Will Repeated Reading Benefit Natural Language Understanding?

Lei Sha[1], Feng Qian[2], Zhifang Sui[1]

[1]Key Laboratory of Computational Linguistics, Ministry of Education
School of Electronics Engineering and Computer Science, Peking University
[2]Institute of Network Computing and Information Systems, Peking University
{shalei,nickqian, szf}@pku.edu.cn

Abstract. Repeated Reading (re-read), which means to read a sentence twice to get a better understanding, has been applied to machine reading tasks. But there have not been rigorous evaluations showing its exact contribution to natural language processing. In this paper, we design four tasks, each representing a different class of NLP tasks: (1) part-of-speech tagging, (2) sentiment analysis, (3) semantic relation classification, (4) event extraction. We take a bidirectional LSTM-RNN architecture as standard model for these tasks. Based on the standard model, we add repeated reading mechanism to make the model better "understand" the current sentence by reading itself twice. We compare three different repeated reading architectures: (1) Multi-level attention (2) Deep BiLSTM (3) Multi-pass BiLSTM, enforcing apples-to-apples comparison as much as possible. Our goal is to understand better in what situation repeated reading mechanism can help NLP task, and which of the three repeated reading architectures is more appropriate to repeated reading. We find that repeated reading mechanism do improve performance on some tasks (sentiment analysis, semantic relation classification, event extraction) but not on others (POS tagging). We discuss how these differences may be caused in each of the tasks. Then we give some suggestions for researchers to follow when choosing whether to use repeated model and which repeated model to use when faced with a new task. Our results thus shed light on the usage of repeated reading in NLP tasks.

1 Introduction

Bidirectional LSTM-RNN (BiLSTM) architecture has been successfully applied to sentiment classification [9], question answering [23], Sequence Tagging [6], by capturing syntactic and semantic aspects of text.

Repeated reading, first proposed to alleviate the lack of large scale training and test datasets [4, 14]. The motivation is: if we read a sentence again (re-read), we may get a better understanding of the sentence. There are three alternative repeated reading architectures: (1) Multi-level attention (2) Deep BiLSTM (3) Multi-pass BiLSTM.

Multi-level attention is mentioned as impatient reader in [4], which takes first-level attention as weight and take the weighted sum of the BiLSTM outputs as first-level memory, then use first-level memory to generate the second-level attention and the second-level memory and so on. The attention on each word are updated in each level, which lead to the update of memory. The extra information brought by re-read is contained in the second-level attention / memory.

Deep BiLSTM is an alternative repeated reading architecture. After the current sentence is read by an BiLSTM, the outputs of which are taken as the input of the second BiLSTM with different parameters. By reading again, the model can observe the sentence from a more integrated level, which is the effect of repeated reading.

Multi-pass BiLSTM is the third architecture. In this architecture, after the current sentence is read by an BiLSTM, A second BiLSTM with different parameters reads a delimiter and the current sentence again, but its memory state is initialized with the last cell state of the previous BiLSTM. Then the information brought by the second BiLSTM would contribute to the NLP task as extra information.

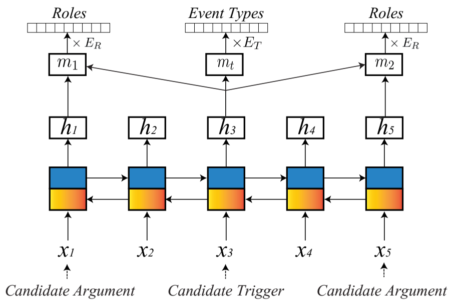

Fig. 1. The Standard BiLSTM Model.

Our goal in this paper is thus to investigate a number of NLP tasks with the goal of understanding if repeated reading models is useful to natural language understanding, and which kind of repeated reading model would offer specific advantages. We investigate four tasks to show the effect of repeated reading (re-read): (1) part-of-speech tagging: check the effect of re-read on word-level semantic classification, (2) sentiment analysis: check the effect of re-read on phrase-level and sentence-level sentiment classification, (3) semantic relation classification: check the effect of re-read on learning long-distance relationships between two words that may be far apart sequentially, (4) event extraction: check the effect of re-read on extracting hierarchical structure from text.

The principal motivation for this paper is to understand better when repeated reading models are needed to outperform simpler models by enforcing apples-to-apples comparison as much as possible. This paper applies existing models to existing tasks, barely offering novel algorithms or tasks. Our goal is rather an analytic one, to investigate different versions of repeated reading models. This work helps understand the pros and cons of different repeated reading architectures.

