Hierarchical Inter-Attention Network for Document Classification with Multi-Task Learning

Bing Tian[1] , **Yong Zhang**[1] , **Jin Wang**[2] and **Chunxiao Xing**[1]
[1]RIIT, TNList, Dept. of Computer Science and Technology, Tsinghua University, Beijing, China.
[2]Computer Science Department, University of California, Los Angeles
tb17@mails.tsinghua.edu.cn, {zhangyong05, xingcx}@tsinghua.edu.cn, jinwang@cs.ucla.edu

Abstract

Document classification is an essential task in many real world applications. Existing approaches adopt both text semantics and document structure to obtain the document representation. However, these models usually require a large collection of annotated training instances, which are not always feasible, especially in low-resource settings. In this paper, we propose a multi-task learning framework to jointly train multiple related document classification tasks. We devise a hierarchical architecture to make use of the shared knowledge from all tasks to enhance the document representation of each task. We further propose an inter-attention approach to improve the task-specific modeling of documents with global information. Experimental results on 15 public datasets demonstrate the benefits of our proposed model.

1 Introduction

Document classification is a fundamental task in Natural Language Processing (NLP). The goal is to assign proper class labels to documents. It is essential in many real world applications, such as sentiment analysis [Maas *et al.*, 2011], topic labeling [Wang and Manning, 2012] and financial analysis [Luo *et al.*, 2018].

Recently deep neural networks have shown great success in learning distributed representation of texts. Given a varilength text, they represent it as a fixed-length vector using different approaches, such as Convolutional Neural Network (CNN) [Wang *et al.*, 2017], Recurrent Neural Network (RNN) [Sutskever *et al.*, 2014] and syntactic composition models [Socher *et al.*, 2012]. There are also some studies utilizing deep neural networks for document classification [Yang *et al.*, 2016]. Since deep learning models always contain huge number of parameters, they require a large volume of labeled corpus to learn a good document representation. However, in many situations, it is difficult to construct large training sets because acquiring manually labeled documents is very expensive. In this case, the performance of such models would be limited due to the lack of training data.

Multi-Task Learning (MTL) [Caruana, 1997] can jointly train several tasks with the goal of mutual benefits. It is inspired by human learning activities where people often apply the knowledge learned from previous tasks to help learn a new task [Zhang and Yang, 2017]. Similar to human learning activities, every task in MTL can make use of information from other tasks to improve the performance. It is a good solution to the problem of insufficient training data as it is able to utilize the correlation between different tasks, which can lead to better performance compared with learning them individually. For text classification, there have already been some previous studies of multi-task neural network models [Liu *et al.*, 2017; Subramanian *et al.*, 2018; Chen *et al.*, 2018]. They use an external layer to learn the common knowledge among multiple tasks and integrate such global information into each single task. In spite of their success, they have certain limitations in the task of document classification. Firstly, they are just designed for general NLP tasks, which might be suboptimal for document classification as they cannot capture the hierarchical structure of documents. Secondly, they directly use variants of RNNs to encode the document and ignore the fact that different parts could make different contributions in determining semantics of the document.

Case 1. There's really great fashion coverage that is not at all ***cookie-cutter*** and formulaic like some of the other women 's books.

Case 2. The Incredibles family members aren't ***cookie-cutter*** cartoon characters but real people who just happen to look like something out of a cartoon.

Case 3. However, Scorsese is unable to do any of this in his cape fear, giving the film a ***cookie-cutter*** early 1990s look.

Case 4. The problem with this sound is that much of it was "***cookie-cutter***" and would all sound alike.

Figure 1: An Example of Inter-Attention

In this paper, we propose **M**ulti-**T**ask **H**ierarchical **I**nter-**A**ttention Network (MT-HIA), a multi-task learning based framework to jointly train multiple related document classification tasks. In order to capture the inherent structure of documents, we adopt a hierarchical Bi-directional Long Short-Term Memory (Bi-LSTM) network [Hochreiter and Schmidhuber, 1997] in both word and sentence levels to model the sequence. We further devise an inter-attention mechanism which adopts the information from other tasks to distinguish the importance of different parts within a document so as to enhance the document representation. Unlike traditional intra-attention mechanism [Yang *et al.*, 2016] which assigns

