Same Representation, Different Attentions: Shareable Sentence Representation Learning from Multiple Tasks

Renjie Zheng*, Junkun Chen and Xipeng Qiu†
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University
School of Computer Science, Fudan University, 825 Zhangheng Road, Shanghai, China

Abstract
Distributed representation plays an important role in deep learning based natural language processing. However, the representation of a sentence often varies in different tasks, which is usually learned from scratch and suffers from the limited amounts of training data. In this paper, we claim that a good sentence representation should be invariant and can benefit the various subsequent tasks. To achieve this purpose, we propose a new scheme of information sharing for multi-task learning. More specifically, all tasks share the same sentence representation and each task can select the task-specific information from the shared sentence representation with attention mechanisms. The query vector of each task’s attention could be either static parameters or generated dynamically. We conduct extensive experiments on 16 different text classification tasks, which demonstrate the benefits of our architecture. Source codes of this paper are available on Github1.

1 Introduction
The distributed representation plays an important role in deep learning based natural language processing (NLP). On the word level, many successful methods have been proposed to learn a good representation for single word, which are also called word embeddings, such as skip-gram [Mikolov et al., 2013], GloVe [Pennington et al., 2014], etc. There are also pre-trained word embeddings, which can easily be used in downstream tasks. However, on the sentence level, there is still no generic sentence representation which is suitable for various NLP tasks.

Currently, most sentence encoding models are trained specifically for a certain task in a supervised way, which results to different representations for the same sentence in different tasks. Taking the following sentence as an example for domain classification task and sentiment classification task,

The infantile cart is easy to use.

Figure 1: Three schemes of information sharing in multi-task learning. (a) stacked shared-private scheme, (b) parallel shared-private scheme, (c) our proposed attentive sharing scheme.

general text classification models always learn two representations separately. For domain classification, the model can learn a better representation of “infantile cart” while for sentiment classification, the model is able to learn a better representation of “easy to use”.

However, to train a good task-specific sentence representation from scratch, we always need to prepare a large dataset which is always unavailable or costly. To alleviate this problem, one approach is pre-training the model on large unlabeled corpora by unsupervised learning tasks, such as language modeling. This unsupervised pre-training may be helpful to improve the final performance, but the improvement is not guaranteed since it does not directly optimize the desired task.

Another approach is multi-task learning [Caruana, 1997], which is an effective approach to improve the performance of a single task with the help of other related tasks. However, most existing models in multi-task learning attempt to divide the representation of a sentence into private and shared spaces. The shared representation is used in all tasks, and the private one is different for each task. The two typical information sharing schemes are stacked shared-private scheme and parallel shared-private scheme (as shown in Figure 1(a) and 1(b) respectively). However, we cannot guarantee that a good sentence encoding model is learned by the shared layer.

To learn a better shareable sentence representation, we propose a new information-sharing scheme for multi-task learning in this paper. In our proposed scheme, the representation of every sentence is fully shared among all different tasks. To extract the task-specific feature, we utilize the attention mechanism and introduce a task-dependent query vector to

*Current address: Oregon State University, Corvallis, OR, USA
† Corresponding Author, xpiu@fudan.edu.cn
1 https://github.com/renj/attentive-representation
select the task-specific information from the shared sentence representation. The query vector of each task can be regarded as learnable parameters (static) or be generated dynamically. If we take the former example, in our proposed model these two classification tasks share the same representation which includes both domain information and sentiment information. On top of this shared representation, a task-specific query vector will be used to focus “infantile cart” for domain classification and “easy to use” for sentiment classification.

The contributions of this papers can be summarized as follows.

- We propose a new information sharing scheme for multi-task learning. As a side effect, the model can be easily visualized and shows what specific parts of the sentence are focused in different tasks.
- In our proposed scheme, we can learn a shareable generic sentence representation, which can be easily transferred to other tasks. The shareable sentence representation can also be improved by the auxiliary tasks, such as POS Tagging and Chunking.
- We conduct extensive experiments on 16 sentiment classification tasks. Experiments show that our proposed model is space efficient and converges quickly.

2 Sentence Encoding in Multi-task Learning

2.1 Neural Sentence Encoding Model

The primary role of sentence encoding models is to represent the variable-length sentence or paragraphs as fixed-length dense vector (distributed representation). Currently, the effective neural sentence encoding models include neural Bag-of-words (NBOW), recurrent neural networks (RNN) [Sutskever et al., 2014; Chung et al., 2014], convolutional neural networks (CNN) [Collobert et al., 2011; Kim, 2014], and syntactic-based compositional model [Socher et al., 2013; Tai et al., 2015; Zhu et al., 2015; Ma et al., 2015].