Task ID	Output	Objective
1	$O_{pos,t} = E\tanh(Wh_t + b_1) + b_2$	$J_{\text{pos}}(\theta) = \sum_i \sum_j \log p(y^{(i)}[j] \| x^{(i)}, \theta)$
2,3	$O_{\text{sa}} = W\left(\frac{1}{N_S}\sum_t^{N_S} h_t\right) + b$	$J_{\text{sa}}(\theta) = \sum_i \log p(y^{(i)} \| x^{(i)}, \theta)$
4	$O_{R,t} = E_R\tanh(W_R[h_t, h_{trig}] + b_{wr}) + b_{er}$ $O_T = E_T\tanh(W_T h_{trig} + b_{wt}) + b_{et}$	$J_{EE}(\theta) = \sum_i \big(\sum_j \log p(y_R^{(i)}[j] \| x^{(i)}, \theta)$ $+\lambda \log p(y_T^{(i)} \| x^{(i)}, \theta)\big)$

Table 1. The output layer function and objective function of the four task. 1: Part-of-speech tagging; 2: Sentiment Analysis; 3: Semantic relation classification; 4: Event Extraction

2 Models

2.1 Notations

We denote the text unit S (a phrase or a sentence) as a sequence of tokens / words: $S = \{w_1, w_2, \cdots, w_{N_S}\}$. Each word has a K-dim embedding e_w.

2.2 Standard BiLSTM Model and Application on Different Tasks

Long Short Term Memory (LSTM) LSTM [5] is defined as follows: given a sequence of inputs $X = \{x_1, x_2, \cdots, x_{n_X}\}$, an LSTM associates each timestep with an input, memory and output gate, respectively denoted as i_t, f_t and o_t. We notationally disambiguate e and h: e_t denotes the vector for individual text units (e.g., word or sentence) at time step t, while h_t denotes the vector computed by the LSTM model at time t by combining e_t and h_{t-1}. σ denotes the sigmoid function. The vector representation h_t for each time-step t is given by:

$$\begin{bmatrix} i_t \\ f_t \\ o_t \\ l_t \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} (W \cdot \begin{bmatrix} h_{t-1} \\ e_t \end{bmatrix} + b) \tag{1}$$

where $W \in \mathbb{R}^{4K \times 2K}$, $b \in \mathbb{R}^{4K \times 1}$.

Bidirectional Models [13] add bidirectionality to the recurrent framework where embeddings are calculated both forwardly and backwardly in each time step:

$$\begin{aligned} h_t^{\rightarrow} &= f(W^{\rightarrow} \cdot [h_{t-1}^{\rightarrow}, e_t] + b^{\rightarrow}) \\ h_t^{\leftarrow} &= f(W^{\leftarrow} \cdot [h_{t+1}^{\leftarrow}, e_t] + b^{\leftarrow}) \end{aligned} \tag{2}$$

Then, the output of time t is $h_t = [h_t^{\rightarrow}, h_t^{\leftarrow}]$.

Baseline Model The baseline model in this paper is shown in Figure 1, which is a standard bidirectional LSTM. Since event extraction is the most complicated task among the five, which needs to identify and classify the trigger as well as arguments, we take event extraction as an example in Figure 1. In event extraction task, the event type and the roles are to be decided by the candidate trigger and candidate arguments. Therefore, each of the output vectors $h_1, h_2, \cdots, h_{N_S}$, after concatenated with the trigger's corresponding output vector h_{trig}, can be taken as the feature vector for role classification. Also, h_{trig} is the feature for event type classification. Then the role output of the t-th word and the event type output can be calculated by Eq 3.

$$\begin{aligned} O_{R,t} &= E_R \tanh(W_R[h_t, h_{trig}] + b_{wr}) + b_{er} \\ O_T &= E_T \tanh(W_T h_{trig} + b_{wt}) + b_{et} \end{aligned} \tag{3}$$

where W_R, E_R, W_T, E_T are weight matrices, each dimension of $O_{R,t}$ represents the score of a role and each dimension of O_T represents the score of an event type.

