Multi-task Learning for Universal Sentence Representations: What Syntactic and Semantic Information is Captured?

Wasi Uddin Ahmad[†], **Xueying Bai**[*], **Zhechao Huang**[§], **Chao Jiang**[*], **Nanyun Peng**[⋆], **Kai-Wei Chang**[†]
wasiahmad@ucla.edu, xb6cf@virginia.edu, huangzhechao1995@gmail.com
cj7an@virginia.edu, npeng@isi.edu, kwchang.cs@ucla.edu
[§]Fudan University, [*]University of Virginia
[⋆]University of Southern California, [†]University of California, Los Angeles

arXiv:1804.07911v1 [cs.CL] 21 Apr 2018

Abstract

Learning distributed sentence representations is one of the key challenges in natural language processing. Previous work demonstrated that a recurrent neural network (RNNs) based sentence encoder trained on a large collection of annotated natural language inference data, is efficient in the transfer learning to facilitate other related tasks. In this paper, we show that joint learning of multiple tasks results in better generalizable sentence representations by conducting extensive experiments and analysis comparing the multi-task and single-task learned sentence encoders. The quantitative analysis of the syntactic and semantic information captured by the sentence embeddings show that multi-task learning captures better syntactic information while the single task learning summarizes the semantic information coherently.

1 Introduction

Learning distributional representations of words, phrases and sentences have long been interested in the natural language processing (NLP) community. In recent years, many successful stories showed that the learned representations are transferable to other tasks. For example, many NLP (Zou et al., 2013; Seo et al., 2017; Lee et al., 2017; Chen et al., 2017) and computer vision applications (Venugopalan et al., 2017; Teney et al., 2017) have successfully utilized the word embeddings (Bengio et al., 2003; Mikolov et al., 2013; Pennington et al., 2014). They are particularly beneficial when there is an insufficient amount of training examples to learn the word embeddings from scratch. While the techniques of training word embedding are relatively matured and pre-trained word embeddings have been publicly available, learning high quality comprehensive sentence representations is still a challenging problem.

Conneau et al. (2017) showed that a long short-term memory network (LSTM) (Hochreiter and Schmidhuber, 1997) based sentence encoder trained on annotated corpus for natural language inference (NLI) task can capture useful features and is transferable to other tasks. By leveraging the annotations of the NLI task, where the semantic relationships between sentences have to be captured, the encoder significantly outperformed existing unsupervised approaches (Kiros et al., 2015; Hill et al., 2016). However, the NLI dataset is domain specific and relatively small in terms of vocabulary size[1], limiting the performance of the universal sentence encoder. This leads to our research question: can we learn a better sentence encoder by leveraging more information from multiple different types of *supervised* tasks?

In this paper, we study multi-task learning for training universal sentence representations. We consider three large-scale text classification corpora, (1) Stanford Natural Language Inference (SNLI) (Bowman et al., 2015), (2) Multi-Genre NLI (Multi-NLI) (Williams et al., 2017), and (3) Quora duplicate question pairs (Quora)[2], which covers a variety of domains and two different tasks – textual entailment and question paraphrasing. We investigate two multi-task learning (MTL) frameworks, fully shared models and shared-private models to train a sentence encoder jointly on three tasks and examine their efficiency on 10 transfer tasks. Our experiments show that sentence embeddings learned through multi-task learning outperform the state-of-the-art universal

[1]The largest NLI dataset, SNLI, only covers 42.7k unique tokens and extracted from image captions.

[2]https://www.kaggle.com/quora/question-pairs-dataset

sentence encoder (Conneau et al., 2017).

We conduct additional quantitative analysis to identify what types of syntactic and semantic information are better captured by the multi-task learned sentence embeddings. For syntactic information, we design six binary classification tasks regarding whether certain POS tags or a non-terminal node appear in the sentence. We also investigate the sentence reordering task as well as designing a regression task for sentence length prediction. The contributions of the paper are summarized as follows.

1. We propose to employ multi-task learning for learning better generalizable sentence representations.
2. Our multi-task learned sentence representations significantly outperform those learned from a single task on 7/10 transfer tasks and achieves comparable results on the rest.
3. We design experiments to quantitatively analyze what syntactic and semantic information is captured by the sentence representations.
4. Our sentence encoders will be publicly available to facilitate research in related areas.

2 Related Work

Our work is closely related to sentence representation learning, transfer learning, and multi-task learning, because we propose to utilize multi-task learning to learn sentence representations that are helpful for transfer learning. We briefly review each of the areas in this section.

• **Sentence Representations Learning.** Training complex networks to generate useful sentence representations has become a core component in many NLP applications. Recent works on learning distributional sentence representations such that they capture the syntactic and semantic regularities within sentences range from models that compose of word embeddings (Le and Mikolov, 2014; Arora et al., 2017; Wieting et al., 2016) to more complex neural network architectures (Zhao et al., 2015; Wang and Jiang, 2016; Liu et al., 2016c; Lin et al., 2017). General purpose distributional sentence representations can be learned from a large collection of unlabeled text corpora.

Kiros et al. (2015) proposed an *unsupervised* approach called SkipThought, by revising the skip-gram model (Mikolov et al., 2013). Skip-thought vectors help to achieve good results on 8 transfer tasks which are further improved by using layer normalization (Ba et al., 2016). Hill et al. (2016) proposed two *unsupervised* objectives but they resulted in lower performance compared to SkipThought. Unlike word embeddings, learning sentence representations in an unsupervised fashion lack the reasoning about semantic relationships between sentences. To learn universal sentence representations from *supervised* natural language inference data, Conneau et al. (2017) propose a BiLSTM network with max pooling that yields the state-of-the-art in sentence encoding methods, outperforming unsupervised approaches like skip-thought vectors. We also investigate *supervised* learning of sentence embeddings, but focus on exploring the benefits of multi-task learning for yielding better sentence encoders.

• **Transfer Learning.** Transfer learning stores the knowledge gained from solving source tasks (usually with abundant annotated data), and apply it to other tasks (usually suffer from insufficient annotated data to train complex models) to combat the inadequate supervision problem. It has become prevalent in many computer vision applications (Sharif Razavian et al., 2014; Antol et al., 2015) where image features were trained on ImageNet (Deng et al., 2009), and NLP applications where word vectors (Pennington et al., 2014; Mikolov et al., 2013) were trained on large unlabeled corpora. Despite the benefits of using pre-trained word embeddings, many NLP applications still suffer from lacking high quality generic sentence representations that are transferable to help other tasks. In this work, we investigate whether multi-task learning can help to obtain more robust sentence representations that transfer better.

