the sequential unfreezing scheme important, as it minimizes the risk of overfitting to small datasets.

**Optimizers.** While pretraining the LM, we use Stochastic Gradient Descent (SGD). When we transfer the LM and fine-tune on each classification task, we use 2 different optimizers: SGD for the pretrained LM (embedding and hidden layer) with a small learning rate, in order to preserve its contextual information. As for the new, randomly initialized LSTM and classification layers, we employ Adam (Kingma and Ba, 2015), in order to allow them to train fast and adapt to the target task.

Dataset	Domain	# classes	# examples
Irony18	Tweets	4	4618
Sent17	Tweets	3	61854
SCv2	Debate Forums	2	3260
SCv1	Debate Forums	2	1995
PsychExp	Experiences	7	7480

Table 1: Datasets used for the downstream tasks.

## 4 Experiments and Results

### 4.1 Datasets

To pretrain the language model, we collect a dataset of 20 million English Twitter messages, including approximately 2M unique tokens. We use the 70K most frequent tokens as vocabulary. We evaluate our model on five datasets: *Sent17* for sentiment analysis (Rosenthal et al., 2017), *PsychExp* for emotion recognition (Wallbott and Scherer, 1986), *Irony18* for irony detection (Van Hee et al., 2018), *SCv1* and *SCv2* for sarcasm detection (Oraby et al., 2016; Lukin and Walker, 2013). More details about the datasets can be found in Table 1.

### 4.2 Experimental Setup

To preprocess the tweets, we use *Ekphrasis* (Baziotis et al., 2017). For the generic datasets, we use NLTK (Loper and Bird, 2002). For the *NBoW* baseline, we use *word2vec* (Mikolov et al., 2013) 300-dimensional embeddings as features.

	Irony18	Sent17	SCv2	SCv1	PsychExp
BoW	43.7	61.0	65.1	60.9	25.8
NBoW	45.2	63.0	61.1	51.9	20.3
P-LM	42.7 $\pm$ 0.6	61.2 $\pm$ 0.7	69.4 $\pm$ 0.4	48.5 $\pm$ 1.5	38.3 $\pm$ 0.3
P-LM + su	41.8 $\pm$ 1.2	62.1 $\pm$ 0.8	69.9 $\pm$ 1.0	48.4 $\pm$ 1.7	38.7 $\pm$ 1.0
P-LM + aux	45.5 $\pm$ 0.9	65.1 $\pm$ 0.6	72.6 $\pm$ 0.7	55.8 $\pm$ 1.0	40.9 $\pm$ 0.5
<b>SiATL</b> (P-LM + aux + su)	<b>47.0 <math>\pm</math> 1.1</b>	<b>66.5 <math>\pm</math> 0.2</b>	<b>75.0 <math>\pm</math> 0.7</b>	<b>56.8 <math>\pm</math> 2.0</b>	<b>45.8 <math>\pm</math> 1.6</b>
ULMFiT (Wiki-103)	23.6 $\pm$ 1.6	60.5 $\pm$ 0.5	<b>68.7 <math>\pm</math> 0.6</b>	<b>56.6 <math>\pm</math> 0.5</b>	21.8 $\pm$ 0.3
ULMFiT (Twitter)	<b>41.6 <math>\pm</math> 0.7</b>	<b>65.6 <math>\pm</math> 0.4</b>	67.2 $\pm$ 0.9	44.0 $\pm$ 0.7	<b>40.2 <math>\pm</math> 1.1</b>
State of the art	53.6	68.5	76.0	69.0	57.0
	(Baziotis et al., 2018)	(Cliche, 2017)	(Ilic et al., 2018)	(Felbo et al., 2017)	

Table 2: Ablation study on various downstream datasets. Average over five runs with standard deviation. *BoW* stands for Bag of Words, *NBoW* for Neural Bag of Words. *P-LM* stands for a classifier initialized with our pretrained LM, *su* for sequential unfreezing and *aux* for the auxiliary LM loss. In all cases,  $F_1$  is employed.