attention weights merely from a document itself, we design a global attention layer which can learn the attention weights from other tasks. This idea comes from the observation that some semantic patterns are very rare in one task. In this case, the intra-attention mechanism would fail to recognize such patterns. But if similar semantic patterns appear in other tasks, we can recognize them and make the correct predication with the help of multi-task learning techniques. An example is shown in Figure 1. Case 1 to 4 are Amazon product reviews coming from different domains. For the *Magazine* review *"There's really great fashion coverage that is not at all cookie-cutter and formulaic like some of the other women's books. "* in Case 1, it is essential to understand the informative word *cookie-cutter* for sentiment classification. However, it is hard for intra-attention based model to capture this kind of expression since it is rare in this corpus. Nevertheless, as this kind of expression is common in datasets related to movies such as *Video* and *IMDB* as shown in Case 2 to 4, our inter-attention based model could take advantage of inter-attention layer to recognize it and finally obtain the right prediction. Moreover, we also propose a global shared layer with another LSTM network to utilize the global information among all tasks to enhance each task-specific representation. We conduct a comprehensive evaluation on 15 real world datasets. Experimental results show that our model outperforms previous approaches by an obvious margin on most datasets.

Contributions of this paper are summarized as following:

- We propose MT-HIA, a hierarchical model with multi-task learning for document classification. Compared with previous studies, our proposed model can take advantage of both inherent document structure and the common knowledge from multi-tasks.
- We introduce a hierarchical inter-attention approach to enhance the document representation of each single task. Moreover, we also devise a global sharing mechanism to utilize the knowledge from multiple tasks.
- We conduct extensive experiments on 15 real world datasets. The results demonstrate the effectiveness of our proposed methods.

2 Related Work

2.1 Document Classification

Recently neural network models have been widely applied in the task of document classification. Wang et al. [2017] integrated information from knowledge bases into CNN models to improve the performance. Sutskever et al. [2014] and Liu et al. [2015] utilized variants of RNN to capture the sequence information in the texts. Socher et al. [2012] proposed Recursive Neural Network by leveraging the syntax information. Zhang et al. [2015] adopted a CNN model in character level for text classification. Yang et al. [2016] proposed HAN, a hierarchical model to learn the document structure and enrich the representation.

2.2 Sentiment Analysis

Sentiment analysis is an important task in text classification which focuses on inferring the sentiment polarity of the text.

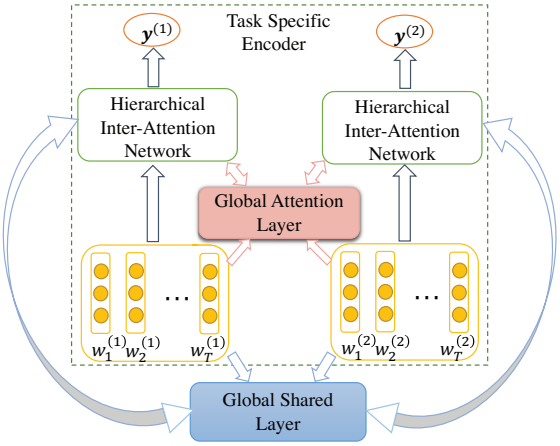

Figure 2: Overall Architecture of MT-HIA

Recently neural network has been widely applied in the task of sentiment analysis. Tang et al. [2015] proposed a neural network based framework which can make use of extra information, i.e., product and user to improve sentiment classification. Johnson et al. [2016] explored region embeddings via one-hot LSTM to improve the performance. McCann et al. [2017] introduced an approach for transferring knowledge from an encoder pretrained on machine translation to sentiment analysis.

2.3 Multi-Task Learning

Multi-task learning [Caruana, 1997] is an approach to learn multiple related tasks simultaneously, aiming at yielding performance gains by leveraging potential correlations and common features among related tasks. It has been widely adopted in many applications, such as speech recognition [Rao *et al.*, 2018], computer vision [Zhang *et al.*, 2012] and natural language processing [Sutskever *et al.*, 2014]. Multi-task learning techniques have already been widely adopted in NLP tasks. For example, Collobert et al. [2008] utilized the word inputs across multiple tasks to improve POS tagging. Firat et al. [2016] adopted multi-task learning in the task of machine translation. Liu et al. [2019] proposed a graph based model to capture the relations between different tasks.

Recently there have been some studies using multi-task neural network for text classification. Liu et al. [2016a; 2016b] proposed a multi-task RNN model for text classification with 3 mechanisms for sharing information. Liu et al. [2017] further employed the adversarial training mechanism to reduce the noise from different tasks. Chen et al. [2018] improved the multi-task RNN models by extracting meta-knowledge from different tasks. These methods just focus on general text classification tasks and cannot make use of the hierarchical structure of documents.

