Bag-of-Words Transfer:
Non-Contextual Techniques for Multi-Task Learning

Seth Ebner[1] **Felicity Wang**[1,2*] **Benjamin Van Durme**[1]
[1]Johns Hopkins University
[2]AI Foundation
{seth,vandurme}@cs.jhu.edu, felicity@aifoundation.com

Abstract

Many architectures for multi-task learning (MTL) have been proposed to take advantage of transfer among tasks, often involving complex models and training procedures. In this paper, we ask if the sentence-level representations learned in previous approaches provide significant benefit beyond that provided by simply improving word-based representations. To investigate this question, we consider three techniques that ignore sequence information: a syntactically-oblivious pooling encoder, pre-trained non-contextual word embeddings, and unigram generative regularization. Compared to a state-of-the-art MTL approach to textual inference, the simple techniques we use yield similar performance on a universe of task combinations while reducing training time and model size.[1]

1 Introduction

Multi-task learning (MTL) is usually framed as a discriminative learning problem in which predictors are learned jointly for multiple related tasks, under the premise that jointly optimizing related tasks will yield more robust parameter estimates.

In this work, we consider a collection of two-sequence classification tasks covering sentiment analysis and textual entailment. Previous work has shown that for these kinds of tasks, models incorporating only bag-of-words (BOW) features are competitive with models based on sequence encoders such as recurrent neural networks (RNNs) and convolutional neural networks (CNNs) that build compositional sequence representations (Iyyer et al., 2015; Wieting et al., 2016; Arora et al., 2017). Arora et al. (2017) suggest that BOW models better exploit the semantics of a sequence than RNNs do.

Arora et al. (2017) show that improving context-independent word-level representations may be sufficient for good performance on particular kinds of tasks. Here we ask if those findings extend to the MTL setting, and in particular how well the BOW techniques capture transfer among tasks.

We additionally observe that the standard MTL framing does not make full use of the available labeled data, as it ignores an important type of related task: generative reconstruction of the observations (§2.3). The MTL framework naturally accommodates reconstruction simply as additional tasks.

In this paper, we: (1) consider bag-of-words techniques including pooling encoders, pre-trained word embeddings, and unigram generative regularization, and (2) demonstrate that bag-of-words techniques are competitive with sequence-level techniques in MTL for sentiment analysis and textual inference (§3).

2 Bag-of-Words Techniques

We employ three approaches that use only bag-of-words representations: pooling (aggregation) encoders, pre-trained word embeddings, and unigram generative regularization. These approaches do not model sequence-level interactions. We do not use contextualized encoders such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) because they incorporate sequence-level and positional representations.

2.1 Pooling Encoders

We first consider a variant of the deep averaging network (DAN) encoder (Iyyer et al., 2015). The DAN encoder is a syntactically-oblivious encoder that consists of three steps: average (mean-pool) a sequence's non-contextual word embeddings, pass the average through feed-forward layers, and then perform linear classification on the final layer's

[*] Work done while at Johns Hopkins University.

[1] Our code is available at https://github.com/felicitywang/tfmtl.

Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo), pages 40–46
Hong Kong, China, November 3, 2019. ©2019 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

representation. We concatenate a max-pooling operation to the mean-pooling used in the first step of the original DAN encoder[2] and use a non-linear transformation in the final layer[3].

Pooling encoders such as DAN and PARAGRAM-PHRASE (which has no parameters) are much faster to train than LSTMs and CNNs, and have been shown to have competitive performance on textual similarity, textual entailment, and sentiment classification tasks (Iyyer et al., 2015; Wieting et al., 2016; Arora et al., 2017).

2.2 Pre-Trained Word Embeddings

A popular way to improve performance over the use of randomly initialized word embeddings is to use pre-trained word embeddings that have been learned from large corpora. The use of pre-trained embeddings is an example of transfer learning, which unlike MTL typically involves a pipeline of tasks rather than a joint training objective. Word embeddings are usually learned by fitting a language model (or other word prediction objective) on an out-of-domain text corpus (Mikolov et al., 2013; Pennington et al., 2014).