For the neural models, we use an LM with an embedding size of 400, 2 hidden layers, 1000 neurons per layer, embedding dropout 0.1, hidden dropout 0.3 and batch size 32. We add Gaussian noise of size 0.01 to the embedding layer. A clip norm of 5 is applied, as an extra safety measure against exploding gradients. For each text classification neural network, we add on top of the transferred LM an LSTM layer of size 100 with self-attention and a softmax classification layer. In the pretraining step, SGD with a learning rate of 0.0001 is employed. In the transferred model, SGD with the same learning rate is used for the pretrained layers. However, we use Adam (Kingma and Ba, 2015) with a learning rate of 0.0005 for the newly added LSTM and classification layers. For developing our models, we use PyTorch (Paszke et al., 2017) and Scikit-learn (Pedregosa et al., 2011).

## 5 Results & Discussion

**Baselines and Comparison.** Table 2 summarizes our results. The top two rows detail the baseline performance of the BoW and NBoW models. We observe that when enough data is available (e.g. *Sent17*), baselines provide decent results. Next, the results for the generic classifier initialized from a pretrained LM (*P-LM*) are shown with and without sequential unfreezing, followed by the results of the proposed model SiATL. SiATL is also directly compared with its close relative ULMFiT (trained on Wiki-103 or Twitter) and the state-of-the-art for each task; ULMFiT also fine-tunes a LM for classification tasks. The proposed SiATL method consistently outperforms the baselines, the *P-LM* method and ULMFiT in all datasets. Even though we do not perform any elaborate learning rate scheduling and we limit ourselves to pre-

training in Twitter, we obtain higher results in two Twitter datasets and three generic.

**Auxiliary LM objective.** The effect of the auxiliary objective is highlighted in very small datasets, such as *SCv1*, where it results in an impressive boost in performance (7%). We hypothesize that when the classifier is simply initialized with the pretrained LM, it overfits quickly, as the target vocabulary is very limited. The auxiliary LM loss, however, permits refined adjustments to the model and fine-grained adaptation to the target task.

**Exponential decay of  $\gamma$ .** For the optimal  $\gamma$  interval, we empirically find that exponentially decaying  $\gamma$  from 0.2 to 0.1 over the number of training epochs provides best results for our classification tasks. A heatmap of  $\gamma$  is depicted in Figure 3. We observe that small values of  $\gamma$  should be employed, in order to scale the LM loss in the same order of magnitude as the classification loss over the training period. Nevertheless, the use of exponential decay instead of linear decay does not provide a significant improvement, as our model is not sensitive to the way of decaying hyperparameter  $\gamma$ .

**Sequential Unfreezing.** Results show that sequential unfreezing is crucial to the proposed method, as it allows the pretrained LM to adapt to the target word distribution. The performance improvement is more pronounced when there is a mismatch between the LM and task domains, i.e., the non-Twitter domain tasks. Specifically for the *PsychExp* and *SCv2* datasets, sequentially unfreezing yields significant improvement in  $F_1$  building upon our intuition.

**Number of training examples.** Transfer learning is particularly useful when limited training data are available. We notice that for our largest dataset



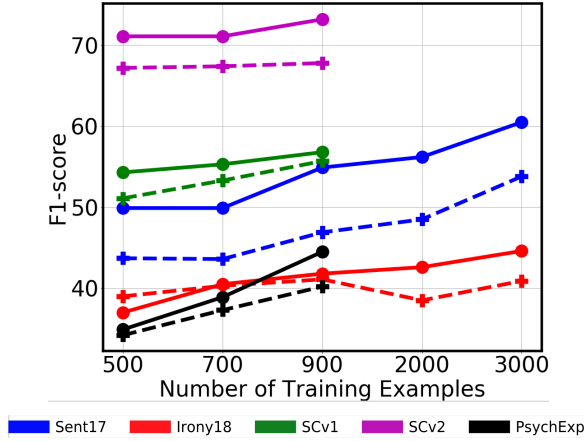


Figure 2: Results of our proposed approach (SiATL) (o) and ULMFiT (+) for different datasets as a function of the number of training examples.