In the event extraction case, given all of our training examples $(x^{(i)}; y_T^{(i)}, y_R^{(i)})$ ($y_T^{(i)}$ is the event type label, $y_R^{(i)}[j]$ is the role label of j-th candidate argument), we can then define the objective function as follows:

$$J_{EE}(\theta) = \sum_i (\sum_j \log p(y_R^{(i)}[j]|x^{(i)}, \theta) + \lambda \log p(y_T^{(i)}|x^{(i)}, \theta)) \tag{4}$$

where λ is a hyperparameter to balance the effect of trigger and argument. The conditional probability for event type and role is obtained in Eq 5:

$$p(r_i|x_t, \theta) = \frac{\exp(O_{R,t}[i])}{\sum_k \exp(O_{R,t}[k])}, p(t_i|x, \theta) = \frac{\exp(O_T[i])}{\sum_k \exp(O_T[k])} \tag{5}$$

The neural network architectures of other tasks have some minor differences from Figure 1. For sequence labeling tasks like POS tagging, we only need to classify each token to a certain type, so after denoting $y^{(i)}[j]$ as the POS tag of j-th token in the i-th case, the output layer and objective function is shown in Table 1. For the sequence classification tasks like task 2, we apply average pooling to the output of the BiLSTM, and send the pooling result to the logistic regression classifier.

For the relation classification task, the model needs to know the position of the two entities, so we assign an entity tag to each word using a commonly used encoding scheme BILOU (Begin, Inside, Last, Outside, Unit)[12]. Each entity tag corresponds to an one-hot vector with only the entity tag's corresponding entry is 1. We concatenate each word's embedding with the corresponding tag's vector, and take them as the input of BiLSTM. Then the relation classification can be transferred into sequence classification problem. The output layer and objective function is shown in Table 1.

2.3 Repeated Reading Models

The three possible repeated reading architecture are shown in Figure 2. All of them are based on the standard bidirectional LSTM architecture.

Multi-level Attention (MLA) In Figure 2a, the MLA model computes a memory vector by attention mechanism using the BiLSTM output as in Eq 6.

$$\begin{aligned} \alpha_1 &= \tanh(W_{b1}H + W_{t1}h_{trig}),\ \alpha_1 \in \mathbb{R}^{K \times N_S} \\ s_1 &= \mathrm{softmax}(w_1\alpha_1),\ s_1 \in \mathbb{R}^{N_S} \\ M_1 &= s_1 H^T,\ M_1 \in \mathbb{R}^K \end{aligned} \tag{6}$$

where $H = [h_1, h_2 \cdots h_{N_S}]$, $W_b, W_t \in \mathbb{R}^{K \times K}$, $w_1 \in \mathbb{R}^K$, s_1 is the attention weight. M_1 is the memory vector. To make the model understand the text better,

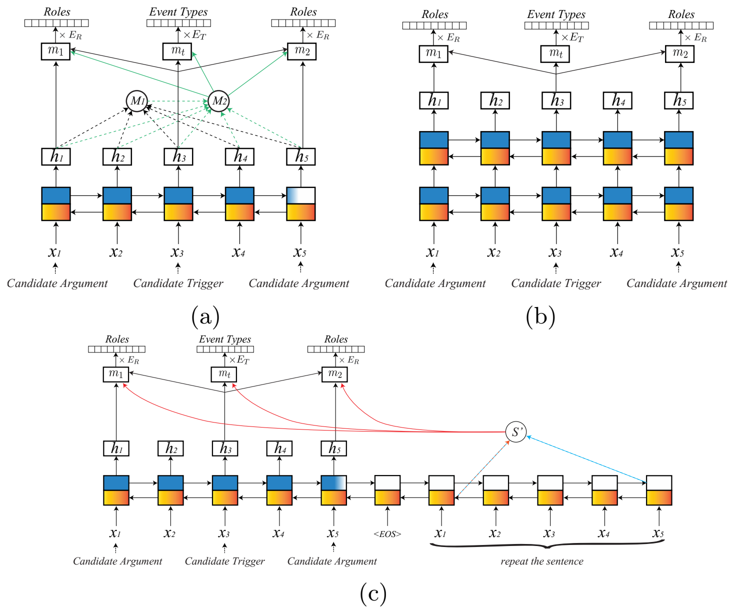

Fig. 2. Three re-read architectures: (a) Multi-level attention (b) Deep BiLSTM (c) Multi-pass BiLSTM

we equip the model with re-read ability by the second attention level as follows [4]:

$$\begin{aligned} \alpha_2 &= \tanh(W_{b2}H + W_{t2}h_{trig} + W_m M_1) \\ s_2 &= \mathrm{softmax}(w_2\alpha_2),\ s_2 \in \mathbb{R}^{N_S} \\ M_2 &= s_2 H^T,\ M_2 \in \mathbb{R}^K \end{aligned} \tag{7}$$