Given a text sequence \( x = \{ x_1, x_2, \cdots, x_T \} \), we first use a lookup layer to get the vector representation (word embedding) \( x_i \) of each word \( x_i \). Then we can use CNN or RNN to calculate the hidden state \( h_i \) of each position \( i \). The final representation of a sentence can be either the final hidden state of the RNN or the max (or average) pooling from all hidden states of RNN (or CNN).

We use bidirectional LSTM (BiLSTM) to gain some dependency between adjacent words. The update rule of each LSTM unit can be written as follows:

\[
\begin{align*}
\overrightarrow{h}_t &= \text{LSTM}(\overrightarrow{h}_{t-1}, x_i, \theta_p), \\
\overleftarrow{h}_t &= \text{LSTM}(\overleftarrow{h}_{t+1}, x_i, \theta_p), \\
h &= \frac{1}{T} \sum_{t=1}^{T} \overrightarrow{h}_t \oplus \overleftarrow{h}_t,
\end{align*}
\]

where \( \theta_p \) represents all the parameters of BiLSTM. The representation of the whole sequence is the average of the hidden states of all the positions, where \( \oplus \) denotes the concatenation operation.

2.2 Shared-Private Scheme in Multi-task Learning

Multi-task Learning [Caruana, 1997] utilizes the correlation between related tasks to improve classification by learning tasks in parallel, which has been widely used in various natural language processing tasks.

To facilitate this, we give some explanation for notations used in this paper. Formally, we refer to \( D_k \) as a dataset with \( N_k \) samples for task \( k \). Specifically,

\[
D_k = \{ (x_i^{(k)}, y_i^{(k)}) \}_{i=1}^{N_k}
\]

where \( x_i^{(k)} \) and \( y_i^{(k)} \) denote a sentence and corresponding label for task \( k \).

Shared-Private Scheme

A common information sharing scheme is to divide the feature spaces into two parts: one is used to store task-specific features, the other is used to capture task-invariant features. As shown in Figure 1(a) and 1(b), there are two schemes: stacked shared-private (SSP) scheme and parallel shared-private (PSP) scheme.

In stacked scheme, the output of the shared LSTM layer is fed into the private LSTM layer, whose output is the final task-specific sentence representation. In parallel scheme, the final task-specific sentence representation is the concatenation of outputs from the shared LSTM layer and the private LSTM layer.

Task-Specific Output Layer

For a sentence \( x^{(k)} \) and its label \( y^{(k)} \) in task \( k \), its final representation is ultimately fed into the corresponding task-specific softmax layer for classification or other tasks.

\[
\hat{y}^{(k)} = \text{softmax}(W^{(k)}h^{(k)} + b^{(k)})
\]

where \( \hat{y}^{(k)} \) is prediction probabilities; \( h^{(k)} \) is the final task-specific representation; \( W^{(k)} \) and \( b^{(k)} \) are task-specific weight matrix and bias vector respectively.

The total loss \( L_{task} \) can be computed as:

\[
\mathcal{L}_{All} = \sum_{k=1}^{K} \alpha_k \mathcal{L}_{Task}(\hat{y}^{(k)}, y^{(k)})
\]

where \( \alpha_k \) (usually set to 1) is the weights for each task \( k \) respectively; \( \mathcal{L}_{Task}(\hat{y}, y) \) is the cross-entropy of the predicted and true distributions.

3 A New Information-Sharing Scheme for Multi-task Learning

The key factor of multi-task learning is the information sharing scheme in latent representation space. Different from the traditional shared-private scheme, we introduce a new scheme for multi-task learning on NLP tasks, in which the sentence representation is shared among all the tasks, the task-specific information is selected by attention mechanism.

In a certain task, not all information of a sentence is useful for the task, therefore we just need to select the key information from the sentence. Attention mechanism [Bahdanau et al., 2014; Mnih et al., 2014] is an effective method to select
related information from a set of candidates. The attention mechanism can effectively solve the capacity problem of sequence models, thereby is widely used in many NLP tasks, such as machine translation [Luong et al., 2015], textual entailment [Zhao et al., 2016] and summarization [Rush et al., 2015].