3 Methodology

Figure 2 displays the overall architecture of our proposed model. Here, we take the case of two tasks as an example. It consists of two components: the task specific encoder and the global shared layer. The task specific encoder is a hierarchical network which involves multi-task driven inter-attention

mechanism from multiple granularities, i.e. both in word-level and sentence-level, to enhance the representation of documents. The global shared layer enables multi-task learning: it adopts a global shared memory to learn the common knowledge from all tasks. Next we will first introduce the task specific encoder and then the global shared layer in details.

3.1 Task Specific Encoder: Hierarchical Inter-Attention Network

The task specific encoder is a hierarchical attention network shown in Figure 3. It consists of three components: a Bi-LSTM layer, a multi-task driven inter-attention layer and an output layer. The Bi-LSTM and attention layers are applied in both word and sentence levels. Next we will describe the details of different components.

Bi-LSTM Based Sequence Encoder

We adopt LSTM as the basic building block for sequence encoder. Given input $\mathbf{x}_t$ and previous hidden state $\mathbf{h}_{t-1}$, the current hidden state $\mathbf{h}_t$ can be updated by:

$$\mathbf{h}_t = LSTM(\mathbf{h}_{t-1}, \mathbf{x}_t, \theta) \tag{1}$$

Here we choose the LSTM architecture devised in [Józefowicz *et al.*, 2015] as our encoder and use the function $LSTM(\cdot,\cdot,\cdot)$ as a shorthand for the encoding process. And θ refers to all parameters of LSTM. To better utilize the contextual information, we use Bi-LSTM to learn the hidden states.

Based on Bi-LSTM, we construct the hierarchical model as following. Given a sentence i with words $w_{it}, t \in [0, T]$ (T is the number of words in the sentence), we first embed the words to low-dimensional vectors through an embedding matrix $\mathbf{W}_e$, i.e. $\mathbf{x}_{it} = \mathbf{W}_e w_{it}$. And then we construct the word-level encoder by feeding $\mathbf{x}_{it}$ into a Bi-LSTM network to obtain the hidden state of each word. The final representation of the t^{th} word in i^{th} sentence is the concatenation of output in both directions:

$$\mathbf{h}_{it} = \begin{bmatrix} \overrightarrow{h_{it}} \\ \overleftarrow{h_{it}} \end{bmatrix} = \begin{bmatrix} \overrightarrow{LSTM}(\overrightarrow{h}_{i(t-1)}, \mathbf{x}_{it}, \theta_w) \\ \overleftarrow{LSTM}(\overleftarrow{h}_{i(t-1)}, \mathbf{x}_{it}, \theta_w) \end{bmatrix}, t \in [1, T] \tag{2}$$

In sentence level, we feed the output of word-level encoders $[\vec{s}_1, \vec{s}_2, ..., \vec{s}_L]$ (L is the number of sentences in the document) into the Bi-LSTM and concatenate $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$, $i \in [1, L]$ to get a representation of i^{th} sentence $\mathbf{h}_i$ in a similar way.

Multi-task Driven Inter-Attention

As different words and sentences make different contributions to the composition of a document, it is necessary to assign different importance to them. To reach this goal, we design an inter-attention mechanism driven by multi-task learning to select the informative words and sentences. The intuition is that as external knowledge, information from the global shared component can provide common and task-invariant knowledge, which can help disambiguate the semantics of words and sentences in the document of each single task. Specifically, we utilize a multi-task based Multi-Layer Perceptron (MLP), which is a global layer shared by all tasks, to help compute attention weights. And we then involve information from this layer into both word-level and sentence-level encoder to help assign weights for each word and sentence.

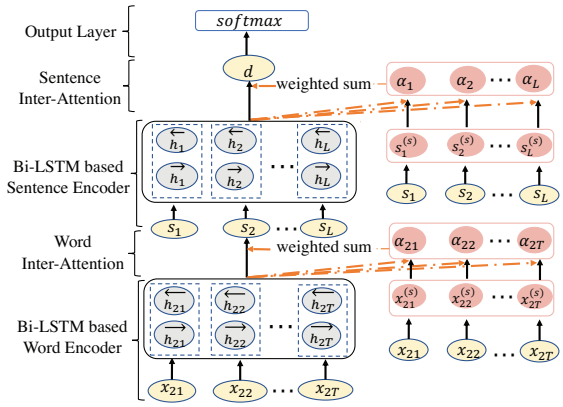

Figure 3: Hierarchical Inter-Attention Network

$$\mathbf{x}_{it}^{(s)} = \textit{Word\_Shared\_MLP}(\mathbf{x}_{it}), t \in [1, T] \tag{3}$$

$$\mathbf{u}_{it}^{(s)} = \tanh(\mathbf{W}_w \mathbf{x}_{it}^{(s)} + \mathbf{b}_w), t \in [1, T] \tag{4}$$