Then the output layer is shown as Eq 8.

$$\begin{aligned} O_{R,t} &= E_R \tanh(W_R[h_t, h_{trig}, M_2] + b_{wr}) + b_{er} \\ O_T &= E_T \tanh(W_T[h_{trig}, M_2] + b_{wt}) + b_{et} \end{aligned} \tag{8}$$

For the other 3 tasks, the attention only depends on the BiLSTM's output and the memory:

$$\begin{aligned} \alpha_1 &= \tanh(W_{b1}H) \\ \alpha_2 &= \tanh(W_{b2}H + W_m M_1) \end{aligned} \tag{9}$$

Deep BiLSTM(DB) In Figure 2b, we take the output of the BiLSTM as the input of the second BiLSTM [18]. Then the second BiLSTM is the re-read mechanism here. The output layer and objective function are the same as the standard BiLSTM (Table 1).

Multi-pass BiLSTM(MPB) The architecture in Figure 2c is the third possible architecture of repeated reading. After the current sentence is read by a BiLSTM, a second BiLSTM with different parameters is reading a delimiter and the current sentence again (the second pass). We denote the outputs of the forward and backward LSTMs of the second pass as $H_s^{\rightarrow}$ and $H_s^{\leftarrow}$ respectively. The encoding of the second pass is formed by the concatenation of the final forward and backward outputs $S' = [H_s^{\rightarrow}(N_S), H_s^{\leftarrow}(1)]$.

Then the output layer of event extraction (Task 4) is shown as Eq 10.

$$\begin{aligned} O_{R,t} &= E_R \tanh(W_R[h_t, h_{trig}, S'] + b_{wr}) + b_{er} \\ O_T &= E_T \tanh(W_T[h_{trig}, S'] + b_{wt}) + b_{et} \end{aligned} \tag{10}$$

The output layer of other tasks are simply replace the "h_t" in Table 1 with "$[h_t, S']$".

3 Experiments

We detail the experiment settings and results in this section. In each case we employed standard training frameworks for the three architectures: for each task, we used stochastic gradient decent using AdaDelta [24] with minibatches. Derivatives are calculated from standard back-propagation [1]. Hyper-parameters are tuned using the development dataset. The hyper-parameters include the size of each hidden layer, learning rate, λ in event extraction and parameters for L_2 penalizations. The model achieving best performance on the development set is used as the final model to be evaluated.

We trained word embeddings on the Wikipedia+Gigaword dataset using the WORD2VEC package[1].

The number of running iterations is also treated as a hyper-parameter to tune. we repeated the training procedure for each algorithm 20 times and report the average accuracies as well as the statistical significance testing result (we use Wilcoxon signed-rank test). Test scores that achieve significance level of 0.05 are marked by an asterisk ($*$).

3.1 Part-of-Speech Tagging

Task Description We use Wall Street Journal (WSJ) data as our testbed. We use Sections 0-18 of the Wall Street Journal (WSJ) data for training, sections 19-21 for validation and sections 22-24 for testing. For the five architectures (standard BiLSTM, 1-level MLA, 2-level MLA, deep BiLSTM, multi-pass BiLSTM), we take the 50 dimensional word vectors as input. The test accuracy and p value are shown in Table 2.

[1] `https://code.google.com/p/word2vec/`

	Accuracy	p value
Standard	**91.30**	-
MLA(1-level)	91.09	0.10960
MLA(2-level)	90.92	0.39532
DB	81.31	0.00096*
MPB	90.74	0.07346

Table 2. Accuracy for Different Models on Part of Speech Tagging. The p values are calculated between repeated reading models and the standard model.

Discussion According to the result in Table 2, we find that the standard BiLSTM model slightly outperforms most of the repeated reading architectures (1-level MLA, 2-level MLA and MPB), and significantly outperforms the DB model. This result implies that the repeated reading mechanism is not able to provide improvement to word-level POS tag classification, sometimes it can even make the performance worse (DB).