### 3.1 Static Task-Attentive Sentence Encoding

We first introduce the static task-attentive sentence encoding model, in which the task query vector is a static learnable parameter. As shown in Figure 2, our model consists of one shared BiLSTM layer and an attention layer. Formally, for a sentence in task $k$, we first use BiLSTM to calculate the shared representation $[h_{1}, \cdots, h_{T}]$. Then we use attention mechanism to select the task-specific information from a generic task-independent sentence representation. Following [Luong et al., 2015], we use the dot-product attention to compute the attention distribution. We introduce a task-specific query vector $q^{(k)}$ to calculate the attention distribution $\alpha^{(k)}$ over all positions:

$$
\alpha^{(k)}_t = \text{softmax}(q^{(k)T}h_t),
$$

where the task-specific query vector $q^{(k)}$ is a learned parameter. The final task-specific representation $c^{(k)}$ is

$$
c^{(k)} = \sum_{t=1}^{T} \alpha^{(k)}_t h_t.
$$

At last, a task-specific fully connected layer followed by a softmax non-linear layer processes the task-specific context $c^{(k)}$ and predicts the probability distribution over classes.

### 3.2 Dynamic Task-Attentive Sentence Encoding

Different from the static task-attentive sentence encoding model, the query vectors of the dynamic task-attentive sentence encoding model are generated dynamically. When each task belongs to a different domain, we can introduce an auxiliary domain classifier to predict the domain (or task) of the specific sentence. Thus, the domain information is also included in the shared sentence representation, which can be used to generate the task-specific query vector of attention.

The original tasks and the auxiliary task of domain classification (DC) are joint learned in our multi-task learning framework.

The query vector $q^{(DC)}$ of DC task is static and needs be learned in training phrase. The domain information is also selected with attention mechanisms.

$$
a^{(DC)}_t = \text{softmax}(q^{(DC)T}h_t),
$$

$$
c^{(DC)} = \sum_{t=1}^{T} a^{(DC)}_t h_t,
$$

$$
\hat{y}^{(DC)} = \text{softmax}(W^{(DC)}c^{(DC)} + b^{(DC)}),
$$

where $a^{(DC)}$ is attention distribution of auxiliary DC task, and $c^{(DC)}$ is the attentive information for DC task, which is fed into the final classifier to predict its domain $\hat{y}^{(DC)}$.

Since $c^{(DC)}$ contains the domain information, we can use it to generate a more flexible query vector

$$
q^{(k)} = Uc^{(DC)} + b^{(k)},
$$

where $U$ is a shared learnable weight matrix and $b^{(k)}$ is a task-specific bias vector. When we set $U = 0$, the dynamic query is equivalent to the static one.

### 4 Experiment

In this section, we investigate the empirical performances of our proposed architectures on three experiments.

#### 4.1 Exp 1: Sentiment Classification

We first conduct a multi-task experiment on sentiment classification.

**Dataset**

We use 16 different datasets from several popular review corpora used in [Liu et al., 2017]. These datasets consist of 14 product review datasets and two movie review datasets. All the datasets in each task are partitioned randomly into training, development and testing sets with the proportion of 70%, 10% and 20% respectively. The detailed statistics about all the datasets are listed in Table 1.

**Competitor Methods**

We compare our proposed two information sharing schemes, static attentive sentence encoding (SA-MTL) and dynamic attentive sentence encoding (DA-MTL), with the following multi-task learning frameworks.

---

Figure 2: Static Task-Attentive Sentence Encoding

Figure 3: Dynamic Task-Attentive Sentence Encoding
Table 1: Statistics of the 16 datasets. The columns 2-5 denote the number of samples in training, development and test sets. The last two columns represent the average length and vocabulary size of the corresponding dataset.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Train</th>
<th>Dev.</th>
<th>Test</th>
<th>Avg. L</th>
<th>Vocab.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Books</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>159</td>
<td>62K</td>
</tr>
<tr>
<td>Elec.</td>
<td>1398</td>
<td>200</td>
<td>400</td>
<td>101</td>
<td>30K</td>
</tr>
<tr>
<td>DVD</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>173</td>
<td>69K</td>
</tr>
<tr>
<td>Kitchen</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>89</td>
<td>28K</td>
</tr>
<tr>
<td>Apparel</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>57</td>
<td>21K</td>
</tr>
<tr>
<td>Camera</td>
<td>1397</td>
<td>200</td>
<td>400</td>
<td>130</td>
<td>26K</td>
</tr>
<tr>
<td>Health</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>81</td>
<td>26K</td>
</tr>
<tr>
<td>Music</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>136</td>
<td>60K</td>
</tr>
<tr>
<td>Toys</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>90</td>
<td>28K</td>
</tr>
<tr>
<td>Video</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>156</td>
<td>57K</td>
</tr>
<tr>
<td>Baby</td>
<td>1300</td>
<td>200</td>
<td>400</td>
<td>104</td>
<td>26K</td>
</tr>
<tr>
<td>Mag.</td>
<td>1370</td>
<td>200</td>
<td>400</td>
<td>117</td>
<td>30K</td>
</tr>
<tr>
<td>Soft.</td>
<td>1315</td>
<td>200</td>
<td>400</td>
<td>129</td>
<td>26K</td>
</tr>
<tr>
<td>Sports</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>94</td>
<td>30K</td>
</tr>
<tr>
<td>IMDB</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>269</td>
<td>44K</td>
</tr>
<tr>
<td>MR</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>21</td>
<td>12K</td>
</tr>
</tbody>
</table>

