Multi-glance Reading Model for Text Understanding

Pengcheng Zhu[1,2], **Yujiu Yang**[1], **Wenqiang Gao**[1], **and Yi Liu**[2]
Graduate School at Shenzhen, Tsinghua University[1]
Peking University Shenzhen Institute[2]
zhupc15@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn,
gwq16@mails.tsinghua.edu.cn, eeyliu@gmail.com

Abstract

In recent years, a variety of recurrent neural networks have been proposed, e.g LSTM, however, existing models only read the text once, it cannot describe the situation of repeated reading in reading comprehension. In fact, when reading or analyzing a text, we may read the text several times rather than once if we couldn't well understand it. So, how to model this kind of the reading behavior? To address the issue, we propose a multi-glance mechanism (MGM) for modeling the habit of reading behavior. In the proposed framework, the actual reading process can be fully simulated, and then the obtained information can be consistent with the task. Based on the multi-glance mechanism, we design two types of recurrent neural network models for repeated reading: Glance Cell Model (GCM) and Glance Gate Model (GGM). Visualization analysis of the GCM and the GGM demonstrates the effectiveness of multi-glance mechanisms. Experiments results on the large-scale datasets show that the proposed methods can achieve better performance.

1 Introduction

Text understanding is one of the fundamental tasks in Natural Language Processing areas. These years we have seen significant progress in applying neural networks to text analysis applications. Recurrent neural network is widely used because of its effective capability of capturing the sequential information. Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent neural network (Chung et al., 2014) have achieved state-of-the-art performance in many areas, such as sentiment analysis (Tang et al., 2014; Chen et al., 2016), document classification (Yang et al., 2016) and neural machine translation (Bahdanau et al., 2014). Besides the success achieved by these basic recurrent neural models, there are also a lot of interesting research works conducted in text analysis (Kim, 2014; Zhang et al., 2015). Depending on the parsing tree structures, tree-LSTM (Tai et al., 2015) and recursive neural network(Socher et al., 2013) are proposed. Bidirectional recurrent neural networks (Schuster and Paliwal, 1997) can get the backward features. In order to align the hidden states, attention mechanism is widely used in language processing (Bahdanau et al., 2014; Vaswani et al., 2017).

One of the common characteristics of these existing models is to model only single reading processing and generate a sequence of hidden states h_t, as a function of the previous hidden states h_{t-1} and the current input (Sutskever et al., 2014; Karpathy et al., 2015). However, the fact is that when we read a text only once, we may merely know the general idea of it, especially when the text is long and obscure. More often than not, we know that fast repeated reading is more effective than slow careful reading, so, for the obscure text, our primary school teacher always teaches us to read several times to get the theme of the text. In addition, this kind of rereading can help us find some of the details that are ignored when we first glance.

In this paper, we propose a novel multi-glance mechanism to model our reading habit: when reading a text, first we will glance through it to get the general meaning and then based on the information we obtained, we will read the text again in order to find some important contents. Based on the multi-glance mechanism we proposed (Fig.1), we design different models for processing the obtained information by the last glance, that it,

Proceedings of the Eighth Workshop on Cognitive Aspects of Computational Language Learning and Processing, pages 27–35
Melbourne, Australia, July 19, 2018. ©2018 Association for Computational Linguistics

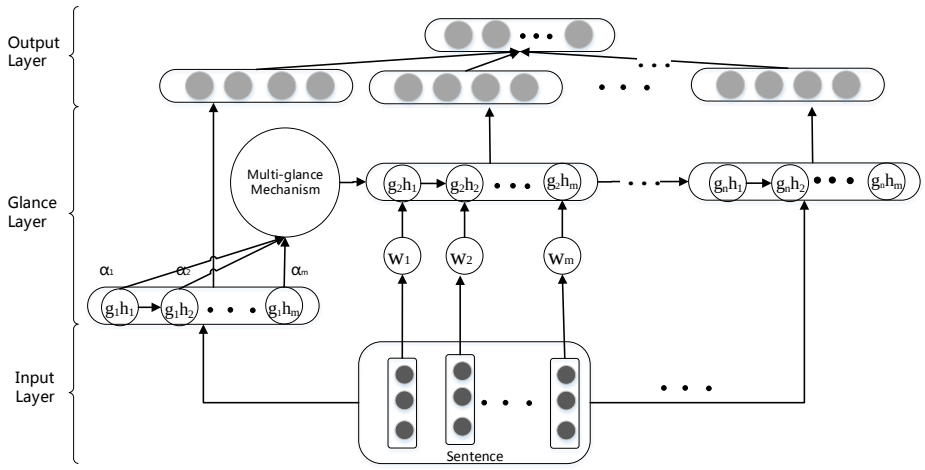


Figure 1: The architecture of Multi-glance Mechanism (MGM) model

Glance Cell Model (GCM) and Glance Gate Model (GGM). GCM has a special cell to memorize the first impression information obtained after finishing the first reading. GGM has a special gate to control current input and output in order to filter words that are not important. The main contributions of this work are summarized as follows:

- We propose a novel multi-glance mechanism which models the habit of reading. Comparing to traditional sequential models, our proposed models can better simulate people's reading process and better understand the content.
- Based on multi-glance mechanism, we propose GCM which can takes the first impression information into consideration. Glance cell model has a special cell to memorize the global impression information we obtain and add it into the current calculation.
- Based on multi-glance mechanism, we propose GGM which adopts a extra gate to ignore the less important words and focus on details in the contents.

2 Related Work

Recurrent neural network has achieved great success because of its effective capability to capture the sequential information. The RNN handles the variable-length sequence by having a recurrent hidden state whose activation at each time step is dependent on that of the previous time. To reduce the negative impact of gradient vanishing, a long short-term memory unit (Hochreiter and Schmidhuber, 1997), which has a more sophisticated activation function, was proposed. Bidirectional recurrent neural networks (Schuster and Paliwal, 1997), e.g bidirectional LSTM networks (Augenstein et al., 2016), combine forward features as well as reverse features of the text. Bidirectional networks, which get the forward features and the reverse features separately, are different from our multi-glance mechanism. A Gated Recurrent Unit (GRU) (Cho et al., 2014) is a good extension of a LSTM unit, because GRU maintains the performance and makes the structure to be simpler. Comparing to a LSTM unit, a GRU has only two gates, an update gate and a reset gate, so it will be faster to train a GRU than a LSTM unit. Attention mechanism (Bahdanau et al., 2014) is used to learn weights for every input, so it can reduce the impact of information redundancy. Now, attention mechanism is commonly used in various models.

3 Methods

In this section, we will introduce the proposed multi-glance mechanism models in detail. We first describe the basic framework of multi-glance mechanism. Afterwards, based on multi-glance mechanism, we describe two glance models, glance cell model and glance gate model.

3.1 Multi-glance Mechanism Model

When reading or analyzing a text, we may read it several times rather than once if we couldn't fully understand its meaning. To model our reading habit, we propose the multi-glance mechanism. The core architecture of the proposed model is shown in Fig.1.

In the following paper, we will describe how the models work when processing a text. Given a training text T, in order to better analyze it, we

will read T many times. As shown in Fig. 1, n is the times we will read the text.

For the sake of convenience, we give an example of the 2-glance process here.

First, we glance through the text to capture a general meaning. We use the recurrent network to read the embedding of each word and calculate the hidden states $\{g_1h_1, g_1h_2, \cdots, g_1h_m\}$, where m is the length of the text T. After finishing reading it, we have an impression on the text T. Next, with the guidance of the impression, we give these hidden states weight parameters and feed them into the glance model to continue to read the text for the second time. As we can see, if we read the text only once and don't adopt multi-glance mechanism, this model can be simplified as traditional attention based recurrent model.

At the second time of reading, in view of the general idea of the content we have got, we may ignore the less interesting words and focus on some details in the text. So we utilize a novel glance recurrent model to read embedding $T = \{w_1, w_2, \cdots, w_m\}$ again and calculate the output state $\{g_2h_1, g_2h_2, \cdots, g_2h_m\}$. Based on multi-glance mechanism, we propose two glance recurrent models: that is, **Glance Cell Model**(GCM) and **Glance Gate Model**(GGM).

Comparing to basic recurrent model, glance cell model has a special cell to memorize the general meaning calculated after finishing the first time of reading. Besides, glance gate model has a binary gate to filter the less important words. We describe how two glance recurrent models operate in section 3.2 and section 3.3. Fig.1 gives the main process of the multi-glance mechanism.

3.2 Glance Cell Model

Based on multi-glance mechanism, we propose the glance cell model (GCM). After we finish reading the text T for the first time, we know any of the general meaning of it. This means we have some first impression information about the text. As shown in Fig.2, comparing to the traditional recurrent network, the GCM has a special cell to keep the first impression information. LSTM has been widely adopted for text processing, so we use LSTM to calculate the hidden states g_1h_i.