We suggest a few reasons to explain this phenomena. POS tagging is a very simple task, the model can receive enough information after read the sentence once. The MLA keeps a memory which records the understanding of the whole sentences. However, intuitively, in most cases, to decide whether a word is a noun, a verb or an adjective, the context information is enough [22]. The general comprehension of the sentence not only is not necessary, it may sometimes do harm to the understanding of each word. For example, the attention mechanism in MLA will bring semantic information of many other words, which may mislead the model to wrong POS tags.

3.2 Sentiment Analysis

Task Description For sentiment analysis, we experiment on the Stanford Sentiment TreeBank [20]. We intend to find the effect of repeated reading mechanism on short sequence classification and long sequence classification in this experiment.

	Phrase-level	Root-level	Total
Train	3.71	19.14	4.12
Dev	3.67	19.32	4.08
Test	3.67	19.19	4.08

Table 3. The average sentence length (word number) of Stanford Sentiment Treebank

Stanford Sentiment TreeBank contains gold-standard labels for every parse tree constituent, from the sentence to phrases to individual words. We transformed the dataset as illustrated in Figure 3. Each phrase is reconstructed from parse tree nodes and treated as a separate data point. Since the original treebank contains 11,855 sentences with 215,154 phrases, the reconstructed dataset contains 215,154 examples. All models (standard model & repeated reading models) are evaluated at both the phrase level (82,600 instances) and the sentence root level (2,210 instances). The average sentence length of this dataset is shown in Table 3. The evaluation result is shown in Table 4 and Table **??**.

Fine-grained	Phrase-level	Root-level	Total
Standard	80.72	**42.25**	79.91
MLA(1-level)	81.25(+0.53)	40.68(-1.57)	80.06(+0.15)
p value	0.0002*	0.00578*	0.0008*
MLA(2-level)	**81.61**(+0.89)	39.58(-2.67)	**80.15**(+0.24)
p value	0.008*	0.006*	0.0007*
DB	79.61(-1.11)	41.63(-0.62)	78.26(-1.63)
p value	0.0003*	0.03156*	0.03156*
MPB	81.11(+0.39)	42.08(-0.17)	79.88(-0.08)
p value	0.0003*	0.10524	0.87288

(a)

Coarse-grained	Phrase-level	Root-level	Total
Standard	80.79	72.57	79.89
MLA(1-level)	81.47(+0.68)	73.04(+0.47)	80.95(+1.06)
p value	0.0022*	0.0028*	0.0129*
MLA(2-level)	**81.65**(+0.86)	**73.64**(+1.07)	**81.31**(+1.42)
p value	0.0008*	0.0006*	0.0033*
DB	75.99(-4.80)	69.10(-3.47)	75.51(-4.38)
p value	0.0004*	0.0001*	0.0002*
MPB	80.71(-0.08)	72.60(+0.03)	79.78(-0.11)
p value	0.0600	0.158	0.0238

(b)

Table 4. (a) Fine-grained classification result (very negative, negative, neutral, positive, very positive). Test set accuracies on the Stanford Sentiment Treebank at phrase-level, root-level and total. The number in brackets are the different from the standard model's result at the corresponding level. The p values are calculated between repeated reading models and the standard model. (2) Coarse-grained classification result (negative, positive). Test set accuracies on the Stanford Sentiment Treebank at phrase-level, root-level and total.

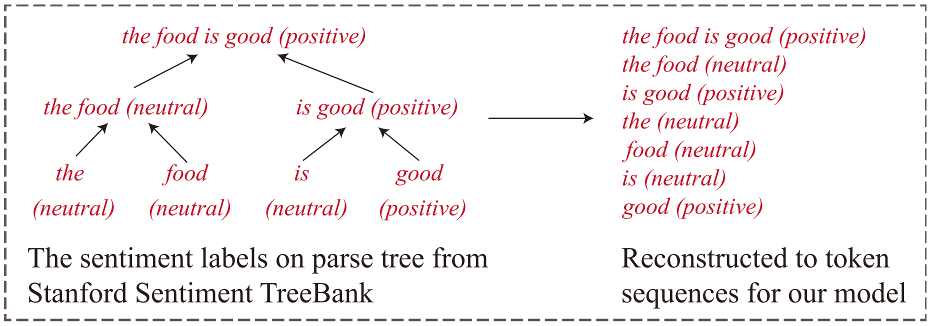

Fig. 3. Transforming Stanford Sentiment Treebank to Sequences for Sequence Models.