Table 2 shows the performances of the different methods. From the table, we can see that the performances of most tasks can be improved with the help of multi-task learning. FS-MTL shows the minimum performance gain from multi-task learning since it puts all private and shared information into a unified space. SSP-MTL and PSP-MTL achieve similar performance and are outperformed by ASP-MTL which can better separate the task-specific and task-invariant features by using adversarial training. Our proposed models (SA-MTL and DA-MTL) outperform ASP-MTL because we model a richer representation from these 16 tasks. Compared to SA-MTL, DA-MTL achieves a further improvement of +0.6 accuracy with the help of the dynamic and flexible query vector. It is noteworthy that our models are also space efficient since the task-specific information is extracted by using only a query vector, instead of a BiLSTM layer in the shared-private models.

We also present the convergence properties of our models on the development datasets compared to other multi-task models in Figure 4. We can see that PSP-MTL converges much more slowly than the rest four models because each task-specific classifier should consider the output of shared layer which is quite unstable during the beginning of training phase. Moreover, benefit from the attention mechanism which is useful in feature extraction, SA-TML and DA-MTL are converged much more quickly than the rest of models.

**Visualization**

Since all the tasks share the same sentence encoding layer, the query vector \( q \) of each task determines which part of the sentence to attend. Thus, similar tasks should have similar query vectors. Here we simply calculate the Frobenius norm of each pair of tasks’ \( q \) as the similarity. Figure 5 shows the similarity matrix of different task’s query vector \( q \) in static attentive model. A darker cell means the higher similarity of the two task’s \( q \). Since the cells in the diagonal of the matrix denotes the similarity of one task, we leave them blank because they are meaningless. It’s easy to find that \( q \) of “DVD”, “Video” and “IMDB” have very high similarity. It makes sense because they are all reviews related to movies. However, another movie review “MR” has very low similarity to these three tasks. It’s likely because the text in “MR” is very short which makes it different from these tasks. The similarity of \( q \) from “Books” and “Video” is also very high because these two datasets share a lot of similar sentiment expressions.

As shown in Figure 6, we also show the attention distributions on a real example selected from the book review dataset. This piece of text involves two domains. The review is negative in the book domain while it is positive from the perspective of movie review. In our SA-MTL model, the “Books” review classifier from SA-MTL focus on the negative aspect of the book and evaluate the text as negative. In contrast, the “DVD” review classifier focuses on the positive part of the movie and produces a positive result. In case of DA-MTL, the model first focuses on the two domain words “book” and “movie” and judge the text is a book review because “book”
<table>
<thead>
<tr>
<th>Task</th>
<th>Single Task</th>
<th>Multiple Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>81.0</td>
<td>82.0</td>
</tr>
<tr>
<td>att-BiLSTM</td>
<td>81.8</td>
<td>82.4</td>
</tr>
<tr>
<td>Avg.</td>
<td>84.0</td>
<td>85.5</td>
</tr>
<tr>
<td>FS-MTL</td>
<td>84.0</td>
<td>85.5</td>
</tr>
<tr>
<td>SSP-MTL</td>
<td>85.5</td>
<td>87.0</td>
</tr>
<tr>
<td>PSP-MTL</td>
<td>87.0</td>
<td>89.0</td>
</tr>
<tr>
<td>ASP-MTL*</td>
<td>87.5</td>
<td>88.0</td>
</tr>
<tr>
<td>SA-MTL</td>
<td>86.8</td>
<td>88.5</td>
</tr>
<tr>
<td>DA-MTL</td>
<td>88.5</td>
<td></td>
</tr>
</tbody>
</table>