$$\alpha_{it} = \frac{\exp((\mathbf{u}_{it}^{(s)})^T \mathbf{u}_w)}{\sum_t \exp((\mathbf{u}_{it}^{(s)})^T \mathbf{u}_w)}, t \in [1, T] \tag{5}$$

$$\mathbf{s}_i = \sum_t \alpha_{it} \mathbf{h}_{it}, t \in [1, T] \tag{6}$$

That is, we first feed the word vector $\mathbf{x}_{it}$ into a Word-level Shared MLP layer (*Word_Shared_MLP*) which consists of a MLP to get the shared representation $\mathbf{x}_{it}^{(s)}$ for computing attention weights in Eq. 3. Then we get a hidden representation $\mathbf{u}_{it}^{(s)}$ of $\mathbf{x}_{it}^{(s)}$ containing common and task-invariant knowledge in Eq. 4. We measure the importance of t^{th} word in i^{th} sentence by an attention weight α_{it} which can be computed as the inner product of $\mathbf{u}_{it}^{(s)}$ and a word level weight vector $\mathbf{u}_w$ followed by a *softmax* layer in Eq. 5. The word level weight vector $\mathbf{u}_w$ is randomly initialized and jointly learned during the training process. Finally, we obtain the sentence representation by computing the weighted sum of the word hidden states based on the weights in Eq. 6.

Similarly, in order to obtain a better document representation with selected information, we also adopt multi-task driven inter-attention mechanism and propose a Sentence-level Shared MLP layer (*Sentence_Shared_MLP*). We first feed the sentence vector $\mathbf{s}_i$ into the *Sentence_Shared_MLP* to get the representation $\mathbf{s}_i^{(s)}$. Then we can correspondingly get the attention weights and the document representation $\mathbf{d}$ following the same route with the word-level encoder.

Output Layer

Since the document vector $\mathbf{d}$ is the high level representation of the document, it can be then fed into a *softmax* layer for classification:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_y \mathbf{d} + \mathbf{b}_y) \tag{7}$$

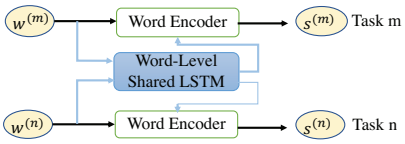

Figure 4: Word Level Shared Layer of Multi-Task Learning

3.2 Global Shared Layer for Multi-Task Learning

Based on the task-specific model, we then describe our multi-task learning based architecture.

The task of document level classification is to assign a label Y to a document X. Supposing there are K related tasks, we denote the corpus of the k^{th} task with N_k samples as:

$$D_k = \{(X_i^{(k)}, Y_i^{(k)})\}_{i=1}^{N_k}, \tag{8}$$

where $X_i^{(k)}$ and $Y_i^{(k)}$ are i^{th} sample and its label in the k^{th} task.

The key factor of multi-task learning is the sharing scheme among different tasks. Considering the hierarchical structure of documents, we design the global shared layer to determine the common features in both word-level and sentence-level. Figure 4 is the illustration of word-level shared layer. Specifically, for each word $\mathbf{x}_t^{(k)}$ at time step t in task k, we first compute its shared representation $\mathbf{h}_t^{(s)}$ by an LSTM network:

$$\mathbf{h}_t^{(s)} = \textit{word\_shared\_LSTM}(\mathbf{x}_t^{(k)}, \mathbf{h}_{t-1}^{(s)}, \theta^{(s)}) \tag{9}$$

And then we utilize the gating mechanism to control the portion of information flows from the global shared layer to each task:

$$\tilde{\mathbf{c}}_t^{(k)} = \tanh(\mathbf{W}_{\mathbf{c}}^{(k)} \begin{bmatrix} \mathbf{x}_t^{(k)} \\ \mathbf{h}_{t-1}^{(k)} \end{bmatrix} + \mathbf{g}^{(k)} \odot (\mathbf{U}_c^{(k)} \mathbf{h}_t^{(s)} + \mathbf{b}_c^{(k)})) \tag{10}$$

$$\mathbf{g}^{(k)} = \sigma(\mathbf{W}_g^{(k)} \begin{bmatrix} \mathbf{x}_t^{(k)} \\ \mathbf{h}_t^{(s)} \end{bmatrix} + \mathbf{b}_g^{(k)}) \tag{11}$$

$$\mathbf{c}_t^{(k)} = \tilde{c}_t^{(k)} \odot \mathbf{i}_t^{(k)} + \mathbf{c}_{t-1}^{(k)} \odot \mathbf{f}_t^{(k)} \tag{12}$$

where $\mathbf{g}^{(k)}$ controls the portion of information flows from the word-level global shared layer to task k, based on the correlation strength between $\mathbf{x}_t^{(k)}$ and $\mathbf{h}_t^{(s)}$ at the current time step.

