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

Bing Tian\textsuperscript{1} , Yong Zhang\textsuperscript{1} , Jin Wang\textsuperscript{2} and Chunxiao Xing\textsuperscript{1}

\textsuperscript{1}RIIT, TNList, Dept. of Computer Science and Technology, Tsinghua University, Beijing, China.
\textsuperscript{2}Computer Science Department, University of California, Los Angeles

\texttt{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 variable length 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 a large 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.

In this paper, we propose Multi-Task Hierarchical Inter-Attention 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 adapts 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. 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 \( x_i \) and previous hidden state \( h_{t-1} \), the current hidden state \( h_t \) can be updated by:

\[
h_t = LSTM(h_{t-1}, x_t, \theta)
\]

(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 \( \theta \) 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 \( W_e \), i.e. \( x_{it} = W_e w_{it} \). And then we construct the word-level encoder by feeding \( x_{it} \) 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:

\[
h_{it} = \left[ \tilde{h}_{it}^\top \right] = \begin{bmatrix} LSTM(h_{i(t-1)}, x_t, \theta_w) \\ LSTM(h_{it-1}, x_t, \theta_w) \end{bmatrix}, t \in [1, T]
\]

(2)

In sentence level, we feed the output of word-level encoders \( [s_1, s_2, ..., s_L] \) (\( L \) is the number of sentences in the document) into the Bi-LSTM and concatenate \( \tilde{h}_i \) and \( \tilde{h}_i^\top \) to get a representation of \( t \)th sentence \( h_i \) 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.

\[
x_{it}^{(s)} = Word_{Shared\_MLP}(x_{it}), t \in [1, T]
\]

(3)

\[
u_{it}^{(s)} = \tanh(W_w x_{it}^{(s)} + b_w), t \in [1, T]
\]

(4)

\[
\alpha_{it} = \exp((u_{it}^{(s)})^T u_{it}), t \in [1, T]
\]

(5)

\[
s_t = \sum_t \alpha_{it} h_{it}, t \in [1, T]
\]

(6)

That is, we first feed the word vector \( x_{it} \) into a Word-level Shared MLP layer (\textit{Word\_Shared\_MLP}) which consists of a MLP to get the shared representation \( x_{it}^{(s)} \) for computing attention weights in Eq. 3. Then we get a hidden representation \( u_{it}^{(s)} \) of \( x_{it}^{(s)} \) 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 \( \alpha_{it} \) which can be computed as the inner product of \( u_{it}^{(s)} \) and a word level weight vector \( u_{w} \) followed by a \textit{softmax} layer in Eq. 5. The word level weight vector \( u_{w} \) 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 (\textit{Sentence\_Shared\_MLP}). We first feed the sentence vector \( s_t \) into the \textit{Sentence\_Shared\_MLP} to get the representation \( s_t^{(s)} \). Then we can correspondingly get the attention weights and the document representation \( d \) following the same route with the word-level encoder.

Output Layer

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

\[
y = \text{softmax}(W_\theta d + b_\theta)
\]

(7)
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:

$$D_k = \{(X^{(k)}_i, Y^{(k)}_i)\}_{i=1}^{N_k},$$

where $X^{(k)}_i$ and $Y^{(k)}_i$ 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 $x^{(k)}_t$ at time step $t$ in task $k$, we first compute its shared representation $h^{(s)}_t$ by an LSTM network:

$$h_t^{(s)} = \text{word} \_\text{shared} \_\text{LSTM}(x^{(k)}_t, h_{t-1}^{(s)}, \theta^{(s)})$$

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

$$\tilde{c}_t^{(k)} = \tanh(W_c^{(k)} \left[ x^{(k)}_t, h_{t-1}^{(s)} \right]) + g^{(k)} \odot (U_c^{(k)} h_t^{(s)} + b_c^{(k)})$$

$$g^{(k)} = \sigma(W_g^{(k)} \left[ x^{(k)}_t, h_{t}^{(s)} \right] + b_g^{(k)})$$

$$c_t^{(k)} = \tilde{c}_t^{(k)} \odot i_t^{(k)} + c_{t-1}^{(k)} \odot f_t^{(k)}$$

where $g^{(k)}$ controls the portion of information flows from the word-level global shared layer to task $k$, based on the correlation strength between $x^{(k)}_t$ and $h_{t}^{(s)}$ 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 $L(y^{(k)}, \hat{y}^{(k)})$ 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.