Thus the glance cell state gc_t^c can be calculated from the weighted sum of hidden states

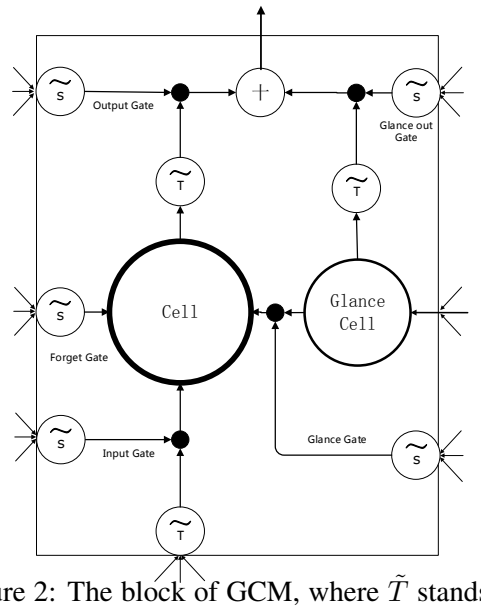


Figure 2: The block of GCM, where $\tilde{T}$ stands for $tanh()$ and $\tilde{S}$ stands for $sigmoid()$.

$\{g_1h_1, g_1h_2, \cdots, g_1h_m\}$:

$$gc_t^c = \sum_{i=1}^{m} \alpha_i \cdot g_1h_i \tag{1}$$

where α_i measures the impression of i_{th} word for the current glance cell state gc_t^c. Because GCM is a recurrent network as well, the current glance cell state gc_t^c is also influenced by the previous state $g_2h_{t-1}^c$ and the current input w_t. Thus the impression α_i can be defined as:

$$\alpha_i = \frac{exp(f(g_2h_{t-1}^c, w_t, g_1h_i^{lstm}))}{\sum_{i=1}^{m} exp(f(g_2h_{t-1}^c, w_t, g_1h_i^{lstm}))} \tag{2}$$

where f is the impression function and it can be defined as:

$$f(g_2h_{t-1}^c, w_t, g_1h_i^{lstm}) = gw_c^T \cdot \tanh(W_g^c \cdot [g_2h_{t-1}^c, w_t, g_1h_i^{lstm}] + b^c) \tag{3}$$

where W_g^c is the weight matrices and gw_c^T is the weight vector.

Besides, glance cell is used to memorize the prior knowledge, we also have a cell, at the second time reading in multi-glance mechanism, to read the text. We use three gates to update and output the cells states, and they can be defined as:

$$i_t^c = \sigma(W_i^c \cdot [g_2h_{t-1}^c, w_t] + b_i^c) \tag{4}$$
$$f_t^c = \sigma(W_f^c \cdot [g_2h_{t-1}^c, w_t] + b_f^c) \tag{5}$$
$$o_t^c = \sigma(W_o^c \cdot [g_2h_{t-1}^c, w_t] + b_o^c) \tag{6}$$
$$\tilde{c}_t^c = \tanh(W_c^c \cdot [g_2h_{t-1}^c, w_t] + b_c^c) \tag{7}$$

where i_t^c, f_t^c and o_t^c are the gates states, $\sigma()$ is the sigmoid function and $\tilde{c}_t^c$ stands for the input state.

In GCM, in order to adopt the first impression knowledge in the current cell state calculation and output the glance cell state, we use glance input gate and output gate to connect the glance cell and the cell state. The two gates can be defined as:

$$gi_t^c = \sigma(W_{gi}^c \cdot [g_2h_{t-1}^c, w_t, gc_t^c] + b_{gi}^c) \quad (8)$$
$$go_t^c = \sigma(W_{go}^c \cdot [g_2h_{t-1}^c, w_t, gc_t^c] + b_{go}^c) \quad (9)$$

where gi_t^c and go_t^c are the gate states.Thus the cell state can be calculated as:

$$c_t^c = f_t^c \odot c_{t-1}^c + i_t^c \odot \tilde{c}_t^c + gi_t^c \odot gc_t^c \quad (10)$$

where $\odot$ stands for element-wise multiplication.

According to the function, when we read the text at the second time, the current cell state c_t^c contains the previous cell state c_{t-1}^c, current input state $\tilde{c}_t^c$ and the current glance cell state gc_t^c, which is different from the existing recurrent models.