Although pre-trained word embeddings are learned in context and can thereby capture distributional syntactic information, good performance using pre-trained word embeddings would be evidence that sequence-aware models may not be necessary for MTL for the tasks we consider here.

Because we restrict our models to use only bag-of-words features, we seek to avoid any syntactic or sequential information that could be derived from our inputs. Any syntactic information present in pre-trained word embeddings comes from the sequences used in pre-training, not from the data in our tasks. By using pre-trained word embeddings, we seek only to determine what benefit is provided by initializing the corresponding parameters with the pre-trained embeddings rather than with random embeddings.

Additionally, contextualized encoders would capture sequential or positional information in our data inputs, so we do not use them. By not using contextualized encoders, each word has only one embedding, which is used regardless of its context.

2.3 Unigram Generative Regularization

We examine the incorporation of unigram generative regularization (UGR) for all tasks, in which we reconstruct the input sequence using a *conditional* unigram language model $p_{\boldsymbol{\theta}}(x \mid h)$.[4] Intuitively, generative regularization provides signal that addresses the question, "What do inputs with a particular label tend to look like?" For example, we wish to capture information about inputs that express positive sentiment separately from information about inputs that express negative sentiment.

We explore multi-task UGR in this work because we found that single-task UGR can improve performance (see Table 3). Additionally, multi-task UGR uses no additional data, so we get it "for free." UGR is inherently related to a dataset t's corresponding discriminative task that learns $q_{\boldsymbol{\phi}_t}(y \mid x)$, and it can be viewed as simply another task in the set of auxiliary tasks because it is realized as an auxiliary loss term.

For arbitrary networks $q_{\boldsymbol{\phi}_t}(y \mid x)$ and $p_{\boldsymbol{\theta}}(x \mid h)$, our loss function, $\mathcal{L}_{\text{GMTL}}$, on a single example is:

$$-[\alpha_t \log q_{\boldsymbol{\phi}_t}(y_i^{(t)} \mid x_i^{(t)}) + \beta_t \log p_{\boldsymbol{\theta}}(x_i^{(t)} \mid h_i^{(t)})]$$

for input $x_i^{(t)}$ and its label $y_i^{(t)}$ drawn from dataset t. The conditioning vector for the example, $h_i^{(t)}$, may include information about $y_i^{(t)}$. The discriminative and reconstruction task weights are α_t and β_t, respectively.

3 Experiments

As an external baseline, we compare our approach to methods proposed by Augenstein et al. (2018), herein referred to as **ARS**. ARS achieve state-of-the-art performance on topic-based sentiment analysis. We reimplement their baseline model as an additional comparison in our results (Table 3).

The main contributions of ARS are additional architectural components called the label embedding layer (LEL) and the label transfer network (LTN). In the baseline model, an example's two input sequences, $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, are encoded using a two-stage bi-directional RNN and then passed into a task-specific classification layer. In the LEL model, the task-specific classification layers are replaced by a label embedding matrix shared by all tasks. By embedding all the tasks' labels into a shared space, the LEL learns correlations among the tasks' labels.

[2] We tried combinations of mean-pooling, max-pooling, and min-pooling, and found mean-pooling + max-pooling performed the best based on held-out dev-set performance.

[3] We tried ELU, ReLU, sigmoid, and tanh, and chose ReLU based on held-out dev-set performance.

[4] The conditioning vector h is described in §3.2.

The LTN sits on top of the LEL and induces "pseudo-labels" for main task examples based on predicted distributions over labels made by each of the auxiliary tasks. The LTN is added to the main model after a pre-training step.

We note that ARS deliberately avoid pre-trained word embeddings in order to highlight their modeling contributions. We would expect their results to improve if pre-trained embeddings were used.