• **Multi-task Learning.** Multi-task learning has been successfully used in a wide-range of natural language processing (NLP) applications, including text classification (Liu et al., 2017), machine translation (Luong et al., 2016), sequence labeling (Rei, 2017), sequence tagging (Peng and Dredze, 2017), dependency parsing (Peng et al., 2017) etc. Recently, Mou et al. (2016) evaluated the utility of joint learning on sentence classification tasks to examine how much transferable neural networks are for NLP applications. However, they did not learn universal sentence encoders that can transfer information to unseen tasks as we do in this paper. Liu et al. (2016b); Zhang et al. (2017b) proposed multi-task learning architecture with dif-

ferent methods of sharing information among the participating tasks. To further enhance, such models are augmented with external memory which is shared across the participant tasks (Liu et al., 2016a). To facilitate scaling and transferring when a large number of tasks are involved, Zhang et al. (2017a) propose to embed labels by considering semantic correlations among tasks.

In this work, we study how to adapt the *existing* multi-task learning architectures to learn generic sentence representations that are transferable.

3 Multi-task Learning for Universal Sentence Encoder

While multi-task learning was shown efficient in many NLP tasks, its effectiveness in learning robust sentence representations that can generalize to unseen tasks is largely untested. In this paper, we explore several existing multi-task learning frameworks for our goal of learning sentence representations. We hypothesize that with different training signals from various data and tasks, the learned sentence representations will be more robust and transferable. Our sentence encoders are trained on several sentence similarity or entailment datasets, and are employed to produce generic sentence representations for transfer tasks. We study two hard parameter sharing frameworks[3] and explore two settings: the fully shared (FS) setting and the shared-private (SP) setting (Liu et al., 2017) (see figure 1 in the appendix) to learn generalizable sentence representations.

We start by introducing the sentence encoder, which is our basic building block, then explaining the multi-task learning frameworks we employ.

3.1 Sentence Encoder

A wide variety of neural network architectures can be used to convert sentences into fixed-size representations. We choose one layer bidirectional LSTM (Hochreiter and Schmidhuber, 1997) with max pooling (BiLSTM-Max) sentence encoder as our basic building block for all multi-task learning architectures because it was found very effective in sentence encoding (Conneau et al., 2017). Formally, for a sentence with T words $w = [w_1, w_2, ..., w_T]$, we have:

$$\overrightarrow{h}_t = LSTM(\overrightarrow{h}_{t-1}, w_t), \quad (1)$$

$$\overleftarrow{h}_t = LSTM(\overleftarrow{h}_{t+1}, w_t), \quad (2)$$

$$\text{and} \quad h_t = [\overrightarrow{h}_t, \overleftarrow{h}_t], \quad (3)$$

where $h_t \in R^{2d}$ is the t-th hidden vectors in BiLSTM, d is the dimensionality of the LSTM hidden units. To form a fixed-size vector representation of variable length sentences, the maximum value is selected over each dimension of the hidden units:

$$s_j = \max_{t \in [1,...,T]} \ h_{j,t}, \ j = 1, ..., d, \quad (4)$$

where s_j is the j-th element of the sentence embedding s. Since we are coping with sentence entailment and similarity problems, there are always two sentences in one instance, denoted as (s_{i1}, s_{i2}), where i indexes the instances. To get the representation of an instance, we follow Liu et al. (2016c) and define:

$$s_i = [s_{i1}, s_{i2}, s_{i1} - s_{i2}, s_{i1} \odot s_{i2}],$$

where $\odot$ denotes element-wise multiplication.

3.2 Fully Shared (FS) models

In fully-shared multi-task learning models, a single sentence encoder is shared across all tasks to learn generalizable representations. The sentence representations obtained by following Eq. (4), are passed through task-specific two-layer feed-forward neural networks to predict the task-specific target labels.

$$\hat{y}_i^k = \textit{softmax}(W_2\sigma(W_1 s_i^k + b_1) + b_2)$$

where i and k indexes instances and tasks. σ denotes element-wise sigmoid function. We use cross-entropy loss. Thus the loss of each task is:

$$L_{task}^k = -\sum_{i=1}^{N_k} y_i^k \log(\hat{y}_i^k),$$

where y_i^k is the gold label of the i-th instance in task k. The final multi-task loss L_{task} is defined as a simple summation over the loss of each task.

3.3 Shared-Private (SP) models

Shared-private models contain shared encoders and private ones, which encourage the task-specific features being learned by the private encoders and generic features by the shared layers.

[3] Two most common ways to perform multi-task learning are hard and soft parameter sharing. The former shares some parameters across all tasks while maintaining some task-specific parameters; the latter contains only task-specific parameters, but regularize the parameters of different tasks to be similar to encourage generalizable parameters. We confine our study to use hard parameter sharing models as they are widely used in the NLP community.

In addition, each task has its own task-specific models to make the final prediction. We hypothesize that after proper multi-task training, the generic encoders are more useful than the private ones for transfer learning. In this study, we design one private BiLSTM-Max sentence encoder for each task, and one shared BiLSTM-Max encoder for all the tasks to capture generic features. Sentence embeddings produced by private and shared encoders are concatenated to form the final sentence representations. Formally, for any sentence in a given task k, its shared representation s_s^k and private representation s_p^k can be computed using Eq. (1) – (4), and the private and shared representations are concatenated to construct the sentence representations: $s^k = s_s^k \oplus s_p^k$.

Adversarial Training. Ideally, we want the private encoders to learn only task-specific features, and the shared encoder to learn general and common features. To achieve this goal, we adopt the adversarial training strategy proposed by Liu et al. (2017) to introduce a discriminator on top of the shared BiLSTM-Max sentence encoder. The goal of the discriminator, D is to identify which task an encoded sentence s^k comes from, and the adversarial training requires the shared sentence encoder to generate representations that can "fool" the discriminator. In this way, the shared encoder is forced not to carry task-related information. Detailed information about the adversarial training procedure is provided in the appendix.

3.4 Hyper-parameter Tuning

We trained the aforementioned multi-task learning frameworks on SNLI, Multi-NLI, and Quora datasets. We carefully tune the parameters on the development set and report the testing performance with best parameters. We use SGD with an initial learning rate of 0.1 and a weight decay of 0.99. At each epoch, we divide the learning rate by 5 if the development accuracy decreases. We use mini-batches of size 128 and training is stopped when the learning rate goes below the threshold of 10^{-5}. For the task-specific classifier, we use a multi-layer perceptron with 1 hidden-layer of 512 hidden units. For the hidden size of BiLSTM, we consider the range $[256, 512, 1024, 2048]$ and found 2048 results in best performance. We use 300 dimensional GloVe word vectors (Pennington et al., 2014) trained on 840 billion of tokens as fixed word embeddings.

4 Experimental Setup

We hypothesize that sentence representation learned by multi-task learning can capture generic information and perform better on transfer tasks than single task trained ones. To test this hypothesis, we train the multi-task learning models on three large-scale textual entailment and paraphrasing datasets to learn generic sentence representations and test the generalizability of the sentence encoder on 10 additional transfer tasks. A detailed description of the source and transfer tasks are presented in table 1 in the appendix. In addition, we perform a quantitative analysis using 8 auxiliary tasks to show what syntactic and semantic information is captured by the sentence representations.