*Sent17*, SiATL outperforms ULMFiT only by a small margin when trained on all the training examples available (see Table 2), while for the small *SCv2* dataset, SiATL outperforms ULMFiT by a large margin and ranks very close to the state-of-the-art model (Ilic et al., 2018). Moreover, the performance of SiATL vs ULMFiT as a function of the training dataset size is shown in Figure 2. Note that the proposed model achieves competitive results on less than 1000 training examples for the *Irony18*, *SCv2*, *SCv1* and *PsychExp* datasets, demonstrating the robustness of SiATL even when trained on a handful of training examples.

**Catastrophic forgetting.** We observe that SiATL indeed provides a way of mitigating catastrophic forgetting. Empirical results that are shown in Table 2 indicate that by only adding the auxiliary language modeling objective, we obtain better results on all downstream tasks. Specifically, a comparison of the *P-LM* + *aux* model and the *P-LM* model shows that the performance of SiATL on classification tasks is improved by the auxiliary objective. We hypothesize that the language model objective acts as a regularizer that prevents the loss of the most generalizable features.

## 6 Conclusions and Future Work

We introduce SiATL, a simple and efficient transfer learning method for text classification tasks. Our approach is based on pretraining a LM and transferring its weights to a classifier with a task-specific layer. The model is trained using a task-specific functional with an auxiliary LM loss. SiATL avoids catastrophic forgetting of the lan-

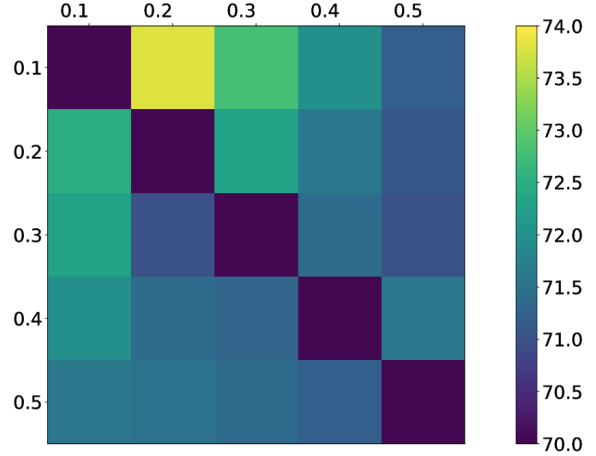


Figure 3: Heatmap of the effect of  $\gamma$  to  $F_1$ -score, evaluated on *SCv2*. The horizontal axis depicts the initial value of  $\gamma$  and the vertical axis the final value of  $\gamma$ .

guage distribution learned by the pretrained LM. Experiments on various text classification tasks yield competitive results, demonstrating the efficacy of our approach. Furthermore, our method outperforms more sophisticated transfer learning approaches, such as ULMFiT in all tasks.

In future work, we plan to move from Twitter to more generic domains and evaluate our approach to more tasks. Additionally, we aim at exploring ways for scaling our approach to larger vocabulary sizes (Kumar and Tsvetkov, 2019) and for better handling of out-of-vocabulary words (OOV) (Mielke and Eisner, 2018; Sennrich et al., 2015) in order to be applicable to diverse datasets.

Finally, we want to explore approaches for improving the adaptive layer unfreezing process and the contribution of the language model objective (value of  $\gamma$ ) to the target task.

## 7 Acknowledgements

We would like to thank Katerina Margatina and Georgios Paraskevopoulos for their helpful suggestions and comments. This work has been partially supported by computational time granted from the Greek Research & Technology Network (GR-NET) in the National HPC facility - ARIS. Also, the authors would like to thank NVIDIA for supporting this work by donating a TitanX GPU.

## References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of*

the International Conference on Learning Representations, San Diego, California.