Similarly, in sentence-level encoder, there also exists a global shared LSTM layer (*sentence_shared_LSTM*) which captures the shared information for all the tasks and can be conveyed to each task. In this way, the hidden states and memory cells of each single task can benefit from extra information from all other tasks.

3.3 Training

The objective of training process is to minimize the cross-entropy of the predicted and true distributions for all tasks. For a single task k, we adopt cross-entropy to compute its loss $L(\hat{\mathbf{y}^{(k)}}, \mathbf{y}^{(k)})$ as shown in Eq. 14 where N_k refers to the number of training samples and C is the class number. $y_{ij}^{(k)}$ is the ground-truth label and $\hat{y}_{ij}^{(k)}$ is the predicted probability.

$$L(\Theta) = \sum_{k=1}^{K} \lambda_k L(\hat{\mathbf{y}^{(k)}}, \mathbf{y}^{(k)}) \tag{13}$$

$$L(\hat{\mathbf{y}^{(k)}}, \mathbf{y}^{(k)}) = -\sum_{i=1}^{N_k} \sum_{j=1}^{C} y_{ij}^{(k)} log(\hat{y}_{ij}^{(k)}) \tag{14}$$

The overall loss of all tasks is shown in Eq. 13, where λ_k is the weight for the k^{th} task and Θ is the set of all trainable parameters.

For multi-task learning, the labeled data for training each task can come from completely different datasets. Following the previous study [Collobert and Weston, 2008], the training process is conducted in a stochastic manner by looping over the tasks:

1. Select a random task.
2. Select a mini-batch of examples from this task.
3. Update the parameters for this task and the global shared layer with respect to this mini-batch.
4. Go to 1.

After the joint learning phase, we can use a fine tuning strategy to further optimize the performance for each task.

4 Evaluation

In this section, we introduce the empirical results of our MT-HIA model. We use the accuracy of classification as the evaluation metrics.

4.1 Experiment Setup

Datasets

We evaluate the effectiveness of our model on 15 document-level corpus. The first 14 datasets are Amazon product reviews coming from different domains such as Books, Music, Baby, etc. These datasets are collected based on the dataset [1]provided by Blitzer et al. [2007]. The last IMDB dataset contains movie reviews with binary classes [Maas *et al.*, 2011].

All these datasets are document-level in which each review consists of several sentences. And the goal is to assign the positive or negative label to each document. Following previous studies, we randomly split these datasets into training sets, development sets and testing sets with the proportion of 70%, 10% and 20% respectively. The detailed statistics about these datasets are displayed in Table 1. To enable multi-task learning, we jointly train all 15 datasets simultaneously.

Competitor Methods

We compare our model with several state-of-the-art methods, including single-task learning based models and multi-task learning based models. For single-task learning models, we compare with four methods:

- **LSTM:** the standard LSTM [Hochreiter and Schmidhuber, 1997].
- **HyperLSTM:** a model which uses a small network to generate weights for a larger network [Ha *et al.*, 2016].

[1]https://www.cs.jhu.edu/~mdredze/datasets/sentiment/

Datasets	Train Size	Dev. Size	Test Size	Class	Avg. Length	Voc Size
Books	1400	200	400	2	159	62K
Elec	1398	200	400	2	101	30K
DVD	1400	200	400	2	173	69K
Kitchen	1400	200	400	2	89	28K
Apparel	1400	200	400	2	57	21K
Camera	1397	200	400	2	130	26K
Health	1400	200	400	2	81	26K
Music	1400	200	400	2	136	60K
Toys	1400	200	400	2	90	28K
Video	1400	200	400	2	156	57K
Baby	1300	200	400	2	104	26K
Mag	1370	200	400	2	117	30K
Soft	1315	200	400	2	129	26K
Sports	1400	200	400	2	94	30K
IMDB	1400	200	400	2	269	44K

Table 1: Statistics of 15 Datasets for Document Classification

- **MetaLSTM:** a model which uses a meta-network to capture the meta-knowledge of semantic composition [Chen *et al.*, 2018].
- **HAN:** a hierarchical attention network for single task document classification [Yang *et al.*, 2016].

For multi-task learning, we compare our model with five models.