Source tasks. The first source task is natural language inference (NLI) which determines whether a natural language hypothesis can be inferred from a natural language premise. We consider the SNLI (Bowman et al., 2015) and the Multi-Genre NLI (Multi-NLI) (Williams et al., 2017) corpus which consist of English sentence pairs, manually labeled with one of the three categories: entailment, contradiction and neutral. The SNLI dataset is constructed from image captions while the Multi-NLI dataset includes a more diverse range of text, as well as a test set for cross-genre transfer evaluation. Following Conneau et al. (2017), we also conduct experiments that combine SNLI and Multi-NLI datasets, which is denoted as AllNLI.

The second source task is the Quora duplicate question detection (DupQuD) task. Quora released a dataset of 404k question pairs to automate the process of identifying semantically equivalent question pairs. We split the Quora dataset as that in Wang et al. (2017).

Transfer tasks. We use the tool[4] developed and released by Conneau et al. (2017) to evaluate the sentence encoders on 10 additional tasks using transfer learning. Among them, six are text classification tasks for sentiment analysis (MR, SST), question-type (TREC), product reviews (CR), subjectivity/objectivity (SUBJ) and opinion polarity (MPQA). For these tasks, we report accuracy. The other four tasks are related to textual entailment and paraphrasing which are SICK-E, SICK-R, MRPC, and STS-14. SICK-E task is very similar to SNLI, while MRPC is a paraphrase identification task which is very similar to the DupQuD

[4]https://github.com/facebookresearch/SentEval/

Model Type	DupQuD		SNLI		Multi-NLI	
	dev	test	dev	test	dev	test
Conneau et al. (2017)	-	-	85.0	84.5	-	-
In domain single task sentence representation learning						
BiLSTM-Max	87.1	86.7	84.7	84.4	70.2/70.8	70.8/69.8
Two-datasets and two tasks (DupQuD and SNLI)						
Fully Shared	86.8	86.6	84.9	84.8	-	-
Shared Private	87.0	86.8	84.8	84.7	-	-
Adversarial Shared Private	87.5	**87.0**	**85.2**	84.9	-	-
Three-datasets and two tasks (DupQuD and AllNLI)						
Fully Shared	87.3	86.8	**85.2**	84.8	69.9/70.3	70.0/68.9
Shared Private	86.9	86.0	85.2	84.7	70.7/70.5	70.8/69.3
Adversarial Shared Private	87.0	86.6	84.7	84.3	70.3/69.4	69.6/68.3
Three-datasets and three tasks (DupQuD, SNLI and Multi-NLI)						
Fully Shared	86.6	85.9	84.3	84.3	70.3/69.7	70.1/69.6
Shared Private	**87.6**	**87.0**	**85.2**	**85.2**	**71.2/71.0**	**71.0/70.1**
Adversarial Shared Private	86.6	86.3	84.6	84.7	70.7/70.7	71.0/70.1

Table 1: Development and test accuracy of source tasks (DupQuD, SNLI and Multi-NLI) obtained through various multi-task learning architectures. Underlined values indicate best performance among models trained on the same set of tasks and bold-faced values indicate best performance across the board.

task. The tool provides accuracy and f1 score for MRPC but only accuracy for SICK-E. Unlike SICK-E, SICK-R and STS-14 tasks predict a score between 0 and 5 given sentence pairs which indicate the level of textual similarity. For the SICK-R task, Pearson correlation is used as a measure and for the STS-14 task, the tool reports both Pearson and Spearman correlation.

Auxiliary tasks. We design seven syntactic and one semantic auxiliary tasks to study what information is embedded in the sentence vectors. The seven syntactic analysis tasks are: sentence length prediction (regression task), whether a sentence contains adjectives, adverbs, proper nouns, possessive pronouns, or cardinal numbers (binary classification tasks), and whether a sentence's constituency parse contains prepositional phrase (binary classification task). We conduct all the syntactic auxiliary experiments based on the Penn Treebank (PTB) (Marcus et al., 1993) dataset. We use a two-layer feed-forward neural network on top of the pretrained sentence encoders for the syntactic auxiliary tasks. We report mean squared error for the regression task and F1 score for the binary classification tasks.

The semantic analysis task is sentence ordering (Li and Jurafsky, 2017), which arranges a set of sentences into a coherent text. This requires a deep understanding of the semantics of sentences. We implement the end-to-end pointer network on top of the pretrained sentence encoders proposed in (Gong et al., 2016) to evaluate the learned sentence representations. We conduct the semantic analysis experiment based on the ROCStories (Mostafazadeh et al., 2016) and the Visual Storytelling (VIST) (Huang et al., 2016) datasets. Both the datasets contain stories consist of five sentences. We shuffled the sentences and use the pointer network to arrange them in the correct order. The pointer network relies on the pre-trained encoders to generate sentence embeddings. We report pairwise metrics (PM), longest sequence ratio (LSR) and perfect match ratio (PMR) for the sentence ordering evaluation. We refer the reader to Gong et al. (2016) for details of the evaluation metrics. Detailed information of the auxiliary task datasets is provided in table 3 in the appendix.

5 Experimental Results

We split our experiments into three parts. The first part examines the efficiency of multi-task learning for the large-scale textual entailment and paraphrasing tasks; the second part evaluates the quality of the learned sentence encoders by using them to generate features for 10 different transfer tasks; and the third part quantitatively analyzes what syntactic and semantic information is captured by the trained sentence encoders.

Model Type	MR	CR	SUBJ	MPQA	SST	TREC	SICK-R	SICK-E	MRPC	STS14
1. Results reported in Conneau et al. (2017)										
1.1. BiLSTM-Max (SNLI)	79.9	84.6	92.1	89.8	83.3	88.7	0.885	86.3	75.1/82.3	.68/.65
1.2. BiLSTM-Max (AllNLI)	81.1	86.3	92.4	90.2	84.6	88.2	0.884	86.3	76.2/83.1	.70/.67
2. Sentence representation learning from single-task										
2.1. BiLSTM-Max (SNLI)	80.1	85.3	92.6	89.1	83.6	89.2	0.885	86.0	75.2/82.4	.66/.64
2.2. BiLSTM-Max (DupQuD)	79.2	84.6	92.6	88.8	83.5	88.0	0.861	82.4	74.8/82.8	.62/.60
2.3. BiLSTM-Max (Multi-NLI)	81.2	85.8	93.1	**89.5**	83.4	88.8	0.863	84.7	75.9/83.1	.66/.63
2.4. BiLSTM-Max (AllNLI)	80.9	86.3	93.2	89.2	83.3	88.8	**0.887**	86.7	76.4/83.4	**.69/.66**
3. Sentence representation learning from three-tasks (DupQuD, SNLI and Multi-NLI)										
3.1. Shared Private	81.6	**86.9**	**93.9**	89.2	**84.4**	90.4	0.883	85.9	76.5/83.3	.66/.63
3.2. Adversarial Shared Private	**82.0**	86.3	93.8	89.4	84.1	**92.2**	0.884	**87.0**	**77.2/83.6**	.68/.65
4. Comparisons between different settings										
4.1. Compare 3.2 to 2.1	+1.9	+1.0	+1.2	+0.3	+0.5	+3.0	−.001	+1.0	+2.0/+1.2	+.02/+.01
4.2. Compare 3.2 to 2.4	+1.1	0.0	+0.6	+0.2	+0.8	+3.4	−.003	+0.3	+0.8/+0.2	−.01/−.01

Table 2: Transfer learning results for various single-task and multi-task learning settings trained on different combinations of DupQuD, SNLI and Multi-NLI datasets. Underlined values indicate the best performances among the same set of tasks, and bold-faced values denote the best results across the board. In the last block (4), we conduct pair-wise comparisons between settings to highlight the differences.