3.1 Datasets

We use the same two-sequence text classification datasets covering textual entailment and sentiment analysis used by ARS[5]: MultiNLI (Williams et al., 2018), ABSA-L/ABSA-R (Pontiki et al., 2016), Target (Dong et al., 2014), Stance (Mohammad et al., 2016), Topic-2/Topic-5 (Nakov et al., 2016), and FNC-1.[6] All of the inputs have two sequences ($\boldsymbol{x}_1$, $\boldsymbol{x}_2$), the second of which (usually a longer text, such as a Tweet or a news document) is read in the context of the first sequence (which is usually shorter, such as the topic/target/aspect of a Tweet, or a news headline). Detailed information about each dataset is shown in Table 1.

For each of our main tasks, we use the best-performing set of auxiliary tasks found by ARS (Table 2). To maintain comparability, we follow the same steps as ARS for preprocessing the data. In particular, MultiNLI was downsampled to the same 10K training examples (2.5%) as ARS, and so we refer to it as MultiNLI$^{2.5\%}$.[7]

3.2 Training Procedure

In all experiments, we seek to optimize performance on the main task, rather than optimize an aggregate metric across main and auxiliary tasks.

We set the discriminative task weights $\alpha_t = \alpha = 1$ for all discriminative tasks, and we fix the reconstruction task weights $\beta_t = \beta$ across all reconstruction tasks for a given set of main and auxiliary tasks. We found performance improves when $\beta \ll \alpha$, which is consistent with the treatment of reconstruction as a regularizing task.[8] In general, α_t and β_t may be tuned separately for each task.

We use 100-dimensional GloVe 6B[9] word embeddings and initialize the embeddings of words that appear in the GloVe vocabulary with their pre-trained embeddings (Pennington et al., 2014). Other words' embeddings are initialized randomly. All embeddings are fine-tuned during training.

Because we want to see if good performance can be attained without sequence-level information, we reconstruct $\boldsymbol{x}_2$ using a unigram decoder, which projects the conditioning information h into a distribution over the vocabulary.

The conditioning vector decomposes as $h := [\boldsymbol{t}, \boldsymbol{y}', \boldsymbol{\pi}_1]$, which consists of: (1) a one-hot encoding $\boldsymbol{t}$ of the task index t; this allows the language model to adapt to different tasks (Daumé III, 2007); (2) a task-specific projection $\boldsymbol{y}' = \mathbf{L}_t \boldsymbol{y}$ of the one-hot label vector $\boldsymbol{y}$, where $\mathbf{L}_t \in \mathbb{R}^{l \times |\mathcal{Y}_t|}$ are trainable task-specific parameters; this projection transforms labels from potentially disparate label spaces $\mathcal{Y}_t$ of different sizes to the same space; and (3) the input encoding $\boldsymbol{\pi}_1$, which conveys information about $\boldsymbol{x}_1$, on which we condition the reading of $\boldsymbol{x}_2$.[10]

Together, the elements of the conditioning vector h provide for controllable text generation, in which the task, label, and context $\boldsymbol{x}_1$ together influence the distribution over words of $\boldsymbol{x}_2$ parametrized by $p_{\boldsymbol{\theta}}$ (Hu et al., 2017).[11]

4 Discussion

Our experimental results are presented in Table 3. For the sake of comparison, we keep with the set of auxiliary tasks used by ARS, which are listed in Table 2. Other combinations of tasks may give better performance for the techniques we examine.

Using just bag-of-words features, our best models outperform the reimplementation of ARS's baseline bi-directional RNN model in 4 of 7 cases and achieve competitive results in the other 3 cases. Our results are also competitive with ARS's best-performing models, which may use the label embedding layer and label transfer network.

The DAN encoder in the single-task learning (STL) setting is competitive with ARS's STL results and with our STL and MTL reimplementa-

[5] We do not include results for FNC-1 as a main task because the FNC-1 development set of ARS consists of examples of only a single label type, making model selection (the intent of a dev-set) problematic.

[6] http://www.fakenewschallenge.org/

[7] p.c. with Isabelle Augenstein.

[8] In preliminary experiments, the hyperparameter β was swept from 10^{-5} to 10^{5} in powers of 10. Because of poor performance for large β, for subsequent experiments we reduced the range to 10^{-5} to 10^{1} in powers of 10.

