Differentiated Attentive Representation Learning for Sentence Classification

Qianrong Zhou, Xiaojie Wang, Xuan Dong
Center for Intelligence Science and Technology,
School of Computer Science, Beijing University of Posts and Telecommunications
{zhouqr,xjwang,dongxuan8811}@bupt.edu.cn

Abstract

Attention-based models have shown to be effective in learning representations for sentence classification. They are typically equipped with multi-hop attention mechanism. However, existing multi-hop models still suffer from the problem of paying much attention to the most frequently noticed words, which might not be important to classify the current sentence. And there is a lack of explicitly effective way that helps the attention to be shifted out of a wrong part in the sentence. In this paper, we alleviate this problem by proposing a differentiated attentive learning model. It is composed of two branches of attention subnets and an example discriminator. An explicit signal with the loss information of the first attention subnet is passed on to the second one to drive them to learn different attentive preference. The example discriminator then selects the suitable attention subnet for sentence classification. Experimental results on real and synthetic datasets demonstrate the effectiveness of our model.

1 Introduction

Representation learning for different linguistic units is a fundamental problem in Natural Language Processing (NLP). Learning algorithms with better representations always achieve better performance on downstream tasks [Collobert *et al.*, 2011; Le and Mikolov, 2014; Kiros *et al.*, 2015]. As one of the most common NLP tasks, sentence classification relies heavily on the learned representation of sentences. Much work, especially with deep neural networks in recent years, has been done on sentence representation learning [Socher *et al.*, 2013; Kim, 2014].

More recently, deep models for representation learning are often augmented with various attention mechanisms and have achieved significant improvements on sentence classification [Yang *et al.*, 2016; Cheng *et al.*, 2016; Chen *et al.*, 2017]. Attention-based models show more interpretability compared to other neural models. They are illustrated to be able to assign high attention weights to important parts of the sentences. Different parts are then combined using attention weights. This is, to some extent, consistent with the way that humans classify sentences, as we often pay more attention to important parts and combine different parts with different attentions to form a whole picture of sentences.

Current methods with attention mechanisms generally fall into two categories: models with single-hop attention and with multi-hop attention. Single-hop models utilize either a randomly initialized context vector or an early-stage representation to locate important words in the sentence [Yang *et al.*, 2016; Liu *et al.*, 2016]. They often get better performance, but likely fail on complicated sentences [Kumar *et al.*, 2016; Lin *et al.*, 2017]. Multi-hop models repeatedly or iteratively extract information from an explicit memory network [Sukhbaatar *et al.*, 2015; Kumar *et al.*, 2016; Munkhdalai and Yu, 2017; Lin *et al.*, 2017]. They are also called Memory Controller (MC) models. MC models show superior performance over the state-of-the-art single-hop models, especially on those complicated sentences.

Despite the large improvements brought by attention mechanisms on sentence classification, there is still an important problem on them. Single-hop mechanism is often insufficient to locate words or features that are useful for classification. As shown by Kumar *et al.* [2016], the attention tends to be paid much to the most frequently noticed words, but less to other words which might be crucial for the task at hand. For example, in a negative polarity sentence "My response to the film is best described as lukewarm.", the strong sentiment word "best" tends to get a lot of attention, while "lukewarm" is in fact the key to the classification of the sentence. The same phenomenon can also be observed in machine translation [Tu *et al.*, 2016]. We refer to this as *attention bias* problem. Multi-hop mechanism tries to alleviate it by reassigning attention weights multiple times. However, multi-hop mechanism typically shares the same structures and parameters in all hops, leading to similar attention weights for all hops [Lin *et al.*, 2017] or more focus on some specific words [Kumar *et al.*, 2016]. It is still not a good solution for the attention bias problem.

Intuitively, we think that an effective way to address the problem is to pass on an explicit signal about how well the previous hop works to the subsequent hop. The subsequent hop then decides if it should shift its attention according to that signal. Specifically, for sentence classification, some sentences are correctly classified and others are not at the first hop. A signal with this information passed on to the second

hop will drive the model to shift its attention to other words or features at the second hop. In this way, the two hops could have differentiated attentive preference. The attention bias problem can be alleviated effectively.