- **ASP-MTL:** an adversarial MTL framework, alleviating the shared and private latent feature spaces from interfering with each other [Liu *et al.*, 2017].
- **Meta-MTL:** a function-level sharing scheme for MTL with a shared meta-network proposed in [Chen *et al.*, 2018].
- **DA-MTL:** a MTL framework with a shared sentence encoding layer and a dynamic task-attentive mechanism [Zheng *et al.*, 2018].
- **CG-MTL:** a graph based MTL framework, which is the up-to-date method [Liu *et al.*, 2019].
- **MT-HA:** a method proposed by ourselves. It is implemented by just replacing the inter-attention with intra-attention mechanism.

Although there are some other related approaches, such as single-task ones [Tang *et al.*, 2015; Zhang *et al.*, 2015] and multi-task ones [Liu *et al.*, 2016a; Liu *et al.*, 2016b], previous studies have shown that they cannot outperform the above selected methods. Therefore, we do not report the results of them here due to the space limitation.

Hyper-Parameter Settings

We initialize the word embeddings with pre-trained GloVe vectors [Pennington *et al.*, 2014]. For out of box words, we randomly initialize their embeddings from uniform distribution in (-0.01, 0.01). The dimension of word embeddings is 200. And the models are trained with backpropagation using Adam optimizer with mini-batch size 30. The detailed settings of hyper-parameters are shown in Table 2.

Word embedding size	d = 200
Size of word-level basic-LSTM layer	h_w = 50
Size of word-level shared-LSTM layer	$h_w^{(s)}$ = 50
Size of sentence-level basic-LSTM layer	h_s = 50
Size of sentence-level shared-LSTM layer	$h_s^{(s)}$ = 50
Initial learning rate	0.001
Regularization	$1E-5$

Table 2: Hyper-Parameter Settings

4.2 Analysis of Results

Table 3 shows the classification accuracies on 15 datasets. The column "Single Task" refers to the results of 4 single task models. The column of "Multiple Tasks" shows the results achieved by six multi-task models including ours. Besides MT-HIA, we also report the results obtained by MT-HA. It stands for the multi-task hierarchical intra-attention network which assigns the attention weights merely from the document itself like attention mechanism in HAN. From the results, we have the following observations. Firstly, the overall performance of multi-task learning based models is much better than single-task ones. This result is consistent with previous studies [Chen *et al.*, 2018] and demonstrates the benefits of multi-task learning techniques. Among them our MT-HIA obtains the best performance. Specifically, it outperforms HAN, the most competitive single-task model by 4.3% on average. Secondly, our model also achieves a better performance than competitors in most cases for multi-task learning. For example, our method shows an average improvement of accuracy 1.8% to CG-MTL, the up-to-date multi-task model and 0.4% to DA-MTL, the most competitive multi-task model. Lastly, we can see that MT-HIA outperforms MT-HA in 13 out of 15 datasets. This demonstrates the advantage of our inter-attention mechanism over the intra-attention one.

Next we look at the specific performance gain of MT-HIA against single-task methods. The column "Single Avg" shows the average accuracy of 4 single-task based models. From this table, we can see that the performance of most tasks can be improved with a large margin with the help of MTL. And our MT-HIA is able to achieve the state-of-the-art performances on each single task in most cases, indicating the effectiveness and robustness of our model. Among all single-task learning methods, HAN performs best since it takes hierarchical structure of documents into consideration and proposes a hierarchical attention architecture. However, as the attention weights of HAN only comes from the document itself, it might be difficult for HAN to assign proper weights for infrequent semantic patterns. As our proposed model utilizes multi-task learning, each task can extract both common and task-invariant knowledge from global shared layers. And we adopts inter-attention mechanism, which can learn the attention weights from the global sharing information contributed by multiple tasks. Therefore, our MT-HIA is more powerful in discovering informative sentences and words in a document and thus achieves better results.

Finally, we make detailed analysis on the results. Compared with other MTL methods, our MT-HIA model can better capture the hierarchical structure of documents. Other