[9] http://nlp.stanford.edu/data/glove.6B.zip

[10] Single-sequence tasks would not condition on $\boldsymbol{\pi}_1$.

[11] Here, the decoder $p_{\boldsymbol{\theta}}$ is coupled with the encoder $q_{\boldsymbol{\phi}_t}$ both in the representation $\boldsymbol{\pi}_1$ and in the word embeddings. In principle, $p_{\boldsymbol{\theta}}$ may be decoupled from $q_{\boldsymbol{\phi}_t}$ entirely except for the word embeddings.

Dataset	# Labels	# Train	Seq 1	Seq 2	Task
MultiNLI$^{2.5\%}$	3	10,001	Hypothesis	Premise	Natural language inference
ABSA-L	3	2,618	Aspect	Review	Aspect-based sentiment analysis, laptop domain
ABSA-R	3	2,256	Aspect	Review	Aspect-based sentiment analysis, restaurant domain
Target	3	5,623	Target	Text	Target-dependent sentiment analysis
Stance	3	3,209	Target	Tweet	Stance detection
Topic-2	2	5,177	Topic	Tweet	Topic-based sentiment analysis, binary
Topic-5	5	7,236	Topic	Tweet	Topic-based sentiment analysis, fine-grained
FNC-1	4	39,741	Headline	Document	Fake News Detection

Table 1: Size of label set, number of training examples, content of sequences, and task description of each dataset.

Main task	Auxiliary tasks
MultiNLI$^{2.5\%}$	Topic-5
ABSA-L	Topic-5
ABSA-R	Topic-5, ABSA-L, Target
Target	FNC-1, MultiNLI$^{2.5\%}$, Topic-5
Stance	FNC-1, MultiNLI$^{2.5\%}$, Target
Topic-2	FNC-1, MultiNLI$^{2.5\%}$, Target
Topic-5	FNC-1, MultiNLI$^{2.5\%}$, ABSA-L, Target

Table 2: Main tasks and their corresponding auxiliary tasks as used here and by Augenstein et al. (2018).

tions, confirming the findings of previous work discussed in §2.1.

The inclusion of unigram generative regularization (UGR) improves STL DAN performance in 5 of 7 cases (GSTL), motivating its use in the MTL setting. If GSTL performance achieves desired performance, then one saves a search over auxiliary tasks, such as those in (Liu et al., 2016; Augenstein et al., 2018). However, UGR hurts MTL performance in 6 of 7 cases (GMTL). Furthermore, GMTL performance is worse than GSTL performance in all cases, while MTL outperforms GSTL in 5 of 7 cases. These trends suggest that UGR is not needed once the regularization from incorporating auxiliary discriminative tasks takes effect. In other words, the parameter updates resulting from UGR are not as informative as the parameter updates resulting from having additional training examples from similar datasets. However, UGR may still be helpful when auxiliary training sets are not available.

Comparing STL to MTL results, we see that the DAN encoder often facilitates transfer across tasks. The best-performing MTL DAN model outperforms or equals the best-performing STL DAN model in 6 of 7 cases (all but Stance). The use of GloVe embeddings in MTL and GMTL improves performance over the use of randomly initialized embeddings because the task-independent information captured by the pre-trained word embeddings serves as good initialization.

Comparisons in training time, model size, and performance between the reimplemented ARS baseline model and the DAN model are given in Table 4 for MultiNLI$^{2.5\%}$ and Topic-5, the largest dataset and the dataset with the most auxiliary tasks, respectively. The DAN model is 33.4% smaller and 7.7x faster than the ARS model for MultiNLI$^{2.5\%}$ but achieves lower accuracy. DAN (run on a CPU) is 1.2x faster and 14.4% smaller than the ARS model (run on a GPU) for Topic-5 and achieves better performance.[12] As expected based on prior work, the training speed of the DAN encoder is substantially faster than that of the bi-RNN encoder, especially for MultiNLI$^{2.5\%}$.

Although the competitive results of the bag-of-words models are somewhat expected given prior work, we find the magnitude of the gains over the MTL bi-RNN reimplementation surprising, especially on Stance and Topic-2. Overall, our results extend the findings of prior work on simple sentence encoders for sentiment analysis and textual inference to the MTL setting.