In view of two cells in GCM, the final output of a single block can be calculated as:

$$g_2h_t^c = o_t^c \odot \tanh(c_t^c) + go_t^c \odot \tanh(gc_t^c) \quad (11)$$

We feed the text $T = \{w_1, w_2, \cdots, w_m\}$ embedding into the glance cell model and then obtain the output hidden states $g_2h^c = \{g_2h_1^c, g_2h_2^c, \cdots, g_2h_m^c\}$.

3.3 Glance Gate Model

Based on multi-glance mechanism, we also propose the Glance Gate Model (GGM). The main block of GGM is shown in Fig.3. When we read the text at the second time, in view of the first impression information we obtained, our habit is to ignore the less interesting words directly rather than still reading them again. However, existing RNN models, e.g. LSTM model, have an input gate to control the current input, it still can't set less interesting or important information to zero.

In GGM, we use a binary glance gate to control the input, and it is defined as:

$$gate_t = softmax(W_g^g \cdot [gg_t^g, w_t, g_2h_{t-1}^g] + b^g) \quad (12)$$

where W_g^g is the projection matrix, and softmax only output two states $\{0, 1\}$. In glance gate model (GGM), gg_t^g still models the impression of the text, and calculated by the weighted sum of hidden states $\{g_1h_1, g_1h_2, \cdots, g_1h_m\}$:

$$gg_t^g = \sum_{i=1}^{m} \beta_i \cdot h_i \quad (13)$$

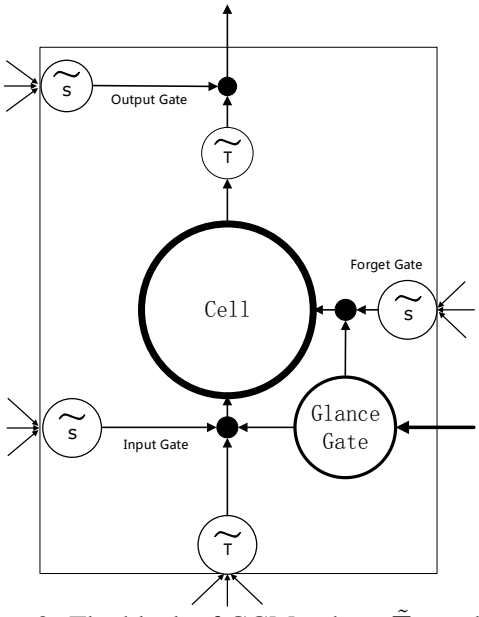


Figure 3: The block of GGM, where $\tilde{T}$ stands for $tanh()$ and $\tilde{S}$ stands for $sigmoid()$.

Where β_i measures the impression of i_{th} word for the current glance gate cell state gg_t^g. For brevity, we will not repeat the function of impression weight β_i and impression function f here.

As shown in the Fig.4, here we give an example of the GGM to process a sentence. Comparing to the LSTM model's input gate, the glance gate only has two states $\{0, 1\}$. When we care about the current word w_i, we input the word w_i into the GGM and update the hidden state. If the current word is meaningless, the GGM will directly discard the input word and keep the previous state without updating the hidden state. Thus the gates, cells states and output hidden states are defined as follows:

$$i_t^g = \sigma(W_i^g \cdot [g_2h_{t-1}^g, w_t] + b_i^g) \odot gate_t \quad (14)$$
$$f_t^g = \sigma(W_f^g \cdot [g_2h_{t-1}^g, w_t] + b_f^g) \odot gate_t \oplus (\mathbf{1} - gate_t) \quad (15)$$
$$o_t^g = \sigma(W_o^g \cdot [g_2h_{t-1}^g, w_t] + b_o^g) \odot gate_t \quad (16)$$
$$\tilde{c}_t^g = \tanh(W_c^g \cdot [g_2h_{t-1}, w_t] + b_c^g) \quad (17)$$
$$c_t^g = f_t^g \odot c_{t-1}^g + i_t^g \odot \tilde{c}_t^g \quad (18)$$
$$g_2h_t^g = o_t^g \odot \tanh(c_t^g) \odot gate_t + g_2h_{t-1}^g \odot (\mathbf{1} - gate_t) \quad (19)$$

where $\oplus$ stands for the element-wise addition.