This paper proposes a Differentiated Attentive Representation Learning Model (DARLM) to implement the above idea on sentence classification. DARLM has two branches of attention subnets and an example discriminator. One branch tries its best to classify all sentences, while the other is enabled for sentences that cannot be handled well by the former. The two branches are jointly trained. To achieve this, a specially designed signal related to the loss of the first branch is passed on to the second one. It ensures the differentiated training of the two-branch architecture and promotes the latter branch to shift its attention to different parts of a sentence. The example discriminator is introduced to select one branch to give the final label for each sentence. Experimental results on real and synthetic datasets demonstrate the effectiveness of DARLM comparing with a number of competitive baselines. The results show that DARLM is flexible for giving differentiated attention and capable of producing more discriminative representations for different sentences.

The contributions of this paper can be summarized as follows.

1. We propose a novel DARLM architecture for alleviating the attention bias problem on sentence classification. The model has two attention subnets and an example discriminator which is used to select the suitable attention subnet when classifying each sentence.
2. We introduce an explicit signal to drive attention subnets to learn differentiated attention, as well as a joint training method for DARLM to improve its effectiveness.
3. We illustrate the differentiated attentive preference learned by two attention subnets.

2 Related Work

Most recent representation learning models for sentence classification or some other tasks fall into two categories: attention-based models and composition-based models. Attention-based models, including single-hop models and multi-hop models, focus on extracting task-relevant information to compose the representation. Among single-hop models, Yang *et al.* [2016] proposed a two-level single-hop mechanism to locate important words and sentences, Liu *et al.* [2016] utilized an early-stage sentence representation to attend key words. However, single-hop mechanism was often insufficient to locate useful words or features. Multi-hop mechanism was then proposed to improve the capability of attention in recent work. It explored to use different kinds of control policies between adjacent hops. Kumar *et al.* [2016] introduced an episodic module to iteratively locate the input. Munkhdalai and Yu [2017] proposed a variable sized memory where information can be read and written through attention. Lin *et al.* [2017] proposed a self-attention mechanism with an attention weight regularization to extract different aspects of the sentence into multiple representations. While multi-hop models focused on how to update attentions to the correct parts consistently on all sentences by the final supervision information, our model tries to learn differentiated attention on different hops by passing on a signal about how well the previous hop works to the subsequent hop.

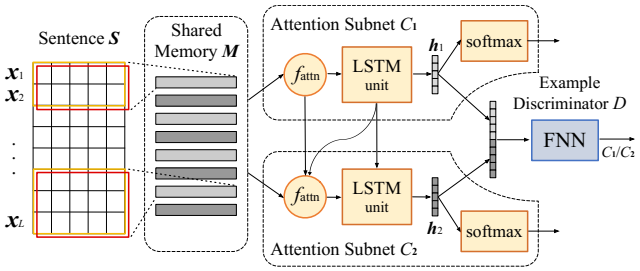


Figure 1: The overall architecture of DARLM.

Unlike attention-based models, composition-based models focus on different compositional strategies that aggregate words [Tai *et al.*, 2015; Zhou *et al.*, 2016; Liu *et al.*, 2017]. The recursive neural networks among them have become top performing techniques [Zhao *et al.*, 2015; Looks *et al.*, 2017]. However, their dependence on syntactic parse trees limits the practical applications. Though there have been few studies on latent tree learning models without restore to conventional parsers [Choi *et al.*, 2017], it still leaves open an important question that whether these models owe their effectiveness to latent grammars [Williams *et al.*, 2017].

3 Proposed Method

The architecture of proposed DARLM for sentence classification is shown in Figure 1. It is composed of a *shared memory* network, two branches of *attention subnets* with same structure but parameterized differently and an *example discriminator* network. The shared memory maintains encoded memories of the sentence. Each attention subnet then reads a set of relevant memories to compose one representation and gives its own prediction. Finally, one branch of attention subnets is selected by the example discriminator to output the final prediction.

In the following we describe shared memory and attention subnets in Section 3.1 and Section 3.2 respectively, and example discriminator in Section 3.3 together with the joint and differentiated training method for the whole model.

3.1 Shared Memory

Given a sentence with L tokens, it can be represented as:

$$\mathbf{S} = \oplus_i^L \mathbf{x}_i, \tag{1}$$

where $\oplus$ denotes concatenation, $\mathbf{x}_i$ is a d dimensional word embedding for i-th token in the sentence, and $\mathbf{S}$ is therefore a $d \times L$ matrix. A one-layer Convolutional Neural Network (CNN) is used to encode the matrix. A set of fixed-width-window convolutional filters slide over $\mathbf{S}$ (padded where necessary), generating one memory slot at a time. Each slot $\mathbf{m}_i$ contains information about a n-gram:

$$\mathbf{m}_i = \mathrm{ReLU}\left(\mathbf{W}_m \odot \mathbf{x}_{i-\lfloor\frac{n-1}{2}\rfloor:i+\lceil\frac{n-1}{2}\rceil} + \mathbf{b}_m\right), i \in [1, L], \tag{2}$$

where $\mathbf{W}_m$ and $\mathbf{b}_m$ are convolution parameters involving d filters, ReLU is the rectified linear unit function and $\odot$ denotes convolutional operation. Thus, all tokens are encoded to a shared memory $\mathbf{M}$:

$$\mathbf{M} = \oplus_i^L \mathbf{m}_i. \tag{3}$$

3.2 Attention Subnets

As shown in Figure 1, both attention subnets C_1 and C_2 have access to the shared memory $\mathbf{M}$ with attention mechanism. They extract the context vectors and then input them into two connected Long-Short Term Memory (LSTM) units. LSTM units then generate the high-level representations. Finally, two attention subnets predict the probabilities of sentence $\mathbf{S}$ belonging to class labels $Y = \{y_1, ..., y_k\}$ respectively. In this process, two attention subnets are not independent but correlated by the inside attention operations and LSTM states.

Specifically, for each attention subnet $C_t\,(t \in \{1, 2\})$, we calculate attention weight $\boldsymbol{\alpha}_t$ and context vector $\mathbf{s}_t$ as:

$$\begin{aligned}\mathbf{Z}_t = \tanh\big(&\mathbf{W}_t^s\,(\mathbf{s}_{t-1} \otimes \mathbf{e}_L) + \mathbf{W}_t^h\,(\mathbf{h}_{t-1} \otimes \mathbf{e}_L) \\ &+ \mathbf{W}_t^m \mathbf{M}\big),\end{aligned} \tag{4}$$

$$\boldsymbol{\alpha}_t = \mathrm{softmax}(\mathbf{w}_t \mathbf{Z}_t), \tag{5}$$

$$\mathbf{s}_t = \mathbf{M}\boldsymbol{\alpha}_t^T, \tag{6}$$

where $\mathbf{W}_t^s$, $\mathbf{W}_t^h$, $\mathbf{W}_t^m$ and $\mathbf{w}_t$ are attention parameters, $\mathbf{s}_{t-1}$ and $\mathbf{h}_{t-1}$ represent the context vector and high-level representation of previous attention subnet respectively, $\mathbf{e}_L$ is vector of ones, $\otimes$ denotes the out product of two vectors.

The LSTM unit takes $\mathbf{s}_t$ and $\mathbf{h}_{t-1}$ as inputs and outputs the current high-level representation:

$$\mathbf{h}_t = \mathrm{LSTM}_t\,(\mathbf{s}_t, \mathbf{h}_{t-1})\,. \tag{7}$$

The high-level representation $\mathbf{h}_t$ is then fed through a softmax classifier to predict the probability distribution over class labels:

$$P_t(Y|\mathbf{S}) = \mathrm{softmax}\,(\mathbf{U}_t \mathbf{h}_t + \mathbf{b}_t)\,, \tag{8}$$

where $\mathbf{U}_t$ are parameter matrix and $\mathbf{b}_t$ is the bias term.

The initial $\mathbf{s}_0$ is predicted by an average of memory slots fed through a one-layer feed forward neural network:

$$\mathbf{s}_0 = \mathrm{FNN}_s\left(\frac{1}{L}\sum_j^L \mathbf{m}_j\right), \tag{9}$$

and $\mathbf{h}_0$ is initialized with a vector of zeros.

3.3 Training of DARLM

Ideally, C_2 should learn to pay more attention to those important words that have not been noticed by C_1. However, optimizing multiple classification losses on the same label directly cannot reach this goal. Therefore, we propose a differentiated training method to drive C_2 to shift its attention to different parts of the sentence.

Differentiated Loss

We introduce our method starting from cross-entropy loss:

$$l_t = -\log p_t, \tag{10}$$

where $p_t \in [0, 1]$ is the estimated probability for the *target* label by $C_t\,(t \in \{1, 2\})$. Instead of simply summing up the losses of two attention subnets, we add a modulating term $\Phi(\cdot)$ to the loss function of C_2. More formally, we define the differentiated loss function as:

$$l_c = l_1 + \Phi(p_1)\,l_2. \tag{11}$$

The modulating term $\Phi(\cdot)$ could take different function forms. We adopt Beta Probability Density Function (Beta-PDF):