| Books     | 81.0 | 82.0 | 81.5 | 84.0 | 85.5 | 85.5 | 87.0 | 85.5 | 87.0 | 86.8 | 88.5 |
| Electronics| 81.8 | 83.0 | 82.4 | 84.8 | 86.8 | 87.3 | 89.0 | 87.5 | 89.0 |
| DVD       | 83.3 | 83.0 | 83.1 | 85.0 | 85.3 | 84.5 | 87.4 | 87.3 | 88.0 |
| Kitchen   | 80.8 | 80.3 | 80.5 | 87.0 | 86.5 | 87.5 | 87.2 | 89.3 | 89.0 |
| Apparel   | 87.5 | 86.5 | 87.0 | 86.8 | 85.3 | 85.8 | 88.7 | 87.3 | 88.8 |
| Camera    | 87.0 | 89.5 | 88.3 | 89.0 | 90.5 | 90.3 | 91.3 | 90.3 | 91.8 |
| Health    | 87.0 | 84.3 | 83.0 | 88.5 | 88.3 | 87.5 | 88.1 | 88.3 | 90.3 |
| Music     | 81.8 | 82.0 | 81.8 | 81.0 | 84.5 | 83.0 | 82.6 | 84.0 | 85.0 |
| Toys      | 81.5 | 85.0 | 85.4 | 88.3 | 87.0 | 87.8 | 88.8 | 89.3 | 89.5 |
| Video     | 83.0 | 83.5 | 83.3 | 85.0 | 87.3 | 88.0 | 85.5 | 88.5 | 89.5 |
| Baby      | 86.3 | 86.0 | 86.1 | 89.0 | 88.3 | 90.0 | 89.8 | 88.8 | 90.5 |
| Magazine  | 92.0 | 92.0 | 92.0 | 92.0 | 92.8 | 92.8 | 92.4 | 93.9 | 90.8 |
| Software  | 84.5 | 83.0 | 83.8 | 86.3 | 88.5 | 90.3 | 87.3 | 89.3 | 90.8 |
| Sports    | 86.0 | 84.8 | 85.4 | 88.3 | 88.8 | 86.8 | 86.7 | 89.8 | 89.8 |
| IMDB      | 82.5 | 83.5 | 83.0 | 82.3 | 84.0 | 84.5 | 85.8 | 87.5 | 89.8 |
| MR        | 74.8 | 76.0 | 75.4 | 71.3 | 70.8 | 69.0 | 77.3 | 73.0 | 75.5 |
| AVG       | 83.7 | 84.0 | 83.9 | 85.5(+1.6) | 86.2(+2.3) | 87.2(+3.3) | 87.6(+3.7) | 88.2(+4.3) | 88.2(+4.3) |
| # Param.  | 644 K × 16 | 645 K × 16 | – | 644 K | 16,074 K | 10,972 K | 5,490K | 668 K | 818 K |

Table 2: Performances on 16 tasks. The column of “Single Task” includes bidirectional LSTM (BiLSTM), bidirectional LSTM with attention (att-BiLSTM) and the average accuracy of the two models. The column of “Multiple Tasks” shows several multi-task models. * is from [Liu et al., 2017].

Figure 5: Similarity Matrix of Different Task’s query vector $q_k$

A good sentence representation should include its linguistic information. Therefore, we incorporate sequence labeling task (such as POS Tagging and Chunking) as an auxiliary task into the multi-task learning framework, which is trained jointly with the primary tasks (the above 16 tasks of sentiment classification). The auxiliary task shares the sentence encoding layer with the primary tasks and is connected to a private fully connected layer followed by a softmax non-linear layer to process every hidden state $h_t$ and predicts the labels.

**4.3 Exp III: Introducing Sequence Labeling as Auxiliary Task**

A good sentence representation should include its linguistic information. Therefore, we incorporate sequence labeling task (such as POS Tagging and Chunking) as an auxiliary task into the multi-task learning framework, which is trained jointly with the primary tasks (the above 16 tasks of sentiment classification). The auxiliary task shares the sentence encoding layer with the primary tasks and is connected to a private fully connected layer followed by a softmax non-linear layer to process every hidden state $h_t$ and predicts the labels.

**Dataset**

We use CoNLL 2000 [Sang and Buchholz, 2000] sequence labeling dataset for both POS Tagging and Chunking tasks. There are 8774 sentences in training data, 500 sentences in development data and 1512 sentences in test data. The average sentence length is 24 and has 17k vocabularies.