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

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

The overall loss of all tasks is shown in Eq. 13, where $\lambda_k$ is the weight for the $k^{th}$ task and $\Theta$ 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 HyperLIA 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\footnote{https://www.cs.jhu.edu/~mdredze/datasets/sentiment/} 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].
Table 1: Statistics of 15 Datasets for Document Classification

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Train Size</th>
<th>Dev. Size</th>
<th>Test Size</th>
<th>Class Avg. Length</th>
<th>Voc Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Books</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>159</td>
</tr>
<tr>
<td>Elec</td>
<td>1398</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>101</td>
</tr>
<tr>
<td>DVD</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>173</td>
</tr>
<tr>
<td>Kitchen</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>89</td>
</tr>
<tr>
<td>Apparel</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>57</td>
</tr>
<tr>
<td>Camera</td>
<td>1397</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>130</td>
</tr>
<tr>
<td>Health</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>81</td>
</tr>
<tr>
<td>Music</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>136</td>
</tr>
<tr>
<td>Toys</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>Video</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>156</td>
</tr>
<tr>
<td>Baby</td>
<td>1300</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>104</td>
</tr>
<tr>
<td>Mag</td>
<td>1370</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>117</td>
</tr>
<tr>
<td>Soft</td>
<td>1315</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>129</td>
</tr>
<tr>
<td>Sports</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>29</td>
</tr>
<tr>
<td>IMDB</td>
<td>1400</td>
<td>200</td>
<td>400</td>
<td>2</td>
<td>269</td>
</tr>
</tbody>
</table>

Table 2: Hyper-Parameter Settings

<table>
<thead>
<tr>
<th>Word embedding size</th>
<th>d = 200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size of word-level basic-LSTM layer</td>
<td>$h_w = 50$</td>
</tr>
<tr>
<td>Size of word-level shared-LSTM layer</td>
<td>$h_{iw}^{(s)} = 50$</td>
</tr>
<tr>
<td>Size of sentence-level basic-LSTM layer</td>
<td>$h_s = 50$</td>
</tr>
<tr>
<td>Size of sentence-level shared-LSTM layer</td>
<td>$h_{is}^{(s)} = 50$</td>
</tr>
<tr>
<td>Initial learning rate</td>
<td>0.001</td>
</tr>
<tr>
<td>Regularization</td>
<td>$1E^{-5}$</td>
</tr>
</tbody>
</table>

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.
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”, “dis-appointed”. However the intra-attention mechanism fails to capture the semantic pattern “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).