$$\Phi(p_1) = \frac{1}{B\,(a, b)}\,(p_1)^{a-1}\,(1 - p_1)^{b-1}\,, \tag{12}$$

where a and b are two positive shape hyper-parameters, $B(\cdot)$ is beta function. $\Phi(p_1)$ has some nice properties when defined as Beta-PDF. For example, for $\Phi(p_1)$ with $a = 1$ and $b = 3$. If C_1 correctly classifies a sentence, $\Phi(p_1)$ will have a small value. It shrinks the total loss of two attention subnets and gets C_2 to be rarely trained on that sentence. If C_1 misclassifies a sentence, $\Phi(p_1)$ tends to have a high value. It enlarges the total loss of two attention subnets and makes C_2 receive more training on that sentence, thereby giving a chance to C_2 to shift its attention to more useful words that are not noticed by C_1. We can see that through the modulating term $\Phi(p_1)$, an explicit signal about how well C_1 works is passed on to C_2. Two attention subnets then could have differentiated attentive preference and are complementary to each other (see more details in Section 4.2).

Example Discriminator

The example discriminator D is introduced to select one attention subnet to output the final label. It typically consists of two-layer feed forward neural network followed by a softmax classifier, and is trained simultaneously with other components. The high-level representations $\mathbf{h}_1$ and $\mathbf{h}_2$ are supplied as input, and the probabilities p_1 and p_2 provide supervised information. Thus, the probability distribution over two attention subnets and the loss function are defined as follows:

$$P_D(C|\mathbf{S}) = \mathrm{softmax}\,(\mathrm{FNN}_D\,(\mathbf{h}_1 \oplus \mathbf{h}_2))\,, \tag{13}$$

$$l_d = -\frac{p_1}{p_1 + p_2}\log p_{c_1} - \frac{p_2}{p_1 + p_2}\log p_{c_2}, \tag{14}$$

where p_{c_1} and p_{c_2} represent D's estimated probabilities for C_1 and C_2 respectively. From Eq. (14), we can see that the goal of D is to learn which attention subnet gives more accurate estimattion to the probability of target label.

Confidence Penalization

In practice, we find out that, when C_1 places all probability on a single class in the training set, C_2 tends to be inadequately trained. Or conversely, when C_1 predicts a much smoother output distribution, the modulating term $\Phi(p_1)$ has little or no effect on C_2. Thus, we add a penalization term that prevents peaked or smooth output distributions of C_1:

$$l_e = H\,(P_1(Y|\mathbf{S}))\,, \tag{15}$$

where $H(\cdot)$ represents the entropy of a probability distribution.

Put It All Together

Finally, we combine all above loss functions:

$$\mathcal{L}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{j}^{\mathcal{D}} l^{(j)}, \tag{16}$$

$$l^{(j)} = l_c^{(j)} + l_d^{(j)} - \lambda l_e^{(j)}, \tag{17}$$

where $\mathcal{D}$ is the training corpora and λ controls the strength of the penalty. All trainable parameters in the model are jointly learned by minimizing $\mathcal{L}(\mathcal{D})$.

4 Experiments

In this section, we empirically evaluate the performance of DARLM and compare it with the state-of-the-art models. To further show the effectiveness of our model and give some insights into what differentiated attention our model learns, we also investigate a sentence classification task involving a synthetic dataset - *Toy Zoo*. The codes and datasets are publicly available at `https://github.com/Chanrom/DARLM`.

4.1 Comparison Experiments

Datasets

We evaluate our model on four different datasets. Each dataset is briefly described as follows.

SST is a popular sentiment classification dataset introduced by Socher *et al.* [2013]. The review sentences are annotated with five classes. We only use sentence-level annotation. Standard train/dev/test split is used.

TREC is a question type classification dataset [Li and Roth, 2002], where questions are labeled with six classes. We randomly split 500 questions in the training set into a development set.

SUBJ is a subjectivity dataset where each snippet can be classified as subjective or objective. [Pang and Lee, 2004].

MR is a movie reviews with positive/negative labels [Pang and Lee, 2005]. We follow the same split as [Liu *et al.*, 2017] on above two datasets.

Implementation Details

The word embedding size, LSTM hidden size and number of hidden units inside all fully connected layers are set to 300. Convolution window sizes are 3, 4 and 5, and each window size has 100 filters. The word embeddings are initialized with the pre-trained GloVe vectors [Pennington *et al.*, 2014] and fine-tuned during training. Other parameters are initialized from a uniform distribution in $[-0.1, 0.1]$. For regularization, we apply dropout [Srivastava *et al.*, 2014] with a dropout rate of 0.5 to all layers (except those in example discriminator) and batch normalization to the outputs of one-layer CNN.