Note that when the GGM close the glance gate, gate=$\{0\}$, the formulations above can be transformed as:

$$i_t^g = \mathbf{0} \qquad f_t^g = \mathbf{1} \qquad o_t^g = \mathbf{0}$$
$$c_t^g = f_t^g \odot c_{t-1}^g + i_t^g \odot \tilde{c}_t^g = c_{t-1}^g$$
$$g_2h_t^g = o_t^g \odot \tanh(c_t^g) \odot gate_t + g_2h_{t-1}^g \odot (\mathbf{1} - gate_t)$$
$$= g_2h_{t-1}^g$$

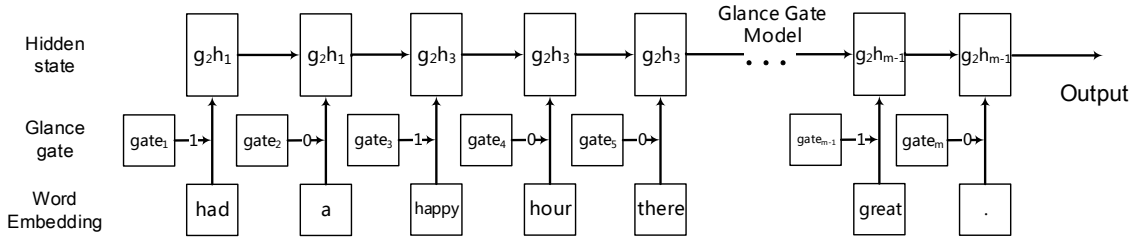


Figure 4: An example of the proposed GGM to process a sentence. In this example, when the glance gate open, the current word will input into the GGM, then output the hidden state. When the glance gate close, the model will ignore the current inputted word and keep the previous hidden state.

so when the glance gate close, the GGM will keep the previous state unchanged. Besides, when the GGM open the glance gate, namely gate=$\{1\}$, the formulations above can be transformed as:

$$
\begin{aligned}
i_t^g &= \sigma(W_i^g \cdot [g_2h_{t-1}^g, w_t] + b_i^g) \\
f_t^g &= \sigma(W_f^g \cdot [g_2h_{t-1}^g, w_t] + b_f^g) \\
o_t^g &= \sigma(W_o^g \cdot [g_2h_{t-1}^g, w_t] + b_o^g) \\
c_t^g &= f_t^g \odot c_{t-1}^g + i_t \odot \tilde{c}_t^g \\
g_2h_t^g &= o_t^g \odot \tanh(c_t^g)
\end{aligned}
$$

So the model can obtain the current input state $\tilde{c}_t^g$ and update the cell state c_t^g. We feed the text T into the GGM and obtain the output hidden states $g_2h^g = \{g_2h_1^g, g_2h_2^g, \cdots, g_2h_m^g\}$.

3.4 Model Training

To train our multi-glance mechanism models, we adopt softmax layer to project the text representation into the target space of C classes:

$$y = softmax(\tanh(W_s \cdot [\hat{g_2h}, \hat{g_1h}] + b_s)) \quad (20)$$

where $\hat{g_2h}$ is the attention weighted sum of the glance hidden states $\{g_2h_1, g_2h_2, \cdots, g_2h_m\}$, $\hat{g_1h}$ is the attention weighted sum of the hidden states $\{g_1h_1, g_1h_2, \cdots, g_1h_m\}$.

We use the cross-entropy as training loss:

$$L = -\sum_i \hat{y_i} \cdot \log(y_i) + \alpha|\theta|_2 \quad (21)$$

where $\hat{y_i}$ is the gold distribution for text i, θ represents all the parameters in the model.

4 Experiment

In this section, we conduct experiments on different datasets to evaluate the performance of multi-glance mechanism. We also visualize the glance layers in both glance models.

4.1 Datasets and Experimental Setting

We evaluate the effectiveness of our glance models on four different datasets . Yelp 2013 and Yelp2014 are obtained from the Yelp Dataset Challenge. IMDB dataset was built by Tang et al. (2015). Amazon reviews are obtained from Amazon Fine Food reviews. The statistics of the datasets are summarized in Table 1.

datasets	rank	docs	$\frac{sens}{docs}$	vocs
IMDB	1-10	84,919	16.08	105373
Amazon	1-5	556,770	5.67	119870
Yelp2013	1-5	78,966	10.89	48957
Yelp2014	1-5	231,163	11.41	93197

Table 1: Statistical information of IMDB, Amazon, Yelp 2013, Yelp 2014 datasets

The datasets are split into training, validation and test sets with the proportion of 8:1:1. We use the Stanford CoreNLP for tokenization and sentence splitting. For training, we pre-train the word vector and set the dimension to be 200 with Skip-Gram (Mikolov et al., 2013). In our glance models, the dimensions of hidden states and cells states are set to 200 and the hidden states and cells states initialized randomly. We adopt AdaDelta (Zeiler, 2012) to train our models , select the best configuration based on the validation set, and evaluate the performance on the test set.