<table>
<thead>
<tr>
<th>Task</th>
<th>LSTM</th>
<th>Hyper LSTM</th>
<th>Meta LSTM</th>
<th>HAN</th>
<th>Avg</th>
<th>ASP-MTL</th>
<th>Meta-MTL</th>
<th>DA-MTL</th>
<th>CG-MTL</th>
<th>MT-HA(ours)</th>
<th>MT-HIA (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Books</td>
<td>79.5</td>
<td>78.3</td>
<td>83</td>
<td>85</td>
<td>81.5</td>
<td>87</td>
<td>87.5</td>
<td>88.5</td>
<td>86.7</td>
<td>87.5</td>
<td>89.2</td>
</tr>
<tr>
<td>Electronics</td>
<td>80.5</td>
<td>80.7</td>
<td>82.3</td>
<td>81.7</td>
<td>81.3</td>
<td>89</td>
<td>89.5</td>
<td>89</td>
<td>88.5</td>
<td>88</td>
<td>87.5</td>
</tr>
<tr>
<td>DVD</td>
<td>81.7</td>
<td>80.3</td>
<td>82.3</td>
<td>83.3</td>
<td>81.9</td>
<td>87.4</td>
<td>88</td>
<td>88</td>
<td>86.5</td>
<td>88.3</td>
<td>89.2</td>
</tr>
<tr>
<td>Kitchen</td>
<td>78</td>
<td>80</td>
<td>83.3</td>
<td>85.8</td>
<td>81.8</td>
<td>87.2</td>
<td>91.3</td>
<td>87.9</td>
<td>87.7</td>
<td>91.6</td>
<td>91.7</td>
</tr>
<tr>
<td>Apparel</td>
<td>83.2</td>
<td>85.8</td>
<td>86.5</td>
<td>85</td>
<td>85.1</td>
<td>88.7</td>
<td>87</td>
<td>88.8</td>
<td>87</td>
<td>86.7</td>
<td>87.5</td>
</tr>
<tr>
<td>Camera</td>
<td>85.2</td>
<td>88.3</td>
<td>89.3</td>
<td>90</td>
<td>88</td>
<td>91.3</td>
<td>89.7</td>
<td>91.8</td>
<td>89.5</td>
<td>91.1</td>
<td>92.2</td>
</tr>
<tr>
<td>Health</td>
<td>84.5</td>
<td>84</td>
<td>86.3</td>
<td>85.9</td>
<td>85.2</td>
<td>88.1</td>
<td>90.3</td>
<td>90.3</td>
<td>89.5</td>
<td>88.9</td>
<td>91.1</td>
</tr>
<tr>
<td>Music</td>
<td>76.7</td>
<td>78.5</td>
<td>80</td>
<td>82.5</td>
<td>79.4</td>
<td>82.6</td>
<td>86.3</td>
<td>85</td>
<td>85.2</td>
<td>85.8</td>
<td>88.3</td>
</tr>
<tr>
<td>Toys</td>
<td>83.2</td>
<td>83.7</td>
<td>84.3</td>
<td>85.4</td>
<td>84.2</td>
<td>88.8</td>
<td>88.5</td>
<td>89.5</td>
<td>89</td>
<td>88.3</td>
<td>85.4</td>
</tr>
<tr>
<td>Video</td>
<td>81.5</td>
<td>83.7</td>
<td>84.3</td>
<td>85.6</td>
<td>83.8</td>
<td>85.5</td>
<td>88.3</td>
<td>89.5</td>
<td>87</td>
<td>88.9</td>
<td>90</td>
</tr>
<tr>
<td>Baby</td>
<td>84.7</td>
<td>85.5</td>
<td>84</td>
<td>86.7</td>
<td>85.2</td>
<td>89.8</td>
<td>88</td>
<td>90.5</td>
<td>89.2</td>
<td>88.4</td>
<td>91.7</td>
</tr>
<tr>
<td>Magazines</td>
<td>89.2</td>
<td>91.3</td>
<td>92.3</td>
<td>92</td>
<td>91.2</td>
<td>92.4</td>
<td>91</td>
<td>92</td>
<td>92</td>
<td>92.2</td>
<td>95.6</td>
</tr>
<tr>
<td>Software</td>
<td>84.7</td>
<td>86.5</td>
<td>88.3</td>
<td>86.7</td>
<td>86.6</td>
<td>87.3</td>
<td>88.5</td>
<td>90.8</td>
<td>89.7</td>
<td>90</td>
<td>91.7</td>
</tr>
<tr>
<td>Sports</td>
<td>81.7</td>
<td>82</td>
<td>82.5</td>
<td>84.2</td>
<td>82.6</td>
<td>86.7</td>
<td>87</td>
<td>89.8</td>
<td>87.7</td>
<td>87.5</td>
<td>87.5</td>
</tr>
<tr>
<td>IMDB</td>
<td>81.7</td>
<td>77</td>
<td>83.5</td>
<td>84.2</td>
<td>81.6</td>
<td>85.8</td>
<td>88</td>
<td>89.8</td>
<td>87</td>
<td>88.9</td>
<td>90</td>
</tr>
</tbody>
</table>

**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
References


[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.