The model is trained using mini-batch stochastic gradient descent with the RMSProp optimizer in a total of 30 epochs. The initial learning rate is set to 0.0005 and mini-batch size is 16. The hyper-parameter a is set to 1 for all experiments, while b is estimates by grid search across the set $\{2, 3, 4, 5\}$. For the coefficient λ, we empirically set it to a positive or negative number for different datasets.

Comparison Methods

We compare DARLM with three types of strong baselines: single-hop models, multi-hop models and composition-based models.

HAN: A hierarchical attention model for text classification. We only adopt the word-level attention mechanism [Yang *et al.*, 2016].

Bi-LSTM-IA: A bidirectional LSTM (Bi-LSTM) with inner-attention, utilizing the sentence first-stage representation to attend words [Liu *et al.*, 2016].

DMN: Dynamic memory network, using an iterative attention process to search the relevant facts [Kumar *et al.*, 2016].

NSE: Neural semantic encoder, equipped with a variable sized memory which can be accessed during read and write according to attention weights [Munkhdalai and Yu, 2017].

SELF-ATTN: A structured self-attentive model, utilizing multi-hop mechanism to extract different aspects of the sentence into multiple vector representations [Lin *et al.*, 2017].

Tree-LSTM: Tree-Structured LSTM that requires predefined syntactic structures [Tai *et al.*, 2015].

AdaSent: A self-adaptive model that forms a hierarchy of representations from words to sentences through recursive gated networks [Zhao *et al.*, 2015].

BLSTM-2DCNN: It utilizes 2D max pooling to extract features from Bi-LSTM hidden states [Zhou *et al.*, 2016].

DSCNN: Dependency sensitive CNN, which utilizes an LSTM to extract low-level representation, and then apply a CNN to extract task-specific features [Zhang *et al.*, 2016].

DC-TreeLSTM: Dynamic compositional neural networks over tree structure, in which the compositional function is dynamically generated by a meta network [Liu *et al.*, 2017].

LR-Bi-LSTM: A Bi-LSTM for sentiment classification, leveraging the linguistic knowledge by imposing sentiment regularizers on intermediate outputs [Qian *et al.*, 2017].

To investigate the effect of the differentiated loss, we also compare against Attentive Representation Learning Models (ARLMs). ARLM is composed of a memory network and only one branch of attention subnet. It is a standard MC model and trained in a normal way. We get ARLM-Single without unfolding LSTM unit in the attention subnet, and ARLM-Multi by unfolding LSTM unit two or more times (the best results are reported). The final hidden state of LSTM unit is used as sentence representation for classification. For the models without results on the above four datasets, we reimplement them and perform grid search over key hyper-parameters (such as learning rate, batch size and the number of hops). Statistical significance tests are adopted for comparisons.

Results Analysis

Table 1 shows the experimental results of DARLM and other baselines. The classification performance is evaluated in terms of accuracy. Among all models, DARLM consistently outperforms single-hop and multi-hop models on all datasets (with a p-value that is smaller than 0.05), and it outperforms almost all composition-based models. Specifically, compared with the best single-hop models, DARLM achieves a 1.8% improvement on SST, a 2.8% improvement on TREC, a 0.8% improvement on SUBJ and a 1.0% improvement on MR.

Models	SST	TREC	SUBJ	MR
HAN	47.0	92.8	92.9	82.2
Bi-LSTM-IA	47.0	92.8	93.3	81.5
ARLM-Single	46.2	93.2	92.5	80.8
DMN	47.3	93.8	93.3	82.1
NSE	48.7	94.8	93.5	81.7
SELF-ATTN	47.4	95.0	93.5	82.3
ARLM-Multi	47.8	93.8	93.2	81.4
Tree-LSTM	48.1	-	93.2	80.7
AdaSent	-	92.4	95.5	83.1
BLSTM-2DCNN	-	96.1	94.0	82.3
DSCNN	-	95.6	93.9	82.2
DC-TreeLSTM	-	93.8	93.7	81.7
LR-Bi-LSTM	48.6	-	-	82.1
DARLM	**48.8**	96.0	94.1	**83.2**

Table 1: Evaluation results of DARLM and other models.