Discussion According to Table 4(a), in the phrase-level sentiment classification, repeated reading models have shown its effective. 1-level MLA, 2-level MLA and MPB model are significantly higher than the standard model. However, in the root-level sentiment classification, the repeated reading models are defeated by the standard model. Since the amount of phrase-level cases is much larger than the root-level cases, the performance of repeated reading models are still higher than the standard model in the total dataset.

We found that in this experiment, repeated reading models perform better at phrase-level sentiment classification but worse at sentence-level sentiment classification in fine-grained classification. But in coarse-grained classification, repeated reading models achieves better performance at both level sentiment classification. As is known to all, much of the sentiment label only depends on single sentiment words like "good" or "bad" instead of the whole phrase or sentence. In the Stanford sentiment Treebank, long sequences contain so much information that it is hard for even the attention mechanism to focus on the sentiment words. Too much redundant information brought by re-reading will mislead the model to judge the sentences as "neutral" instead of their original sentiment class "positive" or "negative". Therefore, repeated reading models were beaten by standard model in root-level fine-grained classification. After we delete the neutral cases (coarse-grained), we can see that repeated reading models can still achieve good results in Table 4(b). On the contrary, short sequences only contain limited information, so it is easier for repeated models to focus on the sentiment words. As a result, repeated reading models achieve better phrase-level result in both fine-grained and coarse-grained classification.

3.3 Semantic Relation Classification

Task Description We use the SemEval-2010 Task 8 dataset [3] to evaluate the performance of repeated reading models on finding semantic relationships between pairs of nominal. The dataset includes 8,000 training instances and 2,717 test instances. There are 9 relationships (with two directions) and an undirected "Other" class, so the task is formalized as a 19-class classification problem. For example, in "This [machine]$_{e_1}$ has two [units]$_{e_2}$." classifying the relation between [machine] and [units] as Component-Whole(e_2, e_1). The evaluation result is shown in Table 5.

	Accuracy	p value
Standard	75.54	-
MLA(1-level)	75.83 (+0.29)	0.06010
MLA(2-level)	**76.24** (+0.70)	0.01552$*$
DB	66.23 ($-$9.31)	0$*$
MPB	75.43 ($-$0.11)	0.81034

Table 5. Test set accuracies on the SemEval-2010 Semantic Relationship Classification task.

Discussion Unlike the earlier tasks, here MLA models (1-level MLA, 2-level MLA) yield much better performance than standard model, MPB model did not achieve significant result, and DB model still cannot beat the standard model. These results suggest that for semantic relation classification task, read for once is not enough. Intuitively, to find semantic relationships, for human beings, they first need to understand the whole sentence, then pay attention to the two nominal, finally consider what kind of relation they have. Just read the sentence for once is not likely to complete such complex process. Since the BiLSTM is a sequential model, it has to remember one nominal until the other one appears. So the attention mechanism may help highlight the important nominal to the model, which may lead to the success of MLA models.

3.4 Event Extraction

Task Description Event Extraction on the ACE 2005 dataset aims to discover event triggers with specific types and their arguments. ACE 2005 defines the event extraction task as three sub-tasks: identifying the trigger of an event, identifying the arguments of the event and distinguishing their corresponding roles. The newswire texts in ACE2005 dataset are divided into training(529 documents) / dev(10 documents) / testing(40 documents).

ACE2005 defines event as a structure composed of a trigger (a word which can best express the event) and several arguments (each plays a role in the event, 35 roles in total). The events can be classified into 8 types (33 subtypes).

We follow the previous works [7, 11, 10, 15, 17, 16] to evaluate the results.

- A trigger is considered to be correct if and only if its event type and offsets can match the reference trigger;
- An argument is correctly identified if and only if its event type and offsets can match any of the reference arguments;
- An argument is correctly identified and classified if and only if its event type, offsets and role can match any of the reference arguments.

The evaluation result is shown in Table 6.

	Trigger Cl	Argument Id	Argument Cl
	$F_1(\%)$	$F_1(\%)$	$F_1(\%)$
Standard	51.68	57.44	42.09
MLA(1-level)	53.77(+2.09)	59.34(+1.90)	41.43(-0.66)
p value	0.0022$*$	0.0003$*$	0.8891
MLA(2-level)	54.68(+3.00)	60.64(+3.20)	42.87(+0.78)
p value	0.0043$*$	0.0001$*$	0.0124$*$
DB	**57.22**(+5.54)	**60.75**(+3.31)	**43.65**(+1.56)
p value	0.0003$*$	0.0001$*$	0.0002$*$
MPB	55.21(+3.53)	59.03(+1.59)	41.32(-0.77)
p value	0.0015$*$	0.0102$*$	0.0230$*$

Table 6. Overall Performance of event extraction on ACE2005, The p values are calculated between repeated reading models and the standard model.