5.1 Evaluation on Source Tasks

In this section, we discuss the performance of both fully shared and shared-private multi-task learning (MTL) frameworks on different combinations of DupQuD, SNLI and Multi-NLI datasets as *source tasks*. For shared-private models, we concatenate the representations generated by shared and private (task-specific) encoders to form sentence embeddings. The results can be found in table 10. We compare the performance of MTL with single task learning (STL) technique proposed in Conneau et al. (2017). We see from table 10 that learning from multiple tasks performs better than learning from a single task. For example, MTL model trained on DupQuD, SNLI, and Multi-NLI outperform STL encoder trained on SNLI by 0.8%.

We summarize our findings by addressing the following research questions.

• *Which MTL framework performs better?*
From table 10, it is evident that shared-private MTL performs better than fully shared MTL, especially when more tasks are involved in training. This is intuitive because when more tasks are involved, it becomes harder to get good performance by only modeling the commonalities. It is sensible to have both task-specific and common features so that the final classifier can combine them.

• *Does adversarial training help MTL ?*
To our surprise, the adversarial training does not *always* help shared-private MTL. When there are fewer data and tasks like {DupQuD, SNLI}, adversarial training helps. However, when more tasks are involved, the adversarial learning does not outperform the simple shared-private model, possibly because the SNLI and Multi-NLI datasets share too many commonalities so that it is impossible to completely distinguish them.

• *Which set of tasks/data works the best for MTL?*
We explore pairing DupQuD with the two datasets used in Conneau et al. (2017): SNLI and AllNLI. We also split AllNLI into SNLI and Multi-NLI to conduct 3-task MTL. Table 10 shows that MTL on {DupQuD, SNLI} brings improvements on both datasets, while MTL on {DupQuD, AllNLI} does not simultaneously improve results on all datasets. We hypothesize that it is because the AllNLI dataset dominates DupQuD on the training data size (993k vs. 404k). The experiments on DupQuD, SNLI, and Multi-NLI as separate tasks partially confirm our hypothesis: we see the best results across the board on all three datasets.

5.2 Evaluation on Transfer Tasks

In this part, we discuss the performances of sentence encoders learned by multi-task learning on 10 transfer tasks. Table 11 summarizes the results. For the shared-private MTL models, we report the results of the concatenation of the sentence representations from both the shared and private encoders[5]. We analyze our findings by addressing the following research questions.

[5]We also compared the performance of private and shared encoders. The results can be found in appendix.

Model Type	Sent. Length	PP in Syn. Parse	Adjectives	Adverbs	Proper Nouns	Possessive Pronouns	Cardinal Numbers
Majority vote classifier	-	84.23	75.33	57.04	67.59	63.70	81.46
Sentence representation learning from single-task							
BiLSTM-Max (SNLI)	11.86	95.06	85.23	79.72	93.78	94.42	95.51
BiLSTM-Max (Multi-NLI)	8.56	94.95	86.88	**83.32**	94.06	**96.30**	**96.89**
Sentence representation learning from three-tasks (DupQuD, SNLI and Multi-NLI)							
Shared Private	7.24	95.03	**88.10**	81.91	95.24	95.48	96.75
Adversarial Shared Private	**7.16**	**96.13**	87.05	81.59	**95.66**	94.52	96.49

Table 3: Auxiliary task results for the best two single-task and multi-task learning settings. We use *mean square error* as evaluation metric for sentence length prediction and *F1* score for other tasks. Bold-faced values denote the best results across the board.

Model Type	ROCStories			VIST		
	PM	LSR	PMR	PM	LSR	PMR
Sentence representation learning from single-task						
BiLSTM-Max (SNLI)	75.45	**75.60**	**18.87**	68.74	69.88	9.01
BiLSTM-Max (AllNLI)	**75.92**	**75.60**	18.17	**69.58**	**70.43**	**9.67**
Sentence representation learning from three-tasks (DupQuD, SNLI and Multi-NLI)						
Shared Private	71.98	73.98	15.66	67.36	69.52	8.92
Adversarial Shared Private	70.30	72.85	14.11	65.94	68.40	7.46

Table 4: Sentence reordering results for the best two single-task and multi-task learning settings. Bold-faced values denote the best results across the board.

• *Do sentence encoders learned through MTL help transfer learning?*
Conneau et al. (2017) showed that sentence encoders trained on SNLI outperformed the unsupervised sentence embeddings on 10 transfer tasks, and the model trained on AllNLI further boosted the performances. In table 11, row 4.1 – 4.2 demonstrates that MTL based sentence representations outperform SNLI on 9 transfer tasks and 7 out of 10 for AllNLI (and comparable performances on the other 3 tasks). Row 4.1 shows that the improvements can be as much as 3.0% and 2.0% higher for TREC and MRPC tasks. The results verify our hypothesis that learning from multiple tasks helps to capture more generalizable features that are suitable for transfer learning.

• *Does adversarial training help learning better encoders for transfer tasks?*
We see the adversarial shared-private model results in the best performances on 4 out of 10 transfer tasks, demonstrates its efficiency. Moreover, the improvement on TREC task is as much as 1.8% higher (92.2% vs. 90.4%). We put detailed results in the supplementary materials.

• *How well do the shared and private encoders perform in transfer learning?*
We observe shared encoders perform better on transfer tasks than private encoders, which confirms that shared encoders provide general and common features that are more suitable for transfer learning. However, the concatenation of representations from both shared and private features gets the best results, which indicates that transfer tasks also get benefited from task-specific features because of the commonality with the source tasks. More details about the effect of the number of tasks on the shared encoders can be found in table 7 in the appendix.

• *Which set of tasks/data help MTL to perform better in transfer tasks?*
We investigate the relations between the performance of transfer tasks and the number of tasks or amount of data involved in MTL. As is shown in Table 9, adding more data helps to improve the performance of transfer tasks.