Compared with multi-hop models DMN, NSE and SELF-ATTN, which use Bi-LSTM or two-layer LSTM to encode the input sentence, DARLM only uses a simple one-layer CNN. But DARLM achieves at least a 0.1% improvement on SST, a 1.0% improvement on TREC, a 0.6% improvement on SUBJ and a 0.9% improvement on MR. Considering that one-layer CNN is less powerful for modeling long term dependencies than Bi-LSTM or two-layer LSTM, it seems clear that differentiated learning of the attention subnets in DARLM plays a key role to locate important words in different parts of a sentence. Moreover, DARLM significantly outperforms some composition-based models which utilize syntactic structures or external linguistic knowledge (Tree-LSTM, DC-TreeLSTM and LR-Bi-LSTM). It achieves the best accuracy on SST and MR over all baseline models and reaches comparable performance to the state-of-the-art model BLSTM-2DCNN on TREC. On the SUBJ dataset, our model is better than all other baseline models except AdaSent.

4.2 Analysis of DARLM

Analysis of Attention

In order to understand how differentiated training method contributes to the model's performance and give some insights into what attentions the model learns, we investigate a classification task on a synthetic dataset, Toy Zoo. The dataset includes 10,000 sentences. Inspired by [Krishnamurthy and Mitchell, 2013], each sentence is designed to contain 5 Noun Phrases (NP) with the form of adverb-adjective-noun. Each NP describes an animal. The name of an animal is called animal-N for simplification. The larger the number N, the bigger the animal itself. But some modifiers, such as very very big and very small, can change the relations. A lookup table for each NP and the corresponding size is maintained. The task is to find the biggest animal mentioned in a sentence. Figure 2 shows an example. There are 5 animals including a quite big animal-6, a very small animal-8 and others. The size of each animal is given in the vector [0.35, 0.43, 0.45, 0.39, 0.41]. The answer for this sentence is the third one. Compared with the real datasets, Toy Zoo has clear linguistic patterns, which might be helpful for us to understand and explain our model.

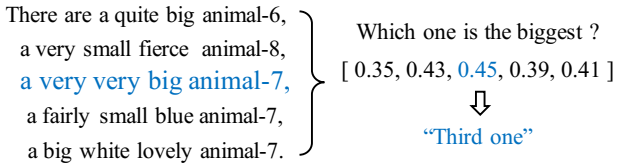


Figure 2: A example sentence in Toy Zoo.

Models	Test Acc.
ARLM	91.8
ARLM + 2-hop + last	92.2
ARLM + 3-hop + last	92.3
ARLM + 4-hop + last	89.8
ARLM + 2-hop + avg pooling	91.0
ARLM + 3-hop + avg pooling	89.6
DARLM	**93.7**

Table 2: Accuracies on Toy Zoo classification task.

The baseline models, ARLMs, create sentence representations by either using the last hidden state or the average pooling of all hidden states. The results are shown in Table 2. We can observe that DARLM significantly outperforms all ARLMs. Varying hops and different representations cannot compensate the lack of differentiated loss. Figure 3 shows an example of attention visualization for DARLM and the best ARLM. We can find that both models locate "animal-10" at first. However, ARLM irretrievably focuses on this wrong word at the second and third hop, while DARLM shifts its attention to "very" correctly at the second hop (namely, C_2).

A more interesting observation is that C_1 and C_2 actually tend to have different attentive preference. As shown in Table 3, we list some words that frequently receive attentions in different attention subnets. We find that C_1 tends to be attentive to nouns, while C_2 tends to be attentive to adverbs and adjectives. The phenomenon of differentiated attention can also be observed on real datasets (see in Table 3). However, ARLM always pays a lot of attention to the same words. Furthermore, if we cut off all connections between C_1 and C_2 and remove the differentiated loss, there is almost no attention complementation on any dataset. From these, it seems that DARLM can alleviate the attention bias problem effectively with differentiated attention.

Performance of Example Discriminator

The example discriminator D is used to select one attention subnet to output the final label. Its prediction dominates the performance of DARLM. Taking the models we used for Toy Zoo, SST and TREC as examples, we investigate how D influences the performance of DARLM.

We first examine the stability that D selects a certain attention subnet as the labeler for a given sentence after some epochs of training. From Eq. (14), we know that the ground truth label of D is automatically determined by the predictions of C_1 and C_2 in the training. However, it's possible that some sentences are better suited to C_2 in early stages of training, while in later stages they may be better suited to C_1, which disrupts the convergence of D. Thus, we define the