5 Related Work

Prior work has shown that bag-of-words pooling encoders compete with sequence encoders on sentiment analysis, textual entailment, and textual similarity for single-task learning (Iyyer et al., 2015; Wieting et al., 2016; Arora et al., 2017). In this work, we explore these tasks in the MTL setting and ask if transfer among the tasks can be captured by bag-of-words features.

Recent work in MTL has explored different parameter sharing schemes in shared neural architectures. Some models incorporate inductive bias by imposing hierarchies over tasks (Søgaard and

[12] We would expect the time contrast for Topic-5 to be more pronounced if the DAN and ARS models were run on the same hardware.

	MultiNLI$^{2.5\%}_{\uparrow}$	ABSA-L$_{\uparrow}$	ABSA-R$_{\uparrow}$	Target$_{\uparrow}$	Stance$_{\uparrow}$	Topic-2$_{\uparrow}$	Topic-5$_{\downarrow}$
Metric	Acc	Acc	Acc	F_1^M	F_1^{FA}	ρ^{PN}	MAE^M
ARS STL (baseline)	49.25	76.74	67.47	64.01	41.1	63.92	0.919
ARS MTL (baseline)	49.39	74.94	82.25	65.73	44.12	80.74	0.859
ARS MTL (best)	49.94*	75.66*†	83.71*†	66.42*	46.26*	80.74	0.803*†
ARS STL (r)	47.71	73.16	72.99	62.44	25.05	63.91	0.903
ARS MTL (r)	49.20	75.03	79.39	63.61	29.30	61.26	0.914
STL DAN (w)	38.82	**74.03**	80.79	63.35	34.31	64.15	0.907
GSTL DAN (w)	41.70	73.53	78.58	63.45	**35.17**	65.09	0.906
MTL DAN (w)	**47.69**	**74.03**	79.86	61.44	31.77	65.42	0.900
MTL DAN + GloVe (w)	43.04	68.91	**81.84**	**63.53**	30.96	**67.85**	**0.856**
GMTL DAN (w)	39.35	69.29	78.23	61.95	25.70	59.88	0.927
GMTL DAN + GloVe (w)	40.41	69.29	80.21	63.01	26.36	61.17	0.958

Table 3: Test results. Acc: accuracy; F_1^M: macro-averaged F_1; F_1^{FA}: macro-averaged F_1 of "favour" and "against" classes; ρ^{PN}: macro-averaged recall, averaged across topics; MAE^M: macro-averaged mean absolute error, averaged across topics. ↑/↓ next to each task name indicates that higher/lower score is better. "STL": single-task setting; "MTL": multi-task setting; "(r)": reimplementation of baseline bi-directional RNN model from ARS (no Label Embedding Layer or Label Transfer Network). *: model uses LEL; †: model uses LTN. Models using only BOW representations are marked with (w). Best results from BOW experiments (bottom section) are **bolded**.

Dataset	Model	Epoch	# Params.	Metric
MNLI$^{2.5\%}$	ARS (r)	268 s	362,608	**49.20**
	DAN	**35 s**	**241,408**	47.69
Topic-5	ARS (r)	93 s (G)	423,918	0.914
	DAN	**75 s**	**362,718**	**0.900**

Table 4: Comparisons of mean training epoch time, number of trainable architecture parameters (i.e., trainable non-word-embedding parameters), and performance of the reimplemented (r) ARS model and the DAN model in the MTL setting for the MultiNLI$^{2.5\%}$ and Topic-5 datasets. (G) denotes the model was run on a GPU, otherwise the model was run on a CPU.

Goldberg, 2016; Hashimoto et al., 2017; Sanh et al., 2019). Ruder et al. (2017) and Liu and Huang (2018) incorporate orthogonality constraints to learn which parameters tasks should share. Previous work in MTL has also lead to non-trivial training procedures. For example, Liu et al. (2017) and Chen and Cardie (2018) use adversarial training, and Ruder and Plank (2018) explore tri-training. The focus of this paper is a collection of BOW tools that form strong baselines upon which architectural or training improvements can be shown.