• *Do improvements in transfer learning only come from having more annotated data?*
Comparing the performance of AllNLI in a single task setting and {SNLI, Multi-NLI} in the multi-task settings, we observe significant improvement in 7 out of 10 tasks. In both settings, the amount of training data is the same. To verify the hypothesis

Task	MTL − STL Same Data Size More Tasks	MTL − STL Larger Data Size More Tasks
MR	+0.1	+1.0
CR	+0.6	+0.8
SUBJ	+1.6	+1.3
MPQA	−0.7	−0.6
SST	+0.0	+0.3
TREC	+1.7	+2.1
SICK-R	−0.008	+0.001
SICK-E	−1.5	+0.6
STS14	+0.7	+1.4
MRPC	−0.003	+0.0

Table 5: The effect of more data/tasks on the sentence encoders for transfer learning. The left column shows the differences between MTL and STL given equal amount of annotated data (from different task v.s. from the same task). The right column demonstrates the differences between MTL on two datasets and STL on one dataset (less data).

that the improvements in transfer learning do not solely come from having more annotated data, we design an experiment that samples equal amount of data from SNLI and DupQuD to match the size of SNLI dataset. Table 9 left column shows that MTL is beneficial in this setting. The right column of Table 9 shows that with additional data, MTL achieves larger gains[6].

5.3 Analysis using Auxiliary Tasks

We hypothesize that the universal sentence encoders embed syntactic and semantic information of sentences so that they would be helpful for various transfer tasks. To test this hypothesis and compare the quality of STL and MTL based sentence encoders, we employ 8 auxiliary tasks to analyze what syntactic and semantic information is captured by the sentence embeddings. We pick the best two STL settings and compare with the best shared-private and adversarial shared-private MTL settings. The results are reported on the concatenated sentence representations of both the shared and private encoders from the shared-private MTL settings[7]. We present our findings by addressing the following research questions.

• *Does MTL perform better than STL in capturing syntactic properties of sentences?*
Table 3 shows that sentence embeddings learned by MTL better capture the majority of syntactic properties. However, the performance of STL on Multi-NLI task is significantly better in identifying the existence of adverbs and possessive pronouns in sentences. Adversarial training demonstrates efficiency capturing syntactic information.

• *Does MTL perform better than STL in capturing semantic properties of sentences?*
Our experiments clearly indicate that MTL fails to outperform STL in capturing semantic information. Although MTL outperformed STL in transfer tasks, the semantic auxiliary task experiment unveils the fact that we need more robust MTL architectures to better capture semantic information.

• *Which data/task contains rich syntactic and semantic information that can be captured by the trained encoders?*
In our experiments, the sentence encoder trained on Multi-NLI consistently out-performs other task-specific or shared encoders on both syntactic and semantic auxiliary tasks. Although we consider Multi-NLI as a single task, it consists of sentence pairs from five different genres of spoken and written text. And as a result, sentence encoders trained on Multi-NLI capture more diversified and rich syntactic and semantic features. We hypothesize having annotated data covering a variety of *domains* can better help multi-task learning which has been overlooked by prior work.

6 Conclusion

In this paper, we investigate the effectiveness of multi-task learning (MTL) for training generalizable sentence representations by evaluating on both source tasks and 10 different transfer tasks. Experiments on two categories of MTL frameworks demonstrate that multi-task learning outperforms single-task learning both on the source tasks and most of the transfer tasks. In addition, we analyze what syntactic and semantic information is captured by the sentence representations.

In the future, we will explore more advanced techniques to model domains: in Multi-NLI, there are five different domains, which we do not explicitly model. We will study whether MTL can benefit from proper modeling of domains. In addition, we will explore better models with different learning techniques to improve the semantic information captured by the universal sentence encoder.

[6] more details are provided in section 4 of the supplementary materials.

[7] detailed results of individual private and shared encoders are provided in table 7 and 8 in the appendix.

References

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. VQA: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*. pages 2425–2433.

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

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* .

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb):1137–1155.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* .

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* .

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* .

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, pages 248–255.

Jingjing Gong, Xinchi Chen, Xipeng Qiu, and Xuanjing Huang. 2016. End-to-end neural sentence ordering using pointer network. *arXiv preprint arXiv:1611.04953* .

Felix Hill, Kyunghyun Cho, and Anna Korhonen. 2016. Learning distributed representations of sentences from unlabelled data. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* .

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

Ting-Hao Kenneth Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. 2016. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. pages 1233–1239.

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.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*. pages 1188–1196.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* .

Jiwei Li and Dan Jurafsky. 2017. Neural net models of open-domain discourse coherence. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. *International Conference on Learning Representations* .

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016a. Deep multi-task learning with shared memory. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* .

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016b. Recurrent neural network for text classification with multi-task learning. *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence* .

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* .

Yang Liu, Chengjie Sun, Lei Lin, and Xiaolong Wang. 2016c. Learning natural language inference using bidirectional lstm model and inner-attention. *arXiv preprint arXiv:1605.09090* .

Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2016. Multi-task sequence to sequence learning. *International Conference on Learning Representations* .

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics* 19(2):313–330.

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.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories .

Lili Mou, Zhao Meng, Rui Yan, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2016. How transferable are neural networks in nlp applications? *Proceedings of the 2016 conference on empirical methods in natural language processing* .

Hao Peng, Sam Thomson, and Noah A Smith. 2017. Deep multitask learning for semantic dependency parsing. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* .

Nanyun Peng and Mark Dredze. 2017. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*. pages 91–100.

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*. pages 1532–1543.

Marek Rei. 2017. Semi-supervised multitask learning for sequence labeling. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* .

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2017. Bidirectional attention flow for machine comprehension. *International Conference on Learning Representations* .

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. pages 806–813.

Damien Teney, Lingqiao Liu, and Anton van den Hengel. 2017. Graph-structured representations for visual question answering. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* .

Subhashini Venugopalan, Lisa Anne Hendricks, Marcus Rohrbach, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2017. Captioning images with diverse objects. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Shuohang Wang and Jing Jiang. 2016. Learning natural language inference with lstm. *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* .

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* .

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

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426* .

Honglun Zhang, Liqiang Xiao, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2017a. Multi-task label embedding for text classification. *arXiv preprint arXiv:1710.07210* .

Honglun Zhang, Liqiang Xiao, Yongkun Wang, and Yaohui Jin. 2017b. A generalized recurrent neural architecture for text classification with multi-task learning. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence* .

Han Zhao, Zhengdong Lu, and Pascal Poupart. 2015. Self-adaptive hierarchical sentence model. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*. pages 4069–4076.

Will Y Zou, Richard Socher, Daniel Cer, and Christopher D Manning. 2013. Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. pages 1393–1398.

A Multi-task Learning Frameworks

The following graphical illustration summarizes the fully shared and shared private multi-task learning frameworks that we explore in this work.