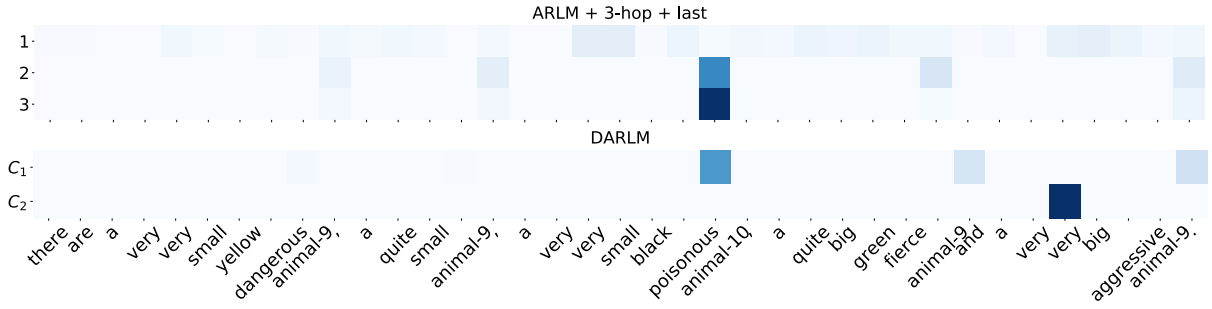
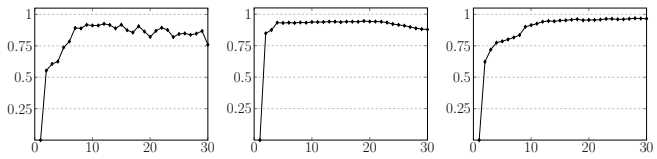


Figure 3: An example of attention visualization for DARLM and the best ARLM.

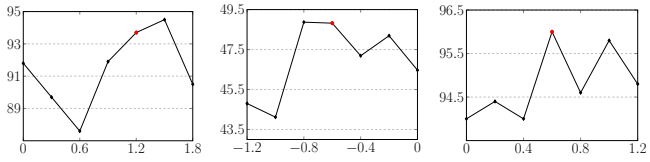
Models		Toy Zoo	SST	TREC
DARLM	C_1	animal-16, animal-13, animal-9, animal-14, animal-12, animal-8	Immediately, Ridiculous, confusing, Skip, jacked-up, stupider	Janice, How, desktop, you, bandwidth, why
	C_2	big, very, quite, pretty, a, fairly	3D, Brilliant, bonehead, Hmm, food, Good	I.V., USPS, NASA, CPR, stand, SOS
ARLM	2^{nd} hop	animal-16, animal-13, animal-14, animal-15, animal-9, “,”	Hmm, bet, Ridiculous, Brimful, Immediately, dreadful	tall, often, cold, fast, far, wide
	3^{rd} hop	animal-16, animal-13, animal-14, “,”, animal-9, animal-15	Hmm, bet, Ridiculous, Brimful, Immediately, dreadful	often, fast, cold, tall, far, wide

Table 3: Some most attentive words on the test sets of Toy Zoo, SST and TREC.



(a) Toy Zoo (b) SST (c) TREC

Figure 4: The stability scores of D across each epoch on training sets.



(a) Toy Zoo (b) SST (c) TREC

Figure 5: The effect of penalization term (by varying coefficient λ).

Datasets	C_1	C_2	D	DARLM
Toy Zoo	91.3	8.7	94.5	93.7
SST	47.3	35.1	51.0	48.8
TREC	95.6	70.2	96.4	96.0

Table 4: The performance of components in DARLM on test sets.

stability score of e-th epoch on the training set:

$$\omega^e = \frac{\sum_{t\in\{1,2\}} |O^{e-1}_{C_t} \cap O^e_{C_t}|}{|\mathcal{D}|}, e \in [1, 30], \qquad (18)$$

where $O^e_{C_t}$ is the set of sentences that are better suited to C_t at e-th epoch and $O^0_{C_t}$ is an empty set. Note that we prevent D from selecting the same attention subnet for all sentences in our experiments. Figure 4 shows the stability scores of D across each epoch on three dataset. It can be seen that for all three datasets, the ω converges to a number close to 1. Next, we present the performance of components in DARLM on the test sets (see in Table 4). We see that, despite the lower performance that C_1 and C_2 shows, the overall accuracy is highest because D selects the most suitable one for a given sentence. It shows that DARLM can produce more discriminative representations, improving the performance of classification.

Effect of Penalization Term

As described in Section 3.3, the purpose of introducing penalization term l_e is to promote the training of C_2. Without l_e, the example discriminator D would always select C_1 for every sentence due to the inadequate training of C_2. We present some results that show the effect of l_e by varying coefficient λ (see in Figure 5).