4.2 Baselines

We compare our glance models with the following baseline methods.

Trigram adopts unigrams, bigrams and trigrams as text features and trains a SVM classifier.

TextFeature adopts more abundant features including n-grams, lexicon features, etc, and

Models \ Datasets	IMDB	Yelp2014	Yelp2013	Amazon
Trigram	39.9	57.7	56.9	54.3
TextFeature	40.2	57.2	55.6	-
PVDM	34.1	56.4	55.4	-
RNTN+RNN	40.0	58.2	57.4	-
NSC	42.7	62.7	62.2	75.1
RNN+ATT	43.1	63.2	62.7	75.4
GGM	43.7	63.4	63.0	75.2
GCM	**44.2**	**64.2**	**63.6**	**76.7**

Table 2: Text analysis results on IMDB, Yelp2014, yelp2013 and Amazon datasets. Evaluation metrics is Accuracy in percentage (higher the better). The best performance in each group is in **bold.**

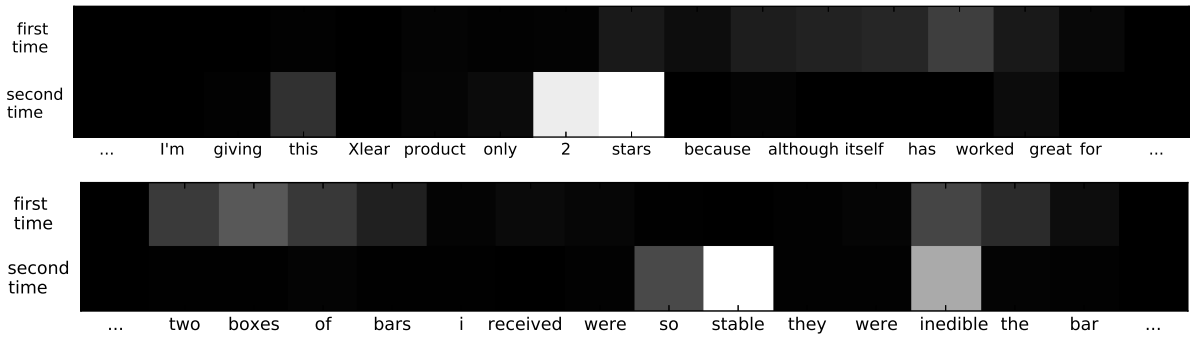


Figure 5: Visualization of the weights when we read the text twice with glance cell model (whiter color means higher weight).

trains a SVM classifier. (Kiritchenko et al., 2014)

RNTN+RNN uses Recursive Neural Neural Tensor Network to represent the sentences and Recurrent Neural Network to document analysis. (Socher et al., 2013)

PVDM leverages Paragraph Vector Distributed Memory (PVDM) algorithm for document classification.(Le and Mikolov, 2014)

NSC regards the text as a sequence and uses max or average pooling of the hidden states as features for classification.(Chen et al., 2016)

RNN+ATT adopts attention mechanism to select the important hidden states and represents the text as a weight sum of hidden states.