Ando and Zhang (2005) motivate the inclusion of auxiliary tasks for MTL. They automatically annotate unlabeled data to create a new labeled dataset that is related to the main task. In this work, our auxiliary tasks are pre-existing labeled datasets for which we include discriminative and reconstruction objectives. Criteria and heuristics for the selection of auxiliary tasks are discussed by Alonso and Plank (2017) and Bingel and Søgaard (2017).

For a given task, it is well-established that the addition of auxiliary word prediction objective terms may help regularize the representations used for prediction (Dai and Le, 2015; Kiros et al., 2015; Rei, 2017). Rei (2017) proposes a semi-supervised MTL framework for sequence tagging that incorporates a secondary language modeling objective. Like that approach, our unigram generative regularization (§2.3) requires no additional data. Our approach differs from Rei (2017) in three ways: we employ a *conditional* language model instead of an unconditional language model, allowing our model to learn in a supervised way from signal derived from the labels; we do not use semi-supervised learning; and we train in a multi-task setting involving both multiple datasets and a compound objective, whereas Rei (2017) optimizes a compound objective on a single dataset for each task (similar to GSTL in Table 3 of this work). To the best of our knowledge, our use of (unigram) generative regularization in the multi-task setting is novel.

6 Conclusion

We showed that bag-of-words techniques such as pooling encoders and non-contextual pre-trained word embeddings can capture transfer among sentiment analysis and textual entailment tasks in multi-task learning. We additionally showed that unigram generative regularization often improved single-task learning performance but not multi-task learning performance, suggesting that generative regularization is not needed once the regularization from incorporating auxiliary discriminative tasks takes effect. The bag-of-words techniques are competitive with a state-of-the-art model, thereby extending the findings of prior work on bag-of-words approaches to sentiment analysis and textual entailment to the multi-task setting.

Acknowledgments

We thank Nicholas Andrews for many helpful discussions throughout the development of this work. We also thank the anonymous reviewers for their feedback. This work was supported in part by IARPA MATERIAL and DARPA AIDA. The views and conclusions contained in this publication are those of the authors and should not be interpreted as representing official policies or endorsements of IARPA, DARPA, or the U.S. Government.

References

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *15th Conference of the European Chapter of the Association for Computational Linguistics*.

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. *International Conference on Learning Representations*.

Isabelle Augenstein, Sebastian Ruder, and Anders Søgaard. 2018. Multi-task learning of pairwise sequence classification tasks over disparate label spaces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1896–1906.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 164–169.

Xilun Chen and Claire Cardie. 2018. Multinomial adversarial networks for multi-domain text classification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1226–1240.

Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pages 3079–3087.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. *ACL 2007*, page 256.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 49–54.

Kazuma Hashimoto, Yoshimasa Tsuruoka, Richard Socher, et al. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1923–1933.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596.

Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. 2015. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 1681–1691.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Pengfei Liu and Xuanjing Huang. 2018. Meta-learning multi-task communication. *arXiv preprint arXiv:1810.09988*.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 2873–2879.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1–10. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Saif Mohammad, Svetlana Kiritchenko, Parinaz Sobhani, Xiaodan Zhu, and Colin Cherry. 2016. Semeval-2016 task 6: Detecting stance in tweets. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 31–41.

Preslav Nakov, Alan Ritter, Sara Rosenthal, Fabrizio Sebastiani, and Veselin Stoyanov. 2016. Semeval-2016 task 4: Sentiment analysis in twitter. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1–18.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016)*, pages 19–30.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2121–2130.

Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Anders Søgaard. 2017. Latent multi-task architecture learning. *stat*, 1050:23.

Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1044–1054.

Victor Sanh, Thomas Wolf, and Sebastian Ruder. 2019. A hierarchical multi-task approach for learning embeddings from semantic tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6949–6956.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Towards universal paraphrastic sentence embeddings. In *Proceedings of International Conference on Learning Representations*.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.