Task	Single Task				Single Avg	Multiple Tasks					
	LSTM	Hyper LSTM	Meta LSTM	HAN		ASP-MTL	Meta-MTL	DA-MTL	CG-MTL	MT-HA(ours)	MT-HIA (ours)
Books	79.5	78.3	83	85	81.5	87	87.5	88.5	86.7	87.5	**89.2**
Electronics	80.5	80.7	82.3	81.7	81.3	89	**89.5**	89	88.5	88	87.5
DVD	81.7	80.3	82.3	83.3	81.9	87.4	88	88	86.5	88.3	**89.2**
Kitchen	78	80	83.3	85.8	81.8	87.2	91.3	89	87.7	91.6	**91.7**
Apparel	83.2	85.8	86.5	85	85.1	88.7	87	**88.8**	87	86.7	87.5
Camera	85.2	88.3	88.3	90	88	91.3	89.7	91.8	89.5	91.1	**92.2**
Health	84.5	84	86.3	85.9	85.2	88.1	90.3	90.3	89.5	88.9	**91.1**
Music	76.7	78.5	80	82.5	79.4	82.6	86.3	85	85.2	85.8	**88.3**
Toys	83.2	83.7	84.3	85.4	84.2	88.8	88.5	**89.5**	89	88.3	85.4
Video	81.5	83.7	84.3	85.6	83.8	85.5	88.3	89.5	87	88.9	**90**
Baby	84.7	85.5	84	86.7	85.2	89.8	88	90.5	89.2	88.4	**91.7**
Magazines	89.2	91.3	92.3	92	91.2	92.4	91	92	92	92.2	**95.6**
Software	84.7	86.5	88.3	86.7	86.6	87.3	88.5	90.8	89.7	90	**91.7**
Sports	81.7	82	82.5	84.2	82.6	86.7	86.7	**89.8**	87.7	87.5	87.5
IMDB	81.7	77	83.5	84.2	81.6	85.8	88	89.8	87	88.9	**90**
AVG	82.4	83.0	84.7	85.6	83.9	87.8	88.6	89.5	88.1	88.8	**89.9**

Table 3: Accuracy of Our Models on 15 Datasets against Baselines

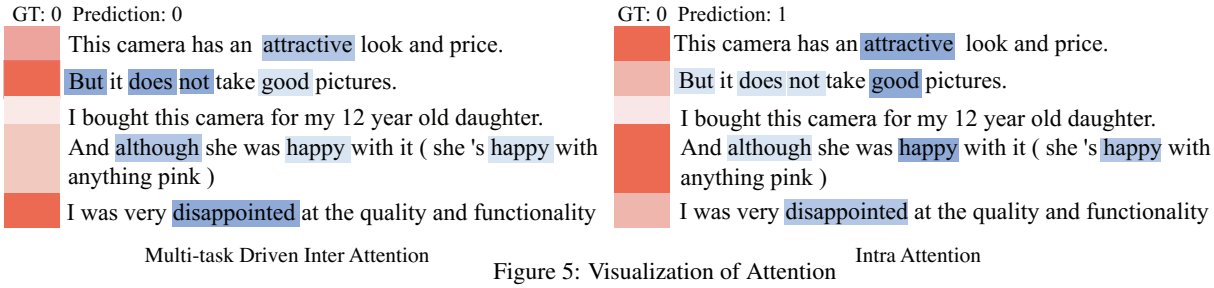

Figure 5: Visualization of Attention

latest multi-task learning methods, such as Meta-MTL and DA-MTL focus on how to make good use of the shared information from text corpus. However, they all ignore the hierarchical structure of documents and will definitely fail to capture sufficient signals for document classification. Thus their performance are not as good as ours. Among all baseline methods, DA-MTL performs the best. The reason is that it proposes a new scheme of information sharing for MTL: all tasks share the same sentence representation and each task can select the task-specific information from the shared representation with attention mechanism. Compared with DA-MTL, our model can also reach the same goal with the help of inter-attention mechanism. As our method can also capture the hierarchical structure, it can clearly beat DA-MTL.

4.3 Case Study

In order to illustrate that our model is capable of better discovering informative sentences and words in a document, we visualize the weights of sentences and words in an hierarchical manner in Figure 5. In this example, we compare MT-HIA with inter attention mechanism with MT-HA that only assigns the attention weights merely from the document itself.

The red color refers to the attention weights over sentences, blue denotes attention weights over words and the depth of color represents the size of the weights. From Figure 5, we can see that for the negative review, both models can select the words carrying strong sentiment like *"attractive"*, *"disappointed"*. However the intra-attention mechanism fails to capture the semantic patten *"although ... I was..."* since similar patterns are rare in this corpus. Therefore, it puts more attention on the fourth sentence and gives a wrong prediction. Nevertheless, as our MT-HIA is able to take advantage of common knowledge from other tasks, it will be easier to capture this pattern once it appears in other tasks. As a result, MT-HIA successfully captures this kind of pattern and chooses to ignore the fourth sentence and finally gets the right prediction.

5 Conclusion

In this paper, we propose the Multi-Task Hierarchical Inter-Attention Network model for document classification. We improve the task-specific document representation by proposing an inter-attention mechanism. We further devise a global shared mechanism to smartly utilize the knowledge shared by multiple tasks. Experimental results on 15 real world datasets show that our proposed model outperforms state-of-the-art methods by a substantial margin. For the future work, we would like to explore our model on more different types of NLP tasks.