4.3 Model Comparisons

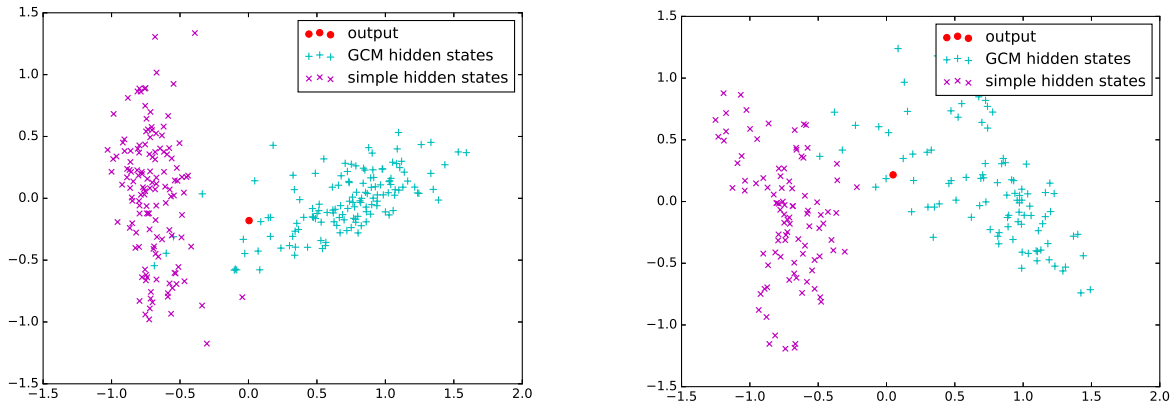
The experimental results are shown in Table 2. We can see that multi-glance mechanism based models, glance gate model (GGM) and glance cell model (GCM), achieve a better accuracy than traditional recurrent models, because of the guidance of the overview meaning we obtain at the first time of reading. With that guidance, we will get a better understanding of the text. While comparing to our glance models, existing RNN models read the text only once so they cannot have the general meaning to help them understand the text.

Comparing to attention-based recurrent models, the proposed glance cell model still has a better performance. The main reason for this is that when we read the text with the multi-glance mechanism, the glance hidden states have a better understanding of the text, so when we calculate the attention weight on each hidden states, the final output will also be better to represent the text.

When comparing the models we proposed, glance cell model gives a better performance than glance gate model. This is because we use multi-glance mechanism to filter words in glance gate model while we use multi-glance mechanism to add general information in glance cell model. Even though we only ignore the less important words in glance gate model when the gate is closed, some information is still lost comparing to glance cell model.

4.4 Model Analysis for Glance Cell Model

To establish the effectiveness of GCM, we choose some reviews in Amazon dataset and visualize them in the Fig.5. In each sub-figure, the first line



(a) (b)

Figure 6: Visualization of the hidden states calculated by simple RNN model $\{g_1h_1, g_1h_2, \cdots, g_1h_t\}$ (the purple spots), Glance Cell Model $\{g_2h_1^c, g_2h_2^c, \cdots, g_2h_t^c\}$ (the blue spots) and the final text representation (the red spots).

actually i'm not sure which film was better meet the parents or meet the fockers. both films were equally enjoyable. this movie is really funny. maybe it's because of a cast but everything works in this film. it's probably one of the best comedies made in this decade. Dustin Hoffman and Barbra Streisand both did great as Gaylord's parents. every character of this movie had it's own opinion and that was well portrayed in their dialogs. not like the original, this part is more making fun of Robert de Nero's character than of Ben Stiller 's character. i noticed that this film has many similarities with it's prequel but that's ok because it still was very funny.

Figure 7: Visualization of the gate state in Glance Gate Model. The words in color (blue and red) are input into the GGM, that meas the gate state is open. The words in gray are ignored by the GGM.

is the visualization of the weights when we read the text at the first time, the second line is the visualization that we read at the second time. Note that, whiter color means higher weight.

As shown in Fig.5, the first review has wrote the ranking stars in the text, which is a determining factor in product reviews, but we ignore them when we read at the first time. Well, with the guidance of multi-glance mechanism, when we read them again, we can not only find the ranking stars, but also give them high weights.

In the second review, comparing the results we read at the first time and the second time, though we may focus on some of the same words, e.g. inedible, we will give them different weights. We can observe that when reading at the second time, we give word 'inedible' a higher weight and word 'the' a lower the weight. The glance cell model can increase the weights of important words, so we can focus on more useful words when using multi-glance mechanism and glance cell model.

Next, we also choose two reviews in the dataset and visualize the hidden states which calculated by the glance cell model and a traditional recurrent model. As aforementioned in this paper, when using multi-glance mechanism, we will get the local information comparing to simple RNN models. As shown in Fig.6, the purple spots and the blue spots are the visualizations of the hidden states, and the purple spots belong to the simple RNN model while the blue spots belong to the glance cell model. The spot in red is the visualization of the final text representation. Note that, we use PCA to reduce the dimensions of the hidden states here. We can see that the blue spots are much more closer to the red spots than the purple spots, which mean the glance cell hidden states are more closer to the final text representations. It is the local information that makes the difference. So we can obtain a more general idea when using the glance cell model we proposed.