Discussion Different from the previous experiment, DB model achieves the best performance in event extraction task. This result can be explained by the property of deep BiLSTM architecture, which can build up progressively higher level representations [2]. Since the event is a hierarchical structure (a trigger with multiple arguments), it requires a higher perspective to better understand the sentence, which can be provided by the deep BiLSTM architecture. At the same time, MLA model also outperforms the standard model. Possible explanation is that the multi-level attention mechanism brings the ability to notice important information in the sentence, which also contributes to the extraction of event's hierarchical structure. However, MPB model didn't achieve significant improvement due to the fact that it is a sequential model, which is difficult to capture the hierarchical information.

4 Discussion

We have compared repeated reading models and standard model for representation learning on 4 distinct NLP tasks. For the best of our knowledge, no one has tried repeated reading mechanism on these tasks. For the comparisons between models, our results come with some caveats: First, we just apply the most basic form of repeated reading mechanism instead of various sophisticated algorithm variant. With the addition of hidden layers and advanced architectures (tensor layer [19], convolution layer [8], highway network [21], etc.) to sophisticated models, it becomes harder and harder to compare models "apple-to-apple". Second, in order to keep fairness when comparing models, we force every model to be trained exactly in the same way: AdaDelta with minibatches, same set of initializations, etc. Therefore, the architecture may not necessarily be the state-of-the-art for the task, and the training method may also not necessarily be the optimal way to train every model. Our conclusions might thus be limited to the algorithms employed in this paper, and it is possible that they can be extended to state-of-the-art models.

Our conclusions can be summarized as follows:

- For tasks like semantic relation extraction and event extraction, they have some relation to structured information. These tasks require several steps for even human beings to analyze. From the experiment, we see that the model need to re-read the sentence to achieve significantly better results. Therefore, re-read process can make the model better understanding a sentence.
- By contrast, in simple tasks like POS tagging, repeated reading mechanism did not bring any improvement. It made the performance worse instead. Since word senses have long been known to be related to POS tags, the word embedding itself is enough for the task. Therefore, repeated reading instead brings redundant information which worsen the performance. We can analogy this to a human psychological phenomena: semantic saturation.
- In phrase-level and sentence-level fine-grained sentiment analysis, we see different effects of repeated reading. In coarse-grained sentiment analysis, we see repeated reading achieves better result than standard model in both phrase-level and sentence-level. Since the neutral sentiment class is easy to be mistakenly classified to when there is redundant information, it would make repeated reading mechanism inefficient. When we cast the neutral class, repeated reading can still show its effectiveness.
- In most tasks, multi-level attention model wins good performance. This suggests that attention mechanism is good at noticing important information in a sentence. For example, it can notice the two nominals in sentiment relation classification task when single nominals need to be associated across a long distance. Also, it can pay attention to the event trigger and arguments when extracting events.
- Deep BiLSTM fails most of the tasks but it achieves the best performance in event extraction. This suggests that if hierarchical structure is not required by the task, the complexity brought by the deep architecture will definitely worsen the performance. The deep architecture is very powerful when dealing with high-level representation.
- Multi-pass BiLSTM seems cannot bring any outstanding benefits for the four tasks. Maybe it is the very reason why it never appears in any previous work.

5 Guidelines for NLPers

After the above discussion, we would like to propose some guidelines for researchers to follow.

When to use? If the task requires to understand the meaning of the whole sentence instead of single words, we suggest to use repeated reading mechanism.

Which to use? If the task requires several specific words (like in sentiment analysis, only several single semantic words can decide the sentiment label of the whole sentence. In relation classification, the relation is between two isolated entities), we suggest to use multi-level attention model.

If the task requires hierarchical structure (like event), we suggest to use deep BiLSTM model.

We do not suggest to use multi-pass BiLSTM model.