Acknowledgements

This work was supported by NSFC(91646202), National Key R&D Program of China(SQ2018YFB140235).

References

[Blitzer *et al.*, 2007] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *ACL*, 2007.

[Caruana, 1997] Rich Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997.

[Chen *et al.*, 2018] Junkun Chen, Xipeng Qiu, Pengfei Liu, and Xuanjing Huang. Meta multi-task learning for sequence modeling. In *AAAI*, 2018.

[Collobert and Weston, 2008] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, pages 160–167, 2008.

[Firat *et al.*, 2016] Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *NAACL-HLT*, pages 866–875, 2016.

[Ha *et al.*, 2016] David Ha, Andrew M. Dai, and Quoc V. Le. Hypernetworks. *CoRR*, abs/1609.09106, 2016.

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

[Johnson and Zhang, 2016] Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using LSTM for region embeddings. In *ICML*, pages 526–534, 2016.

[Józefowicz *et al.*, 2015] Rafal Józefowicz, Wojciech Zaremba, and Ilya Sutskever. An empirical exploration of recurrent network architectures. In *ICML*, pages 2342–2350, 2015.

[Liu *et al.*, 2015] Pengfei Liu, Xipeng Qiu, Xinchi Chen, Shiyu Wu, and Xuanjing Huang. Multi-timescale long short-term memory neural network for modelling sentences and documents. In *EMNLP*, pages 2326–2335, 2015.

[Liu *et al.*, 2016a] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Deep multi-task learning with shared memory for text classification. In *EMNLP*, pages 118–127, 2016.

[Liu *et al.*, 2016b] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Recurrent neural network for text classification with multi-task learning. In *IJCAI*, pages 2873–2879, 2016.

[Liu *et al.*, 2017] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Adversarial multi-task learning for text classification. In *ACL*, pages 1–10, 2017.

[Liu *et al.*, 2019] Pengfei Liu, Jie Fu, Yue Dong, Xipeng Qiu, and Jackie Chi Kit Cheung. Multi-task learning over graph structures. In *AAAI*, 2019.

[Luo *et al.*, 2018] Ling Luo, Xiang Ao, Feiyang Pan, Jin Wang, Tong Zhao, Ningzi Yu, and Qing He. Beyond polarity: Interpretable financial sentiment analysis with hierarchical query-driven attention. In *IJCAI*, pages 4244–4250, 2018.

[Maas *et al.*, 2011] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150, 2011.

[McCann *et al.*, 2017] Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors. In *NIPS*, pages 6297–6308, 2017.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.

[Rao *et al.*, 2018] Jinfeng Rao, Ferhan Türe, and Jimmy Lin. Multi-task learning with neural networks for voice query understanding on an entertainment platform. In *KDD*, pages 636–645, 2018.

[Socher *et al.*, 2012] Richard Socher, Brody Huval, Christopher D. Manning, and Andrew Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*, pages 1201–1211, 2012.

[Subramanian *et al.*, 2018] Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J. Pal. Learning general purpose distributed sentence representations via large scale multi-task learning. *ICLR*, 2018.

[Sutskever *et al.*, 2014] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112, 2014.

[Tang *et al.*, 2015] Duyu Tang, Bing Qin, and Ting Liu. Learning semantic representations of users and products for document level sentiment classification. In *ACL*, pages 1014–1023, 2015.

[Wang and Manning, 2012] Sida I. Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, pages 90–94, 2012.

[Wang *et al.*, 2017] Jin Wang, Zhongyuan Wang, Dawei Zhang, and Jun Yan. Combining knowledge with deep convolutional neural networks for short text classification. In *IJCAI*, pages 2915–2921, 2017.

[Yang *et al.*, 2016] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. Hierarchical attention networks for document classification. In *NAACL-HLT*, pages 1480–1489, 2016.

[Zhang and Yang, 2017] Yu Zhang and Qiang Yang. A survey on multi-task learning. *CoRR*, abs/1707.08114, 2017.

[Zhang *et al.*, 2012] Tianzhu Zhang, Bernard Ghanem, Si Liu, and Narendra Ahuja. Robust visual tracking via multi-task sparse learning. In *CVPR*, pages 2042–2049, 2012.

[Zhang *et al.*, 2015] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, pages 649–657, 2015.

[Zheng *et al.*, 2018] Renjie Zheng, Junkun Chen, and Xipeng Qiu. Same representation, different attentions: Shareable sentence representation learning from multiple tasks. In *IJCAI*, pages 4616–4622, 2018.