We can find that i) l_e helps DARLM to achieve better performance on three datasets by either preventing peaked (i.e. on TREC) or smooth (i.e. on SST) output distributions of C_1; ii) without having l_e, the performance of DARLM drops significantly.

5 Conclusion

In this work, we propose a differentiated attentive representation learning model (DARLM) for sentence classification. With a differentiated training method, two attention subnets in DARLM can have different attentive preference and generate different sentence representations. Experimental results demonstrate the effectiveness of DARLM and show that DARLM can effectively alleviate the problem of attention bias.

In future work, we are going to apply our method to multiple branches and use more powerful memory network.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This paper is supported by NSFC (No. 61273365), NSSFC (2016ZDA055), 111 Project (No. B08004), Beijing Advanced Innovation Center for Imaging Technology, Engineering Research Center of Information Networks of MOE, China.

References

[Chen *et al.*, 2017] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. Recurrent attention network on memory for aspect sentiment analysis. In *EMNLP*, 2017.

[Cheng *et al.*, 2016] Jianpeng Cheng, Li Dong, and Mirella Lapata. Long short-term memory-networks for machine reading. In *EMNLP*, 2016.

[Choi *et al.*, 2017] Jihun Choi, Kang Min Yoo, and Sang-goo Lee. Unsupervised learning of task-specific tree structures with tree-lstms. *arXiv preprint arXiv:1707.02786*, 2017.

[Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The JMLR*, 12(Aug):2493–2537, 2011.

[Kim, 2014] Yoon Kim. Convolutional neural networks for sentence classification. In *EMNLP*, 2014.

[Kiros *et al.*, 2015] Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Skip-thought vectors. In *Advances in NIPS*, pages 3294–3302, 2015.

[Krishnamurthy and Mitchell, 2013] Jayant Krishnamurthy and Tom M Mitchell. Vector space semantic parsing: a framework for compositional vector space models. In *ACL*, page 1, 2013.

[Kumar *et al.*, 2016] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. Ask me anything: dynamic memory networks for natural language processing. In *ICML*, pages 1378–1387, 2016.

[Le and Mikolov, 2014] Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[Li and Roth, 2002] Xin Li and Dan Roth. Learning question classifiers. In *Proceedings of the 19th International Conference on Computational Linguistics-Volume 1*, 2002.

[Lin *et al.*, 2017] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. In *ICLR*, 2017.

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

[Liu *et al.*, 2017] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Dynamic compositional neural networks over tree structure. In *IJCAI*, 2017.

[Looks *et al.*, 2017] Moshe Looks, Marcello Herreshoff, DeLesley Hutchins, and Peter Norvig. Deep learning with dynamic computation graphs. In *ICLR*, 2017.

[Munkhdalai and Yu, 2017] Tsendsuren Munkhdalai and Hong Yu. Neural semantic encoders. In *EACL*, 2017.

[Pang and Lee, 2004] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *ACL*, 2004.

[Pang and Lee, 2005] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, 2005.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: global vectors for word representation. In *EMNLP*, 2014.

[Qian *et al.*, 2017] Qiao Qian, Minlie Huang, Jinhao Lei, and Xiaoyan Zhu. Linguistically regularized lstms for sentiment classification. In *ACL*, 2017.

[Socher *et al.*, 2013] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642, 2013.

[Srivastava *et al.*, 2014] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The JMLR*, 2014.

[Sukhbaatar *et al.*, 2015] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. End-to-end memory networks. In *Advances in NIPS*, pages 2440–2448, 2015.

[Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, 2015.

[Tu *et al.*, 2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. In *ACL*, pages 76–85, 2016.

[Williams *et al.*, 2017] Adina Williams, Andrew Drozdov, and Samuel R Bowman. Learning to parse from a semantic objective: It works. is it syntax? *arXiv preprint arXiv:1709.01121*, 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 *HLT-NAACL*, pages 1480–1489, 2016.

[Zhang *et al.*, 2016] Rui Zhang, Honglak Lee, and Dragomir Radev. Dependency sensitive convolutional neural networks for modeling sentences and documents. In *NAACL-HLT*, pages 1512–1521, 2016.

[Zhao *et al.*, 2015] Han Zhao, Zhengdong Lu, and Pascal Poupart. Self-adaptive hierarchical sentence model. In *IJCAI*, pages 4069–4076, 2015.

[Zhou *et al.*, 2016] Peng Zhou, Zhenyu Qi, Suncong Zheng, Jiaming Xu, Hongyun Bao, and Bo Xu. Text classification improved by integrating bidirectional lstm with two-dimensional max pooling. In *COLING*, 2016.