4.5 Model Analysis for Glance Gate Model

To demonstrate the effectiveness of the glance gate model, we choose a review in IMDB dataset and visualize the values of gates. As mentioned in this paper, the gates only have two states, closed and open. As shown in Fig. 7, the words in gray mean

i tried this tea in seattle two years ago and just loved it. it was unavailable at my local health food store , but i found it on amazon . their price and service are excellent . i would definitely recommend this tea !

(a) Model with Multi-glance Mechanism

i tried this tea in seattle two years ago and just loved it . it was unavailable at my local health food store , but i found it on amazon . their price and service are excellent . i would definitely recommend this tea !

(b) Simple RNN with Attention Mechanism

Figure 9: Visualization of the multi-glance mechanism weights and the simple RNN attention mechanism weights.

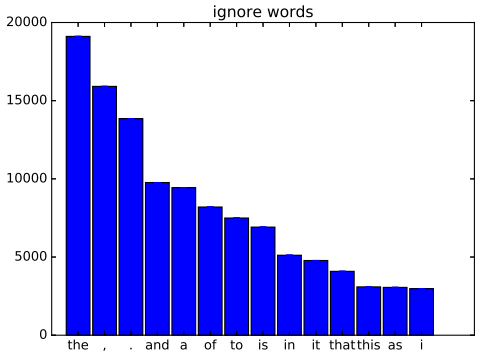


Figure 8: The statistics of the Top-ignored words in 1000 IMDB reviews.

when we read these words, the gates in GGM are closed. So these words are unable to pass through the gate. These words in color (blue and red) mean that the gates are open when we read these words. We can observe that when we read the text again, the glance gate model can ignore the less important words and focus on the more useful words. Surprisingly, the most important words are found, e.g. enjoyable, best comedies and funny (the red words in Fig.7). The model is able to find the adjectives, verbs and some nouns, which is more useful in the text understanding.

Besides, we also count the top-ignored words in 1000 IMDB reviews, and the results are shown in the Fig.8. We can see that most of the prepositions and adverds are ignored. Thus glance gate model can filter the less important words and concentrate on the more informative words.

4.6 Comparing to RNN with Attention Mechanism

To demonstrate the effectiveness of the multi-glance mechanism, we choose a review in Amazon dataset and visualize the parameters of weights in multi-glance model and attention based RNN model. As shown in Fig.9, the words in color (red and blue) are the top 10 important words in the review the word in red color are the top 5 important words. We can observe that multi-glance mechanism can find the more useful words, e.g. loved, excellent. What's more, multi-glance mechansim also can give these important words higher weights comparing to simple attention based RNN models which only read the review once.

5 Conclusion and Future work

In this paper, we propose a multi-glance mechanism in order to model the habit of reading. When we read a text, we may read it several times rather than once in order to gain a better understanding. Usually, we first read the text quickly and get a general idea. Under the guidance of this first impression, we will read many times until we get enough information we need. What's more, based on the multi-glance mechanism, we also propose two glance models, glance cell model and glance gate model. The glance cell model has a special cell to memorize the first impression information we obtain and add it into the current calculation. The glance gate model adopts a special gate to ignore the less important words when we read the text at the second time with multi-glance mechanism. The experimental results show that when we use the multi-glance mechanism to read the text, we are able to get a better understanding of the text. Besides, the glance cell model can memorise the first impression information and the glance gate model is able to filter the less important words, e.g. the, of. We will continue our work as follows:

- How to construct the first impression information more effectively? As proposed in this paper, some of the words in the text are redundant for us to understand text. So, we will sample some words of the text when reading it at the first time.
- The next step will be taken in the direction of algorithm acceleration and model lightweight design.

Acknowledgments

This work was supported in part by the Research Fund for the development of strategic emerging industries by ShenZhen city (No.JCYJ20160331104524983 and No.JCYJ20170412170118573).

References

Isabelle Augenstein, Tim Rockt?schel, Andreas Vlachos, and Kalina Bontcheva. 2016. Stance detection with bidirectional conditional encoding. pages 876–885.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1650–1659.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Gated-attention readers for text comprehension. *arXiv preprint arXiv:1606.01549*.

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

Andrej Karpathy, Justin Johnson, and Li Fei-Fei. 2015. Visualizing and understanding recurrent networks. *arXiv preprint arXiv:1506.02078*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research*, 50:723–762.

Quoc Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, pages 1556–1566.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1555–1565.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.