**Results**

The experiment results are shown in Table 4. We use the same hyperparameters and training procedure as the former

<table>
<thead>
<tr>
<th>SSP-MTL</th>
<th>PSP-MTL</th>
<th>SA-MTL</th>
<th>DA-MTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-task</td>
<td>83.12</td>
<td>83.25</td>
<td>84.38</td>
</tr>
<tr>
<td>Transfer</td>
<td>82.54</td>
<td>82.58</td>
<td>86.50</td>
</tr>
</tbody>
</table>
I have not read the original version of this work, but the translation lacks originality and art. A beautiful story, but the writing style lacks grace and creativity. This is the only time I have liked a movie better than the book. Do yourself a favor and skip the book. The movie is quite beautiful and moving.

(a) Attention of task “Books” in SA-MTL, Output: Negative

I have not read the original version of this work, but the translation lacks originality and art. A beautiful story, but the writing style lacks grace and creativity. This is the only time I have liked a movie better than the book. Do yourself a favor and skip the book. The movie is quite beautiful and moving.

(b) Attention of task “DVD” in SA-MTL, Output: Positive

I have not read the original version of this work, but the translation lacks originality and art. A beautiful story, but the writing style lacks grace and creativity. This is the only time I have liked a movie better than the book. Do yourself a favor and skip the book. The movie is quite beautiful and moving.

(c) Attention of auxiliary Task (Domain Classification) in DA-MTL, Output: Books

I really liked the time-span of story settings, and the mystery that was written down over 2000 years ago! Great for people who like short story mysteries, and as a lead-in to authors you might like.

(a) Model trained without Chunking task, Output: Positive

I really liked the time-span of story settings, and the mystery that was written down over 2000 years ago! Great for people who like short story mysteries, and as a lead-in to authors you might like.

(b) Model trained with Chunking task, Output: Positive

I was trying my hardest to be creep out! But like someone said, I read like a pre-teen novel.

(c) Model trained without Chunking task, Output: Positive

I was trying my hardest to be creep out! But [like] some one said, I read like a pre-teen novel.

(d) Model trained with Chunking task, Output: Negative

Table 4: Average precision of multi-task models with auxiliary tasks.

<table>
<thead>
<tr>
<th></th>
<th>SSP-MTL</th>
<th>PSP-MTL</th>
<th>SA-MTL</th>
<th>DA-MTL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Origin</td>
<td>86.2</td>
<td>86.2</td>
<td>87.59</td>
<td>88.22</td>
</tr>
<tr>
<td>+ Chunking</td>
<td>86.94</td>
<td>86.29</td>
<td>88.62</td>
<td>88.85</td>
</tr>
<tr>
<td>+ POS Tagging</td>
<td>86.83</td>
<td>86.16</td>
<td>88.52</td>
<td>89.04</td>
</tr>
</tbody>
</table>

5 Related Work

Neural networks based multi-task learning has been proven effective in many NLP problems [Collobert and Weston, 2008; Glorot et al., 2011; Liu et al., 2016; Liu et al., 2017; Ruder, 2017] in most of these models, there exists a task-dependent private layer, which plays more important role in these models, separated from the shared layer. Different from them, our model encodes all information into a shared representation layer, and uses attention mechanisms to select the task-specific information from the shared representation layer. Thus, our model can learn a better generic sentence representation, which also has a strong transferability.

Some recent work have also proposed sentence representation using attention mechanisms. [Lin et al., 2017] uses a 2-D matrix, whose each row attending on a different part of the sentence, to represent the embedding. [Vaswani et al., 2017] introduces multi-head attention to jointly attend to information from different representation subspaces at different positions. [Wang et al., 2017] introduces human reading time as attention weights to improve sentence representation. Different from these work, we use attention vector to select the task-specific information from a shared sentence representation. Thus the learned sentence representation is much more generic and easy to transfer information to new tasks.

6 Conclusion

In this paper, we propose a new information-sharing scheme for multi-task learning, which uses attention mechanism to select the task-specific information from a shared sentence encoding layer. We conduct extensive experiments on 16 different sentiment classification tasks, which demonstrates the benefits of our models. Moreover, the shared sentence encoding model can be transferred to other tasks, which can be further boosted by introducing auxiliary tasks.
Acknowledgements
We would like to thank the anonymous reviewers for their valuable comments. The research work is supported by National Natural Science Foundation of China (No. 61751201 and 61672162), and Shanghai Municipal Science and Technology Commission (No. 17JC1404100 and 16JC1420401).

References


[Socher et al., 2013] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In EMNLP, 2013.