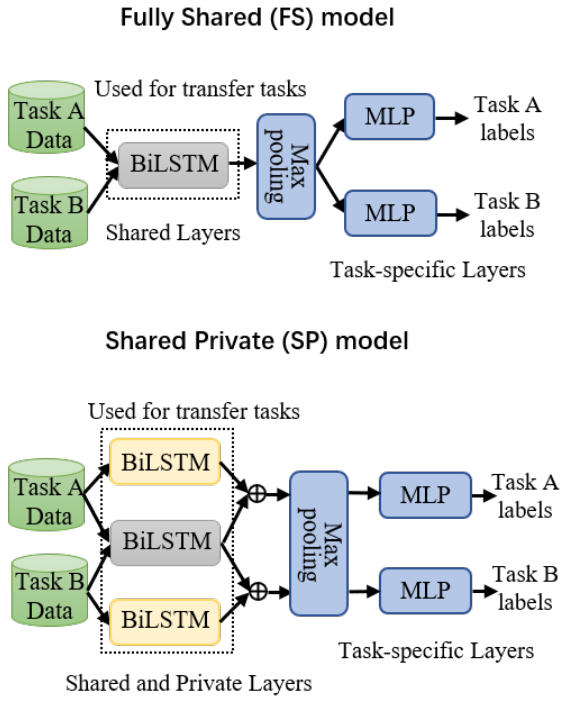


Figure 1: Two different multi-task learning frameworks. $\oplus$ represents concatenation of sentence vectors. Yellow and gray boxes represent private and shared sentence encoders respectively. These encoders, after trained by multi-task learning, will be used for transfer learning.

B Adversarial training

The goal of adversarial training is to encourage shared space to be purer and ensure the shared representation not be contaminated by task-specific features. The discriminator is defined as,

$$D(s^k) = \textit{softmax}(W s^k + b),$$

where $W \in R^{d \times d}$ and $b \in R^d$ are model parameters. Optimizing the adversarial loss,

$$L_{adv} = \min_{\theta_E} \left(\max_{\theta_D} \left(\sum_{k=1}^{K} \sum_{i=1}^{N_k} d_i^k \log[D(E(w))] \right) \right)$$

has two competing goals: the discriminator tries to maximize the classification accuracy (inside the parentheses), and the sentence encoder tries to confuse it (and thus minimize the classification accuracy). E and D represents the shared sentence encoder and the discriminator respectively and θ_E and θ_D are the model parameters of E and D. d_i^k denotes the ground-truth label indicating the type of the current task. To encourage the shared and private encoders to capture different aspects of the sentences, the following term is added.

$$L_{diff} = \sum_{k=1}^{K} \left\| H_s^{k\top} H_p^k \right\|_F^2.$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm. Here, H_s^k and H_p^k are matrices where rows are the hidden vectors (see Eq. (1)) generated by the shared and private encoders given an input sentence of task k.

The final loss function is a weighted combination of three parts:

$$L = L_{multi-task} + \beta L_{adv} + \gamma L_{diff},$$

Name	N	V	Task	C
Binary and multi-class classification tasks				
MR	11k	20.3k	sentiment	2
CR	4k	5.7k	product review	2
SUBJ	10k	22.6k	subj/obj	2
MPQA	11k	6.2k	opinion	2
SST	70k	17.5k	sentiment	2
TREC	6k	9.7k	question-type	6
Recognizing textual entailment tasks				
SNLI†	560k	42.7k	entailment	3
Multi-NLI†	433k	102.7k	entailment	3
SICK-E	10k	2.4k	entailment	3
Paraphrase identification tasks				
DupQuD†	404k	127.5k	paraphrasing	2
MRPC	5.8k	19.5k	paraphrasing	2
Semantic textual similarity tasks				
SICK-R	10k	2.4k	similarity	0 – 5
STS-14	4.5k	11k	similarity	0 – 5

Table 6: Statistics of the datasets for multi-task learning and the transfer tasks. N is the number of samples, V is the vocabulary size, and C is the number of classes or score range. † denotes the datasets that are used in multi-task learning.

Tasks	β	γ
DupQuD and SNLI	0.01	0.05
SNLI and Multi-NLI	0.005	0.001
DupQuD and AllNLI	0.01	0.05
DupQuD, SNLI and Multi-NLI	0.005	0.001

Table 7: Best β and γ values for adversarial shared private model on different set of tasks.

where β and γ are hyper-parameters, $L_{multi-task}$ refers to a simple summation over the cross entropy loss for each task. We tune β and γ in the range $[0.001, 0.005, 0.01, 0.05, 0.1, 0.5]$ and found different combinations perform best for different models. Table 7 indicates the best β and γ values for different multi-task learning settings.

C Dataset Details

Details of the experimental datasets are provided in table 6 and 8.

	PTB		ROCStories		VIST	
	N	S_{Avg}	N	S_{Avg}	N	S_{Avg}
Train	39,832	25.5	45,496	10.2	40,155	11.3
Dev	1,700	25.1	1,871	10.1	4,990	11.3
Test	2,416	25.1	1,871	10.1	5,055	11.5

Table 8: Statistics of the auxiliary task datasets. For ROCStories and VIST, N indicates the number of stories and for PTB, N indicates the number of sentences. S_{Avg} is the average sentence length. For ROCStroies, spring 2016 version is used in the experiments.

D Additional Experiments

Task	**MTL − STL** Same Data Size More Tasks	**MTL − STL** Larger Data Size More Tasks
MR	+0.1	+1.0
CR	+0.6	+0.8
SUBJ	+1.6	+1.3
MPQA	−0.7	−0.6
SST	+0.0	+0.3
TREC	+1.7	+2.1
SICK-R	−0.008	+0.001
SICK-E	−1.5	+0.6
STS14	+0.7	+1.4
MRPC	−0.003	+0.0

Table 9: The effect of more data/tasks on the sentence encoders for transfer learning. The left column shows the differences between MTL and STL given equal amount of annotated data (from different task v.s. from the same task). The right column demonstrates the differences between MTL on two datasets and STL on one dataset (less data).

We conduct an additional experiment by sampling 225k training examples for SNLI and DupQuD datasets, which equals to the total size of the SNLI dataset and we found 0.26% average improvement in transfer tasks compared to single task learning on the SNLI dataset. Performance differences between the two settings for each transfer tasks are listed in table 9 (in the second column).

Our experiments show that both the learning signal from different tasks and the data size play important roles in achieving the better performances. If we consider the complete SNLI and DupQuD dataset, we observe 0.69% average improvement in transfer tasks compared to single task learning on SNLI dataset. Performance differences between the two settings for each transfer tasks are listed in table 9 (in the third column).

So, we can conclude that the improvements in transfer tasks do not solely come from having more annotated data, modeling multiple tasks through MTL learning framework helps.

Model Type	Quora		SNLI		Multi-NLI	
	dev	test	dev	test	dev	test
In domain single task sentence representation learning						
BiLSTM-Max	87.14	86.73	84.65	84.35	70.21/70.83	70.76/69.84
Sentence representation learning from two-datasets and two tasks (SNLI and Multi-NLI)						
Fully Shared	-	-	85.14	85.10	71.39/71.23	70.77/70.10
Shared-Private	-	-	85.02	**85.25**	**71.72/71.42**	**71.83/70.56**
Adversarial Shared-Private	-	-	84.92	84.93	70.89/71.37	71.02/69.97
Sentence representation learning from two-datasets and two tasks (Quora and SNLI)						
Fully Shared	86.82	86.62	84.92	84.81	-	-
Shared-Private	87.01	86.76	84.80	84.73	-	-
Adversarial Shared-Private	87.50	**87.01**	85.21	84.93	-	-
Sentence representation learning from three-datasets and two tasks (Quora and AllNLI)						
Fully Shared	87.31	86.84	**85.22**	84.80	69.93/70.26	70.00/68.88
Shared-Private	86.85	86.03	85.16	84.73	70.66/70.54	70.81/69.25
Adversarial Shared-Private	86.99	86.58	84.66	84.27	70.23/69.41	69.64/68.33
Sentence representation learning from three-datasets and three tasks (Quora, SNLI and Multi-NLI)						
Fully Shared	86.62	85.92	84.33	84.28	70.28/69.74	70.05/69.58
Shared-Private	**87.63**	86.98	85.15	85.18	71.17/71.03	71.03/70.12
Adversarial Shared-Private	86.57	86.29	84.63	84.69	70.72/70.70	71.01/70.11

Table 10: Development and test accuracy of source tasks (Quora, SNLI and Multi-NLI) obtained through various multi-task learning architectures. Underlined values indicate best performance among models trained on same set of tasks and bold-faced values indicate best performance among all models.