- Christos Baziotis, Athanasia Nikolaos, Pinelopi Papalampidi, Athanasia Kolovou, Georgios Paraskevopoulos, Nikolaos Ellinas, and Alexandros Potamianos. 2018. [Ntua-slp at semeval-2018 task 3: Tracking ironic tweets using ensembles of word and character level attentive rnns](#). In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 613–621, New Orleans, Louisiana.
- Christos Baziotis, Nikos Pelekis, and Christos Doukeridis. 2017. [Datastories at semeval-2017 task 4: Deep lstm with attention for message-level and topic-based sentiment analysis](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada.
- Rich Caruana. 1993. Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48.
- Alexandra Chronopoulou, Aikaterini Margatina, Christos Baziotis, and Alexandros Potamianos. 2018. [Ntua-slp at iest 2018: Ensemble of neural transfer methods for implicit emotion classification](#). In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 57–64, Brussels, Belgium.
- Mathieu Cliche. 2017. [Bb\\_twtr at semeval-2017 task 4: Twitter sentiment analysis with cnns and lstms](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 573–580, Vancouver, Canada.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the International Conference on Machine Learning*, pages 160–167.
- Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, pages 2493–2537.
- Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3079–3087.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Bjarke Felbo, Alan Mislove, Anders Sogaard, Iyad Rahwan, and Sune Lehmann. 2017. [Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1615–1625, Copenhagen, Denmark.
- Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, (8):1735–1780.
- Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the Annual Meeting of the ACL*, pages 328–339, Melbourne, Australia.
- Suzana Ilic, Edison Marrese-Taylor, Jorge A. Balazs, and Yutaka Matsuo. 2018. [Deep contextualized word representations for detecting sarcasm and irony](#). In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 2–7.
- Diederik P. Kingma and Jimmy Ba. 2015. [Adam: A method for stochastic optimization](#). In *Proceedings of the International Conference on Learning Representations*.
- Sachin Kumar and Yulia Tsvetkov. 2019. [Von mises-fisher loss for training sequence to sequence models with continuous outputs](#). In *International Conference on Learning Representations*.
- Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.
- Edward Loper and Steven Bird. 2002. [Nltk: The natural language toolkit](#). In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics*, pages 63–70.
- Stephanie Lukin and Marilyn Walker. 2013. [Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue](#). In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 30–40, Atlanta, Georgia.
- Sebastian J. Mielke and Jason Eisner. 2018. [Spell once, summon anywhere: A two-level open-vocabulary language model](#). *CoRR*, abs/1804.08205.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3111–3119.
- Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn A. Walker. 2016. [Creating and characterizing a diverse corpus of sarcasm in dialogue](#). In *Proceedings of the SIGDIAL 2016 Conference, The 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41.

- Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. [Automatic differentiation in pytorch](#).
- Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research*, pages 2825–2830.
- Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar.
- Matthew Peters, Waleed Ammar, Chandra Bhagavathula, and Russell Power. 2017. [Semi-supervised sequence tagging with bidirectional language models](#). In *Proceedings of the Annual Meeting of the ACL*, pages 1756–1765, Vancouver, Canada.
- Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. [Deep contextualized word representations](#). In *Proceedings of the Conference of the NAACL:HLT*, pages 2227–2237, New Orleans, Louisiana.
- Prajit Ramachandran, Peter Liu, and Quoc Le. 2017. [Unsupervised pretraining for sequence to sequence learning](#). In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 383–391, Copenhagen, Denmark.
- Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. Semeval-2017 task 4: Sentiment analysis in twitter. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 502–518, Vancouver, Canada.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.
- Anders Sogaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the Annual Meeting of the ACL*, pages 231–235, Berlin, Germany.
- Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 3104–3112.
- Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. Semeval-2018 task 3: Irony detection in english tweets. In *Proceedings of The 12th International Workshop on Semantic Evaluation (SemEval-2018)*, pages 39–50, New Orleans, Louisiana.
- Harald G. Wallbott and Klaus R. Scherer. 1986. [How universal and specific is emotional experience? evidence from 27 countries on five continents](#). *Information (International Social Science Council)*, (4):763–795.
- Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic coattention networks for question answering.