6 Conclusion

In this paper, we did an empirical study of repeated reading's effect on natural language understanding. We consider four tasks which represents the word-level semantic classification, phrase-level and sentence-level sentiment classification, long-distance relationship classification, and hierarchical structure extraction. Based on the standard BiLSTM architecture, we propose three possible repeated reading architecture (multi-level attention, deep BiLSTM, multi-pass BiLSTM). Then we analyzed on which kind of task and in which situation would each repeated reading model bring significant improvement. The result would shed light on the usage of repeated reading in NLP tasks.

Acknowledgements

We would like to thank our three anonymous reviewers for their helpful advice on various aspects of this work. This research was supported by the National Key Basic Research Program of China (No.2014CB340504) and the National Natural Science Foundation of China (No.61375074,61273318). The contact author for this paper is Zhifang Sui.

References

1. Goller, C., Kuchler, A.: Learning task-dependent distributed representations by backpropagation through structure. In: Neural Networks, 1996., IEEE International Conference on. vol. 1, pp. 347–352. IEEE (1996)
2. Graves, A., Jaitly, N., Mohamed, A.r.: Hybrid speech recognition with deep bidirectional lstm. In: Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. pp. 273–278. IEEE (2013)

3. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. pp. 94–99. Association for Computational Linguistics (2009)
4. Hermann, K.M., Kociský, T., Grefenstette, E., Espeholt, L., Kay, W., Suleyman, M., Blunsom, P.: Teaching machines to read and comprehend. CoRR abs/1506.03340 (2015)
5. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
6. Huang, Z., Xu, W., Yu, K.: Bidirectional lstm-crf models for sequence tagging. CoRR abs/1508.01991 (2015)
7. Ji, H., Grishman, R.: Refining event extraction through cross-document inference. In: Proceedings of the 46st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 254–262 (2008)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)
9. Li, J., Luong, T., Jurafsky, D., Hovy, E.H.: When are tree structures necessary for deep learning of representations? In: EMNLP (2015)
10. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 73–82. Association for Computational Linguistics, Sofia, Bulgaria (August 2013), `http://www.aclweb.org/anthology/P13-1008`
11. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 789–797. Association for Computational Linguistics (2010)
12. Ratinov, L.A., Roth, D.: Design challenges and misconceptions in named entity recognition. In: CONLL (2009)
13. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. Signal Processing, IEEE Transactions on 45(11), 2673–2681 (1997)
14. Sha, L., Chang, B., Sui, Z., Li, S.: Reading and thinking: Re-read lstm unit for textual entailment recognition. In: COLING. pp. 2870–2879 (2016)
15. Sha, L., Li, S., Chang, B., Sui, Z.: Joint learning templates and slots for event schema induction. In: Proceedings of NAACL-HLT. pp. 428–434 (2016)
16. Sha, L., Li, S., Chang, B., Sui, Z., Jiang, T.: Capturing argument relationship for chinese semantic role labeling. In: EMNLP. pp. 2011–2016 (2016)
17. Sha, L., Liu, J., Lin, C.Y., Li, S., Chang, B., Sui, Z.: Rbpb: Regularization-based pattern balancing method for event extraction. In: ACL (1) (2016)
18. Shi, Y., Yao, K., Tian, L., Jiang, D.: Deep lstm based feature mapping for query classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1501–1511. Association for Computational Linguistics, San Diego, California (June 2016), `http://www.aclweb.org/anthology/N16-1176`
19. Socher, R., Chen, D., Manning, C.D., Ng, A.Y.: Reasoning with neural tensor networks for knowledge base completion. In: Advances in Neural Information Processing Systems 26 (2013)
20. Socher, R., Perelygin, A., Wu, J.Y., Chuang, J., Manning, C.D., Ng, A.Y., Potts, C.: Recursive deep models for semantic compositionality over a sentiment tree-

bank. In: Proceedings of the conference on empirical methods in natural language processing (EMNLP). vol. 1631, p. 1642. Citeseer (2013)
21. Srivastava, R.K., Greff, K., Schmidhuber, J.: Highway networks. CoRR abs/1505.00387 (2015)
22. Toutanova, K., Klein, D., Manning, C.D., Singer, Y.: Feature-rich part-of-speech tagging with a cyclic dependency network. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1. pp. 173–180. Association for Computational Linguistics (2003)
23. Wang, D., Nyberg, E.: A long short-term memory model for answer sentence selection in question answering. In: ACL (2015)
24. Zeiler, M.D.: Adadelta: An adaptive learning rate method. arXiv preprint arXiv:1212.5701 (2012)