Model Type	MR	CR	SUBJ	MPQA	SST	TREC	SICK-R	SICK-E	MRPC	STS14
Sentence representation learning from single-task										
BiLSTM-Max (on SNLI)	80.1	85.3	92.6	89.1	83.6	89.2	0.885	86.0	75.2/82.4	.66/.64
BiLSTM-Max (on Quora)	79.2	84.6	92.6	88.8	83.5	88.0	0.861	82.4	74.8/82.8	.62/.60
BiLSTM-Max (on Multi-NLI)	81.2	85.8	93.1	89.5	83.4	88.8	0.863	84.7	75.9/83.1	.66/.63
BiLSTM-Max (on AllNLI)	80.9	86.3	93.2	89.2	83.3	88.8	0.887	86.7	76.4/83.4	.69/.66
Sentence representation learning from two-tasks (Quora and SNLI)										
Fully Shared	79.5	84.6	92.6	89.1	82.0	88.6	0.882	85.4	74.1/82.3	.69/.66
Shared-Private	80.5	84.8	93.4	89.1	84.0	90.2	0.881	86.1	75.1/83.2	.65/.62
Adversarial Shared-Private	80.9	85.4	93.4	89.2	83.6	90.8	0.886	86.9	76.5/82.9	.68/.65
Sentence representation learning from two-tasks (SNLI and Multi-NLI)										
Fully Shared	81.5	**87.2**	92.7	89.3	84.0	89.4	0.883	86.7	75.9/82.8	.69/.66
Shared-Private	81.7	86.4	93.7	89.6	84.8	89.2	0.885	86.7	76.3/82.7	.67/.64
Adversarial Shared-Private	81.2	86.0	93.0	89.3	83.7	90.4	0.886	**87.1**	76.9/83.5	.70/.67
Sentence representation learning from two-tasks (Quora and AllNLI)										
Fully Shared	81.4	86.9	93.0	**89.7**	**85.2**	89.2	0.879	86.1	75.7/82.9	**.71/.67**
Shared-Private	**82.0**	86.1	**93.9**	89.4	84.6	89.6	0.884	86.3	76.4/83.4	.68/.64
Adversarial Shared-Private	81.4	86.3	93.2	89.4	85.1	88.4	**0.888**	86.6	75.5/82.5	.67/.63
Sentence representation learning from three-tasks (Quora, SNLI and Multi-NLI)										
Fully Shared	80.9	85.4	92.5	89.4	83.4	89.8	0.882	85.8	75.3/82.5	.70/.67
Shared-Private	81.6	86.9	**93.9**	89.2	84.4	90.4	0.883	85.9	76.5/83.3	.66/.63
Adversarial Shared-Private	**82.0**	86.3	93.8	89.4	84.1	**92.2**	0.884	87.0	**77.2/83.6**	.68/.65

Table 11: Transfer test results for various single-task and multi-task learning architectures trained on a combination of Quora, SNLI and Multi-NLI datasets. Underlined values indicates the best performance among models trained on same set of tasks. Bold-faced values indicate the best performance among all models in this table.

Model Type	MR	CR	SUBJ	MPQA	SST	TREC	SICK-R	SICK-E	MRPC	STS14
Shared-Private (trained on DupQuD and SNLI)										
Private Encoder (on DupQuD)	79.8	84.1	92.4	88.6	81.3	88.4	0.845	82.5	73.9/81.6	.60/.58
Private Encoder (on SNLI)	79.6	84.8	92.6	89.0	82.5	87.2	0.880	85.6	75.2/83.2	.66/.63
Shared Encoder	79.7	85.0	93.0	89.2	82.3	90.2	0.874	84.3	75.0/83.1	.65/.62
Combined Encoder	80.5	84.8	93.4	89.1	84.0	90.2	0.881	86.1	75.1/83.2	.65/.62
Adversarial Shared-Private (trained on DupQuD and SNLI)										
Private Encoder (on DupQuD)	79.5	83.1	92.1	88.6	82.1	89.0	0.860	83.4	73.9/82.5	.65/.62
Private Encoder (on SNLI)	79.4	84.9	92.6	88.9	82.4	87.8	0.883	85.7	74.6/82.4	.67/.65
Shared Encoder	80.0	84.3	92.6	89.2	81.9	86.6	0.883	85.4	73.9/81.5	.69/.65
Combined Encoder	80.9	85.4	93.4	89.2	83.6	90.8	0.886	86.9	76.5/82.9	.68/.65
Shared-Private (trained on SNLI and Multi-NLI)										
Private Encoder (on SNLI)	79.5	84.0	92.7	89.1	82.0	87.8	0.881	84.8	75.0/82.7	.65/.63
Private Encoder (on Multi-NLI)	80.6	84.6	92.7	89.2	82.9	88.0	0.853	83.8	75.1/82.9	.60/.58
Shared Encoder	80.9	86.4	92.9	89.5	84.0	88.0	0.879	84.8	75.8/83.1	.69/.65
Combined Encoder	81.7	86.4	93.7	**89.6**	84.8	89.2	0.885	86.7	76.3/82.7	.67/.64
Adversarial Shared-Private (trained on SNLI and Multi-NLI)										
Private Encoder (on SNLI)	79.4	84.6	92.1	89.0	82.8	86.6	0.886	85.5	74.0/81.4	.68/.66
Private Encoder (on Multi-NLI)	80.4	84.6	92.6	89.1	83.3	86.8	0.863	83.4	76.0/83.3	.65/.63
Shared Encoder	81.2	86.7	92.4	89.3	84.5	87.0	0.875	85.1	74.8/82.6	.56/.57
Combined Encoder	81.7	86.5	93.4	89.5	**84.9**	90.0	**0.888**	**87.1**	76.4/83.4	.64/.63
Shared-Private (trained on DupQuD and AllNLI)										
Private Encoder (on DupQuD)	79.4	82.6	92.5	88.5	82.2	89.2	0.856	82.9	74.3/82.4	.63/.59
Private Encoder (on AllNLI)	81.1	86.1	92.9	89.5	83.9	90.2	0.876	85.0	76.0/83.3	.67/.64
Shared Encoder	81.2	85.6	93.1	89.2	83.4	88.2	0.880	85.3	75.7/82.8	.69/.66
Combined Encoder	**82.0**	86.1	**93.9**	89.4	84.6	89.6	0.884	86.2	76.4/83.4	.68/.64
Adversarial Shared-Private (trained on DupQuD and AllNLI)										
Private Encoder (on DupQuD)	79.2	81.7	92.1	88.7	80.8	86.6	0.865	83.8	74.1/82.2	.67/.64
Private Encoder (on AllNLI)	81.5	86.5	92.8	89.4	82.9	88.4	0.885	85.5	75.7/83.2	**.70/.67**
Shared Encoder	80.5	84.8	92.6	89.2	83.2	82.6	0.876	84.8	75.5/83.1	.57/.56
Combined Encoder	81.9	85.9	93.0	**89.6**	82.4	90.6	0.887	86.7	76.8/83.3	.61/.60
Shared-Private (trained on DupQuD, SNLI and Multi-NLI)										
Private Encoder (on DupQuD)	78.9	83.5	91.7	88.3	81.4	88.6	0.850	82.1	72.6/81.1	.62/.59
Private Encoder (on SNLI)	79.1	83.9	92.7	89.0	81.3	88.4	0.880	85.2	74.0/82.0	.66/.63
Private Encoder (on Multi-NLI)	81.0	85.7	93.1	89.3	82.9	89.2	0.850	84.0	74.5/82.9	.60/.58
Shared Encoder	80.9	85.8	92.9	89.2	82.7	85.2	0.878	85.8	76.0/83.1	.68/.65
Combined Encoder	81.6	**86.9**	**93.9**	89.2	84.4	90.4	0.883	85.9	76.5/83.2	.66/.63
Adversarial Shared-Private (trained on DupQuD, SNLI and Multi-NLI)										
Private Encoder (on DupQuD)	78.9	82.3	92.2	88.8	82.3	87.2	0.855	83.0	74.3/82.2	.64/.62
Private Encoder (on SNLI)	79.6	84.4	92.0	89.0	82.7	88.2	0.881	85.4	74.6/82.4	.67/.65
Private Encoder (on Multi-NLI)	80.6	84.7	93.0	89.1	83.6	89.2	0.863	84.8	75.8/82.8	.65/.62
Shared Encoder	81.0	85.9	92.7	**89.6**	82.9	87.0	0.876	85.7	74.8/82.8	.66/.64
Combined Encoder	**82.0**	86.3	93.8	89.4	84.1	**92.2**	0.884	87.0	**77.2/83.6**	.66/.64

Table 12: Detailed analysis of the transfer test results for shared-private models trained on different combinations of DupQuD, SNLI and Multi-NLI datasets. Combined encoder means the concatenation of shared encoder and all private encoders. Underlined values indicate the best performance among different encoders of the shared-private models trained on the same set of tasks. Bold-faced values indicate the best performance among all models in this table.

Model Type	Sent. Length	PP in Syn. Parse	Adjectives	Adverbs	Proper Nouns	Possessive pronoun	Cardinal Number
Sentence representation learning from single-task							
BiLSTM-Max (on SNLI)	11.86	95.06	85.23	79.72	93.78	94.42	95.51
BiLSTM-Max (on DupQuD)	12.56	93.92	86.85	77.69	95.14	93.25	95.59
BiLSTM-Max (on Multi-NLI)	8.56	94.95	86.88	**83.32**	94.06	96.30	96.89
BiLSTM-Max (on AllNLI)	10.32	95.37	86.91	82.98	93.94	96.27	95.14
Shared-Private (trained on DupQuD, SNLI, and Multi-NLI)							
Private Encoder (on DupQuD)	13.44	93.91	83.61	76.05	94.35	90.73	94.36
Private Encoder (on SNLI)	12.34	93.83	84.66	76.47	94.16	93.46	94.69
Private Encoder (on Multi-NLI)	9.80	95.37	87.36	81.97	94.07	**96.98**	**97.10**
Shared Encoder	11.17	94.41	86.83	81.30	94.90	93.05	95.36
Combined Encoder	7.24	95.03	**88.10**	81.91	95.24	95.48	96.75
Adversarial Shared-Private (trained on DupQuD, SNLI, and Multi-NLI)							
Private Encoder (on DupQuD)	13.44	94.50	85.52	72.72	94.20	91.98	94.55
Private Encoder (on SNLI)	12.34	95.02	86.18	77.70	93.59	93.61	94.17
Private Encoder (on Multi-NLI)	9.80	94.17	86.80	80.94	93.87	96.03	96.53
Shared Encoder	9.73	93.20	85.94	79.60	94.56	87.51	94.69
Combined Encoder	**7.16**	**96.13**	87.05	81.59	**95.66**	94.52	96.49

Table 13: Auxiliary task results for the best two single-task and multi-task learning settings trained on different combinations of DupQuD, SNLI and Multi-NLI datasets. We use *mean square error* as evaluation metric for sentence length prediction and *F1* score for other tasks. Underlined values indicate the best performance among different encoders of the shared-private models trained on the same set of tasks. Bold-faced values denote the best results across the board.

Model Type	ROCStories			VIST		
	PM	LSR	PMR	PM	LSR	PMR
Sentence representation learning from single-task						
BiLSTM-Max (on SNLI)	75.45	**75.60**	**18.87**	68.74	69.88	9.01
BiLSTM-Max (on DupQuD)	74.54	74.36	16.52	68.11	69.50	8.25
BiLSTM-Max (on Multi-NLI)	74.04	74.74	17.53	68.13	69.88	**9.77**
BiLSTM-Max (on AllNLI)	**75.92**	**75.60**	18.17	**69.58**	**70.43**	9.67
Shared-Private (trained on DupQuD, SNLI, and Multi-NLI)						
Private Encoder (on DupQuD)	71.29	72.97	14.11	66.66	68.93	8.09
Private Encoder (on SNLI)	73.91	74.42	16.09	67.73	69.40	8.72
Private Encoder (on Multi-NLI)	73.59	74.25	15.39	68.40	69.60	8.92
Shared Encoder	71.98	73.98	15.66	67.36	69.52	8.92
Adversarial Shared-Private (trained on DupQuD, SNLI, and Multi-NLI)						
Private Encoder (on DupQuD)	70.80	72.56	13.31	66.22	68.78	8.11
Private Encoder (on SNLI)	71.32	73.36	15.07	67.15	69.40	8.43
Private Encoder (on Multi-NLI)	73.68	74.85	17.37	68.06	69.88	8.80
Shared Encoder	70.30	72.85	14.11	65.94	68.40	7.46

Table 14: Sentence ordering results for the best two single-task and multi-task learning settings. Underlined values indicate the best performance among different encoders of the shared-private models trained on the same set of tasks. Bold-faced values denote the best results across the board. Due to the structure of the sentence ordering model, it is not feasible to conduct experiment on the combined encoder, so we skipped it.